Inference in the Presence of Intractable Normalizing Functions

(Joint work with Jaewoo Park)

Department of Statistics, North Carolina State University

March 2018

Murali Haran

Department of Statistics, Penn State University

Outline

Outline

Research Areas

- Methods
  - I Statistical computing
  - II Spatial models
- Interdisciplinary Research
  - III Climate science
  - IV Infectious diseases

Models are central to these scientific problems – for interpretation and for predictions/projections, e.g. disease dynamics response to vaccination, or sea level rise in response to CO_2 emissions

Lots of overlap in categories: E.g. talk today is on a computing problem motivated by a spatial model for infectious disease

I. Statistical Computing

- Efficient Markov chain Monte Carlo (MCMC) and perfect sampling
- Stopping rules for MCMC algorithms: practical approach with theoretical justification
- Parallelizing MCMC
- Inference for climate and infectious disease models
  - High-dimensional random effects
  - Intractable likelihood functions
    - Intractable normalizing functions (**today's talk**)

II. Spatial Models for Ecology

(Spatial models inspired by bugs)

Zero-inflated hurdle model for spatial data on beetle counts

Spatial point process models with dynamics for ant movement data

II. Spatial Models

- High-dimensional non-Gaussian spatial data
  Subject of NC State talk in November 2017
  - Efficient projection-based approach for latent Gaussian random fields (Guan and Haran, 2018)
- Markov attraction-repulsion point process spatial model for respiratory syncitial virus (RSV) infections

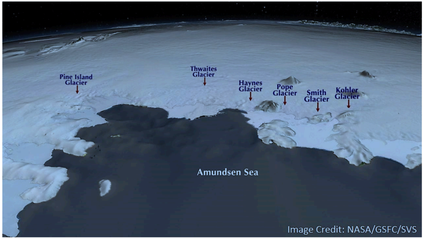
III. Climate Science

What is the future of the West Antarctic ice sheet?

Subject of NC State talk in October 2017

(Chang, Haran et al. 2016a,b; Haran et al., 2017)

National Snow and Ice Data Center, University of Colorado Boulder

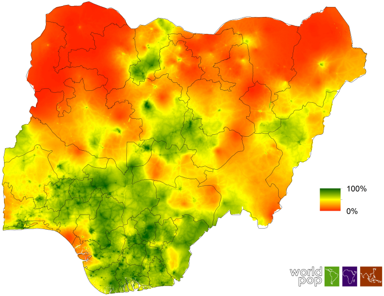


III. Climate Science: Combining Physics and Data

- Projecting future climate involves sophisticated physical models of climate systems
- Combining physics with observations
- Hierarchical modeling, dimension-reduction approaches, spatial models
- Inference for complex computer models
  - Emulation (approximation) and calibration (parameter inference) for these models using Gaussian processes
  - Challenges
    - high-dimensional spatial observations and model output
    - non-Gaussian spatial data
    - modeling data-model discrepancy

IV. Infectious Diseases

Measles in Nigeria



Measles vaccination: photo from UNICEF

Nigeria vaccine coverage map (from *BMC Medicine* Nigeria Health Map: http://ihmeuw.org/3m1c)

Infectious Disease Research Questions

- Impact of vaccination strategies on infectious diseases, e.g. rotavirus in Maradi (southern Niger)
- How do seasons affect meningitis transmission in Nigeria? How does this affect vaccination strategies?
- (New NIH grant) What leads to vaccine refusal in the U.S.? Potential impacts on epidemics?
- Methods
  - Hierarchical models for dynamics in space and time
  - Susceptible-Infected-Recovered (SIR)-type compartmental models for disease transmission

Interdisciplinary Setting

- Involves grads, postdoc, faculty from both disciplines
- Cross-disciplinary grant support. Examples:
  - NSF Computational Data-Enabled Science & Engr
  - NSF Sustainable Climate Risk Management
  - Department of Energy (DOE)
  - NIH MIDAS (Models of Infectious Disease Agent Study)
  - Gates Foundation
- Papers published in
  - Scientific journals, e.g. *Vaccine, Geoscientific Model Development, Nature Climate Change, Geophysical Research Letters, J of Climate*
  - Statistics journals, e.g. *Annals of Applied Stats, JASA, Environmetrics, J of Computational and Graphical Statistics*

PhD Students

Most dissertations combine applications and methodology.

Examples of where my PhD students have gone:

- Postdocs: NC State/SAMSI, U. Washington
- Tenure-track faculty: U. Cincinnati, U. Minnesota
- Research Labs: VA Tech Social Data Analytics, Los Alamos National Labs
- Government: Food and Drug Administration, National Security Agency

Undergraduate advisees have gone to top-flight graduate programs and industry

Outline

Research Overview

- Statistical Computing
- Spatial Models
- Climate Science
- Infectious Disease Modeling

Intractable Normalizing Functions

- Motivation: Attraction-Repulsion Point Process Model
- Bayesian Inference Basics

A Framework for Monte Carlo Methods

Implications for Practice

A Novel Emulation-Based Algorithm

Computing with Intractable Normalizing Functions

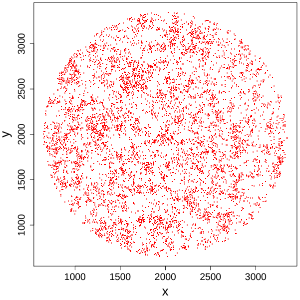
- Framework, comparisons for current algorithms
  - Park and Haran (2018a) “Bayesian Inference in the Presence of Intractable Normalizing Functions” *Journal of the American Statistical Association*, to appear
- New algorithm
  - Park and Haran (2018b) “A Function Emulation Approach for Doubly Intractable Distributions”, in preparation

Models with Intractable Normalizing Functions

- Models with intractable normalizing functions
  - Data: $\mathbf{x} \in \chi$, parameter: $\theta \in \Theta$
  - Model: $h(\mathbf{x}|\theta)/Z(\theta)$, where $Z(\theta) = \int_\chi h(\mathbf{x}|\theta)d\mathbf{x}$ is intractable
- Popular examples
  - Social network models: exponential random graph models (Robins et al., 2002; Hunter et al., 2008)
  - Models for lattice data (Besag, 1972, 1974)
  - Spatial point process models: interaction models (Strauss, 1975, Goldstein, Haran et al., 2015)
- Challenge: likelihood-based inference with $Z(\theta)$

Interaction Point Process

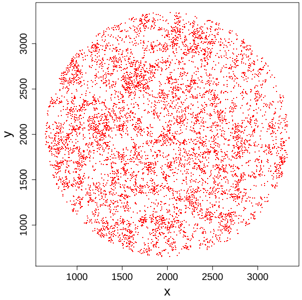
- Biologist's interest: study progression of viral infections
- Our goal: use data from imaging of cell cultures to study the spatial structure of an infection
- An *in vitro* cell culture study identifies and locates cells infected with two strains of the human respiratory syncytial virus (RSV-A and RSV-B)



Cells infected with RSV

Interaction Point Process

- Biologist's interest: study progression of viral infections
- Our goal: use data from imaging of cell cultures to study the spatial structure of an infection
- An *in vitro* cell culture study identifies and locates cells infected with two strains of the human respiratory syncytial virus (RSV-A and RSV-B)



Cells infected with RSV

Question: How does the presence of an infected cell impact infections in neighboring cells?

Attraction-repulsion Model

- Previous models (e.g. Strauss process) do not allow for repulsion *and* attraction
- Our point process model (Goldstein, Haran, et al., 2015) allows for both
- Allows us to easily compare interaction behavior for different strains of RSV
- Our model (like Strauss and other popular models) has an intractable normalizing function
  - Motivation for (i) studying existing algorithms for this problem, and (ii) developing new algorithms

Maximum Likelihood (ML) Inference

$$\hat{\theta} = \arg\max_{\theta \in \Theta} h(\mathbf{x}|\theta)/Z(\theta)$$

- Pseudolikelihood approximation (Besag, 1975)
  - Often a poor approximation
  - Awkward in a hierarchical model (not compatible with a real probability model)
- Markov chain Monte Carlo Maximum Likelihood (Geyer and Thompson, 1994)
  - Sensitive to choice of importance function
  - Optimization can be unstable
  - For some models, obtaining standard errors is challenging

    E.g. Attraction-repulsion point process

Bayesian Inference

- A Bayes approach can sidestep some of the challenges of ML inference
- Bayesian inference
  - Prior : $p(\theta)$
  - Posterior: $\pi(\theta|\mathbf{x}) \propto p(\theta)h(\mathbf{x}|\theta)/Z(\theta)$
- Inference is based on $\pi(\theta|\mathbf{x})$
- Generally approximated via Markov chain Monte Carlo (MCMC)
- MCMC is challenging due to $Z(\theta)$

Markov chain Monte Carlo Basics

- Construct Harris-ergodic Markov chain $\theta_1, \theta_2, \ldots$ with stationary distribution $\pi(\theta \mid \mathbf{x})$
- Treat $\theta_1, \theta_2, \theta_3, ...$ as samples from $\pi(\theta|\mathbf{x})$
- For any real-valued $g(\cdot)$, approximate $E_\pi(g(\theta))$ by

$$\hat{\mu}_n = \frac{\sum_{i=1}^{n} g(\theta_i)}{n}$$

- Under general conditions, $\hat{\mu}_n \to \mu$ as $n \to \infty$

The Metropolis-Hastings Algorithm

Recipe for constructing Markov chain: given θ_n, obtain θ_{n+1}

$$
\begin{tikzcd}
 & & \boxed{\theta_{n+1} = \theta'} \\
\boxed{\theta_n} \arrow[r] & \boxed{\text{propose } \theta' \sim q(\cdot|\theta_n)} \arrow[ur, "\text{accept}"] \arrow[dr, "\text{reject}"'] & \\
 & & \boxed{\theta_{n+1} = \theta_n}
\end{tikzcd}
$$

- Acceptance probability:

$$\alpha = \min\left\{ \frac{\pi(\theta'|\mathbf{x})q(\theta_n|\theta')}{\pi(\theta_n|\mathbf{x})q(\theta'|\theta_n)}, 1 \right\}$$

Form of α satisfies Markov chain “detailed balance” condition (original Metropolis et al. (1953) paper)

MCMC with Intractable Normalizing Functions

- Recall:
  - Prior : $p(\theta)$
  - Posterior: $\pi(\theta|\mathbf{x}) \propto p(\theta)h(\mathbf{x}|\theta)/Z(\theta)$
- Acceptance ratio for Metropolis-Hastings algorithm

$$\frac{\pi(\theta'|\mathbf{x})q(\theta_n|\theta')}{\pi(\theta_n|\mathbf{x})q(\theta'|\theta_n)} = \frac{p(\theta')Z(\theta_n)h(\mathbf{x}|\theta')q(\theta_n|\theta')}{p(\theta_n)Z(\theta')h(\mathbf{x}|\theta_n)q(\theta'|\theta_n)}$$

Cannot evaluate because of $Z(\cdot)$

Outline

- Research Overview
  - Statistical Computing
  - Spatial Models
  - Climate Science
  - Infectious Disease Modeling
- Intractable Normalizing Functions
  - Motivation: Attraction-Repulsion Point Process Model
  - Bayesian Inference Basics
- A Framework for Monte Carlo Methods
- Implications for Practice
- A Novel Emulation-Based Algorithm

Algorithms

Two classes of algorithms for Bayesian inference

I Auxiliary variable methods

- Generate an auxiliary random variate from model $h(\mathbf{x}|\theta)$
- Cancel $Z(\theta)$ in the acceptance ratio

II Likelihood approximation methods

- Approximate $Z(\theta)$ using Monte Carlo
- Use approximation $\widehat{Z}(\theta)$ in acceptance ratio

Exchange Algorithm

(Moller et al. (2006); Murray et al. (2007))

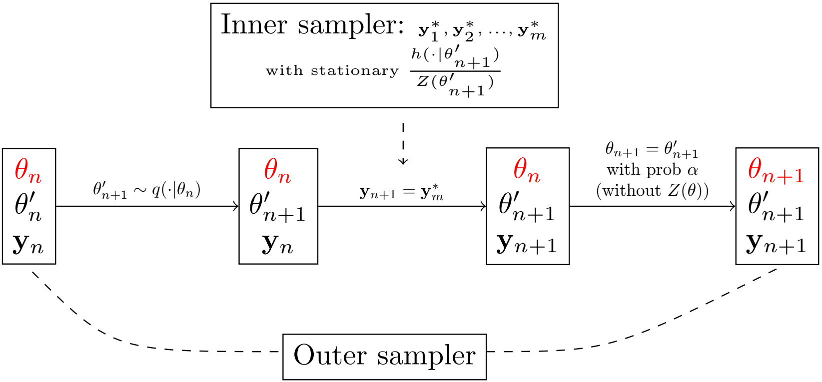
$$\begin{array}{|c|}\hline \theta_n \\ \theta'_n \\ \mathbf{y}_n \\ \hline\end{array} \xrightarrow{\theta'_{n+1} \sim q(\cdot|\theta_n)} \begin{array}{|c|}\hline \theta_n \\ \theta'_{n+1} \\ \mathbf{y}_n \\ \hline\end{array} \xrightarrow{\mathbf{y}_{n+1} \sim \frac{h(\cdot|\theta'_{n+1})}{Z(\theta'_{n+1})} \text{ i.i.d.}} \begin{array}{|c|}\hline \theta_n \\ \theta'_{n+1} \\ \mathbf{y}_{n+1} \\ \hline\end{array} \xrightarrow{\substack{\theta_{n+1} = \theta'_{n+1} \\ \text{with prob } \alpha \\ (\text{without } Z(\theta))}} \begin{array}{|c|}\hline \theta_{n+1} \\ \theta'_{n+1} \\ \mathbf{y}_{n+1} \\ \hline\end{array}$$

- Goal: obtain next state of Markov chain (θ_{n+1}) from current state (θ_n) by using auxiliary variable **y**
  - Update augmented state $(\theta_n, \theta'_n, \mathbf{y}_n)$ instead of updating (θ_n)
  - This cancels out $Z(\theta)$ in acceptance ratio
  - Then take the marginal samples of (θ_n)

Exchange Algorithm

- Asymptotically exact, that is, as $n \to \infty$, transition kernel of Markov chain converges to π
- Very clever and simple (in theory)
- **Requires that we draw exact samples from probability model for each proposed** θ
  - Need to do perfect sampling with Markov chains
  - Infeasible or very expensive in general
- Alternative: Double Metropolis-Hastings (Liang, 2010)

Double Metropolis-Hastings (DMH)



- Theory assumes length of inner and outer sampler go to infinity. **Asymptotically inexact** in practice
- Most practical of algorithms we considered

The Adaptive Exchange Algorithm (AEX)

Liang et al. (2016)

- Basic idea: AEX replaces independent sampling of **y** with a re-sampling method
- With increasing iterations, more samples get added to $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$: re-sampling $\approx$ exact sampling of **y** from $h(\mathbf{y}|\theta')/Z(\theta')$
- **Asymptotically exact** without perfect sampling
- Slow and (extremely) complicated to code/tune
- Huge storage requirements unless sufficient statistics are of low dimensions

Auxiliary Variable Methods: Summary

- List
  - Exchange algorithm (two versions)
  - Adaptive exchange algorithm
  - Double Metropolis-Hastings
- Sequential algorithms, not amenable to easy parallelization
- Each iteration involves runnning a (sequential) MCMC algorithm
- Double M-H: asymptotically inexact but easy to code

Likelihood Approximation Method

Atchade, Lartillot and Robert (ALR) Algorithm

$\{\theta^{(1)}, \ldots, \theta^{(d)}\}$ is particles covering Θ

$\mathbf{x}_{n+1}$ is drawn from $\left\{ \frac{h(\cdot|\theta^{(1)})}{Z(\theta^{(1)})}, \ldots, \frac{h(\cdot|\theta^{(d)})}{Z(\theta^{(d)})} \right\}$

Add $\mathbf{x}_{n+1}$ to $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$

$$\boxed{\theta_n} \longrightarrow \boxed{\theta_{n+1}}$$

$\theta' \sim q(\cdot|\theta_n)$

$\theta_{n+1} = \theta'$ with prob α

(Multiple importance sampling estimate $\widehat{Z(\theta)}$ using $\{\mathbf{x}_1, ..., \mathbf{x}_{n+1}\}$)

- Basic idea: approximate $Z(\theta)$ adaptively through weighted importance sampling (Atchade et al., 2015)

ALR Algorithm

- With increasing iterations, more samples get added to $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$: approximation $\widehat{Z}_{n+1}(\theta)$ becomes more accurate
- **Asymptotically exact** without independent sampling
- Memory issues: have to store large number of sampled data used in importance sampling
- Comparable to AEX algorithm in speed

Summary of Likelihood Approximation Algorithms

- List with lots of overlap
  - ALR (Atchade, Lartillot, Robert, 2012) algorithm
  - Pseudo-marginal MCMC (Andrieu and Roberts, 2009), e.g. Russian roullette algorithm (Lyne et al., 2015)
  - Noisy MCMC (Alqueir et al., 2016) and hybrids
- Huge memory requirements unless there are low-dimensional sufficient statistics

Outline

Interaction Point Process Model

- Simulated example: point process with $n = 200$
  - Data $\mathbf{x} \in R^{200 \times 2}$ are coordinates of point process
  - Evaluating $h(\mathbf{x}|\theta)$ requires calculating distance matrix of $\mathbf{x}$.
  - AEX and ALR are impractical (need to store 200×200-dimensional distance matrices for each particle with each iteration)
- Double Metropolis-Hastings (**inexact**) is only practical approach
- DMH results are accurate if the inner sampler is run for long enough
- For $n \approx 3,000$
  - Very efficiently coded DMH takes $\approx$ 19 hours
  - All other algorithms are infeasible
- Larger problem with $n = 13,000$: Even DMH is infeasible

Outline

New Emulation-Based Algorithm

(Park and Haran, 2018b)

- All existing algorithms are computationally very expensive
- An alternative is desirable
- Basic idea
  - Approximate $Z(\theta)$ using importance sampling on some design points
  - Use Gaussian process emulation approach to interpolate this function at any new value
  - We have some theory to justify this work as number of design points and number of importance sampling draws increases

Normalizing Function Emulation Algorithm

Part 1: Construct two-stage approximation

- Pre-MCMC

1. For each $\theta \in \{\boldsymbol{\theta}^{(1)}, ..., \boldsymbol{\theta}^{(d)}\}$, obtain importance sampling approximation $\widehat{Z}_{IMP}(\theta)$
2. Fit Gaussian process (GP) to $\{\widehat{Z}_{IMP}(\boldsymbol{\theta}^{(1)}), ..., \widehat{Z}_{IMP}(\boldsymbol{\theta}^{(d)})\}$
   Now for each θ obtain GP approximation, $\widehat{Z}_{GP}(\theta)$

Part 2: MCMC algorithm with GP approximation

- Given $\boldsymbol{\theta}_n \in \boldsymbol{\Theta}$ at nth iteration.

3. Propose $\boldsymbol{\theta}' \sim q(\cdot|\boldsymbol{\theta}_n)$
4. Obtain $\widehat{Z}_{GP}(\boldsymbol{\theta}')$, accept $\boldsymbol{\theta}'$ with

$$\alpha = \min\left\{\frac{p(\boldsymbol{\theta}')h(\mathbf{x}|\boldsymbol{\theta}')\widehat{Z}_{GP}(\boldsymbol{\theta})q(\boldsymbol{\theta}|\boldsymbol{\theta}')}{p(\boldsymbol{\theta})h(\mathbf{x}|\boldsymbol{\theta})\widehat{Z}_{GP}(\boldsymbol{\theta}')q(\boldsymbol{\theta}'|\boldsymbol{\theta})}, 1\right\}$$

Computational Benefits

- Can compute in parallel; much of this is done "offline", before running the algorithm
- Preliminary results: major reduction in computing time, likely to be more drastic for harder problems
- Two versions of our approach
  (i) NormEmul emulate $Z(\theta)$ with $\widehat{Z}_{GP}(\theta)$
  (ii) LikEmul emulate $\mathcal{L}(\theta) = h(\mathbf{x}|\theta)/Z(\theta)$ with $\widehat{\mathcal{L}}_{GP}(\theta)$

Theory

(Park and Haran, 2018b)

The Markov chain constructed by the function-emulation algorithm, with $n-$step transition kernel $P^n_{GP}(x,\cdot)$, converges in total variational distance to the target distribution π

$$\lim_{n\to\infty} \|P^n_{GP}(x,\cdot) - \pi(\cdot)\|_{TV} = 0, \forall x \in \Omega$$

- Assumes Markov chain is fast mixing (uniformly ergodic), acceptance probability difference is bounded
- Assumptions are satisfied for all our examples
- Assumes # samples for $\widehat{Z}_{IMP}$ and number of design points for $\widehat{Z}_{GP}$ both go to infinity. Hence, in practice asymptotically inexact (like DMH)

Simulated Data Results

- Consider a real data set, $n = 3000$ points
- Comparing fastest existing algorithm DMH (Double Metropolis-Hastings) with our two new algorithms
- HPD=highest posterior density region

θ_1	Mean	95%HPD	Time(hour)
DMH	1.34	(1.30,1.39)	18.99
NormEmul	1.34	(1.30,1.39)	3.60
LikEmul	1.34	(1.29, 1.39)	2.53

Recap

- Introduced a new model for both attraction and repulsion in point processes (Goldstein, Haran et al., 2015)
- Inference is computationally challenging due to an intractable normalizing function
- Provided a framework and comparison of existing computational methods (Park and Haran, 2018a)
- Fast new algorithms (Park and Haran, 2018b)
- New model + new algorithm: can now fit (for the first time) attraction-repulsion model to large point processes
- (Skipped) Can answer scientific questions based on fitting attraction-repulsion model to multiple cell cultures

The Last Slide

- This methodology is widely applicable, e.g. network (ERGM) models (details in paper)
- Lots of open problems
  - Improving efficiency, e.g. design point selection. No method handles high-dimensional distributions well
  - Stopping rules
  - Automated tuning
  - Theoretical challenges
- General modern statistical computing challenge: How can we carry out inference when likelihood function/objective function is complicated or high-dimensional?

References

- Park and Haran (2018a) Bayesian Inference in the Presence of Intractable Normalizing Functions, *Journal of the American Statistical Association*, to appear
- Møller, J., Pettitt, A. N., Reeves, R., and Berthelsen, K. K. (2006) An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants, *Biometrika*
- Murray, I., Z. Ghahramani, and D. MacKay (2006) MCMC for doubly-intractable distributions. *Proc of 22nd Annual Conf on Uncertainty in Artificial Intelligence* UAI06
- Liang, F. (2010) A Double Metropolis-Hastings sampler for spatial models with intractable normalizing constants. *Journal of Statistical Computation and Simulation*

References

- Atchade, Y., Lartillot, N. and Robert, C. (2013) Bayesian computation for statistical models with intractable normalizing constants. *Brazilian Journal of Probability and Statistics*
- Liang, F., Jin, I. H., Song, Q., and Liu, J. S. (2015) An adaptive exchange algorithm for sampling from distributions with intractable normalising constants. *Journal of the American Statistical Association*
- Goldstein, J., Haran, M., Simeonov, I., Fricks, J., and Chiaromonte, F. (2015) An attraction-repulsion point process model for respiratory syncytial virus infections. *Biometrics*

Exponential Random Graph Models

2-star	Mean	95%HPD	ESS	Time(second)
DMH	1.224	(-0.102,2.587)	986.796	6.620
AEX	1.245	(0.169,2.579)	991.874	126.460
ALR	1.247	(0.160,2.518)	1456.573	2500.370
Gold standard	1.265	(0.084,2.498)	9655.902	

Table: 30,000 MCMC samples are used for all algorithms.

- An exponential random graph model: $\theta \in R^4$

  $h(\theta|\mathbf{x})/Z(\theta) = exp\{\theta' S(\mathbf{x})\}/Z(\theta)$
- Business networks among 16 Florentine families (Padgett, 1994)
  - Data $\mathbf{x} \in R^{16\times16}$ representing connections among 16 families (0-1).
  - For AEX and ALR, we can only store 4-dimensional sufficient statistics for a sampled $\mathbf{x}_{n+1}$ with each iteration.

Exchange Algorithm

- Joint distribution: $\pi(\theta_n, \theta'_n, \mathbf{y}_n|\mathbf{x})$

  $= \pi(\theta_n|\mathbf{x})\pi(\theta'_n|\theta_n)\pi(\mathbf{y}_n|\theta'_n) \propto p(\theta_n)\frac{h(\mathbf{x}|\theta_n)}{Z(\theta_n)}q(\theta'_n|\theta_n)\frac{h(\mathbf{y}_n|\theta'_n)}{Z(\theta'_n)}$
- Algorithm

  1. Update $[\theta'_n, \mathbf{y}_n]$:
     - Propose $\theta'_{n+1} \sim q(\cdot|\theta_n)$
     - Propose $\mathbf{y}_{n+1} \sim h(\cdot|\theta'_{n+1})/Z(\theta'_{n+1})$ independently
  2. Update $[\theta_n]$ through swapping proposal:
     - $S : (\mathbf{x}, \theta_n), (\mathbf{y}_{n+1}, \theta'_{n+1}) \Rightarrow (\mathbf{x}, \theta'_{n+1}), (\mathbf{y}_{n+1}, \theta_n)$
     - Accept $\theta_{n+1} = \theta'_{n+1}$ with probability

$$\alpha = \frac{\cancel{S(\theta_n|\theta'_{n+1})}\,\pi(\theta'_{n+1}, \theta_n, \mathbf{y}_{n+1}|\mathbf{x})}{\cancel{S(\theta'_{n+1}|\theta_n)}\,\pi(\theta_n, \theta'_{n+1}, \mathbf{y}_{n+1}|\mathbf{x})}$$

$$= p(\theta'_{n+1})h(\mathbf{x}|\theta'_{n+1})\cancel{Z(\theta_n)}h(\mathbf{y}|\theta_n)\cancel{Z(\theta'_{n+1})}q(\theta_n|\theta'_{n+1})$$

Attraction-repulsion Model

Previous models did not allow for repulsion *and* attraction

New point process model (Goldstein, Haran, et al., 2015):

The likelihood can be written as

$$\mathcal{L}(X|\Theta) = \frac{f(X|\Theta)}{c(\Theta)}, f(X|\Theta) = \lambda^n \left[\prod_{i=1}^{n} e^{\min\left[\sum_{i \neq j} log(\phi(x_i, x_j)), k\right]} \right]$$

Model parameters:

- λ is the intensity of the process
- $\theta_1, \theta_2, \theta_3$ control the shape of $\phi(r)$.
- R is the minimum distance allowed between points
- k is a truncation constant necessary to prevent “clumping" behavior

Important: $c(\Theta)$ is intractable. This makes computing very

Appendix: Double Metropolis-Hastings (DMH)

- Basic idea: DMH replaces exact sampling of $\mathbf{y}$ with MCMC sampling.
- Algorithm
  1. $\theta' \sim q(\cdot|\theta_n)$
  2. $\mathbf{y} \sim h(\cdot|\theta')/Z(\theta')$ via m-number of MCMC updates.
     - $\mathbf{y}_{new} \sim T(\mathbf{y}_{new}|\mathbf{y}_{old})$, where $T(\mathbf{y}_{new}|\mathbf{y}_{old})$ is proposal distribution.
     - Accept $\mathbf{y}_{new}$ with probability $\frac{h(\mathbf{y}_{new}|\theta')T(\mathbf{y}_{old}|\mathbf{y}_{new})}{h(\mathbf{y}_{old}|\theta')T(\mathbf{y}_{new}|\mathbf{y}_{old})}$.
     - Repeat this procedure m-times, and regard the last state of the resulting Markov chain as sample from $h(\mathbf{y}|\theta')/Z(\theta')$.
  3. Accept $\theta_{n+1} = \theta'$ with probability

$$\alpha = \frac{p(\theta')h(\mathbf{x}|\theta')h(\mathbf{y}|\theta_n)q(\theta_n|\theta')}{p(\theta_n)h(\mathbf{x}|\theta_n)h(\mathbf{y}|\theta')q(\theta'|\theta_n)}$$

Appendix: The Adaptive Exchange Algorithm (AEX)

- Basic idea: AEX replaces exact sampling of **y** with re-sampling method.
- Algorithm (at $n+1$st iteration)

1. **Auxiliary chain**: $\mathbf{x}_{n+1} \sim \{h(\mathbf{x}|\theta^{(1)})/Z(\theta^{(1)}), ..., h(x|\theta^{(d)})/Z(\theta^{(d)})\}$ via stochastic approximation Monte Carlo and keep sampled $\mathbf{x}_{n+1}$.
2. **Target chain**:
   - $\theta' \sim q(\theta'|\theta_n)$
   - $\mathbf{y} \sim \{\mathbf{x}_1, ..., \mathbf{x}_{n+1}\}$ w.r.t. $p(\mathbf{x}_i|\theta')$ for $i = 1, ..., n+1$.
   - Accept $\theta_{n+1} = \theta'$ with probability

$$\alpha = \frac{p(\theta')h(\mathbf{x}|\theta')h(\mathbf{y}|\theta_n)q(\theta_n|\theta')}{p(\theta_n)h(\mathbf{x}|\theta_n)h(\mathbf{y}|\theta')q(\theta'|\theta_n)}$$

Appendix: ALR Algorithm

- Basic idea: approximate $Z(\theta)$ adaptively through weighted importance sampling.
- $\widehat{Z}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \frac{h(\mathbf{x}_i|\theta)}{h(\mathbf{x}_i|\theta^{(1)})}$, where $\mathbf{x}_i \sim h(\mathbf{x}|\theta^{(1)})/Z(\theta^{(1)})$ might be poor if $\theta^{(1)}$ is far from θ.
- Algorithm

  1. $\mathbf{x}_{n+1} \sim \{h(\mathbf{x}|\theta^{(1)})/Z(\theta^{(1)}), ..., h(\mathbf{x}|\theta^{(d)})/Z(\theta^{(d)})\}$ via stochastic approximation and keep sampled $\mathbf{x}_{n+1}$.
  2. $\theta' \sim q(\theta'|\theta_n)$.
  3. $\widehat{Z}(\theta) = \sum_{j=1}^{d} w_j \frac{1}{N_j} \sum_{i=1}^{N_j} \frac{h(\mathbf{x}_i|\theta)}{h(\mathbf{x}_i|\theta^{(j)})}$ using $\{\mathbf{x}_1, ..., \mathbf{x}_{n+1}\}$, where $\sum_{j=1}^{d} N_j = n+1$ and $\sum_{j=1}^{d} w_j = 1$.
  4. Accept $\theta_{n+1} = \theta'$ with probability

$$\alpha = \frac{p(\theta')\widehat{Z}(\theta_n)h(\mathbf{x}|\theta')q(\theta_n|\theta')}{p(\theta_n)\widehat{Z}(\theta')h(\mathbf{x}|\theta_n)q(\theta'|\theta_n)}$$

Appendix: Computational Complexity

DMH	Exponential family	Point process
Complexity	$\mathbf{G(n)} + \mathbf{L(n)}$	$\mathbf{G(n^2)} + 3\mathbf{L(n^2)}$
$*$AEX	Exponential family	Point process
Complexity	$(1-\beta)[\mathbf{G(n)} + \mathbf{L(n)}]$	$(1-\beta)\mathbf{G(n^2)} + [N_1/b + m + 5]$
Memory	$\mathbf{p} + 2$	$\mathbf{n^2} + 3$
$*$ALR	Exponential family	Point process
Complexity	$\mathbf{G(n)} + \mathbf{L(n)}$	$\mathbf{G(n^2)} + [(d + 2m + 2)/c + 1]\mathbf{L}$
Memory	$\mathbf{p} + 1$	$\mathbf{n^2} + d + 1$

Table: Computational complexity and memory costs of algorithms.

- Key notations:

  n: size of data, **p**: dimension of θ,