

Particle Markov Chain Monte Carlo Techniques (PMCMC)

Nathaniel Osgood

Basics

- Supports estimating (via sampling from) **joint** distributions of
 - Parameters (static and evolving)
 - System state over time
- Very computationally expensive
 - 10,000+ MCMC iteration (burn-in period required)
 - Each MCMC iteration requires running particle filtering to sample *trajectory* of latent states
 - Each particle filtering must run model
 - Highly parallelizable
- Family of algorithms

Family of Algorithms

- Presented here: Particle Marginal PMCMC
- Particle Gibbs PMCMC
 - more controversial
- Particle independent PMCMC

Basic Approach to Sampling from $p(\theta, x_{1:T} | y_{1:T})$

- Find initial parameters θ by looping to find non-zero posterior value
- Outer loop: MCMC (Metropolis-Hastings) iteration, sampling parameters θ
- Inner loop: Particle filtering, sampling from trajectories according to weight conditional on the candidate θ^*
 - This particle filtering takes place with respect to current candidate parameter value θ^* for Metropolis Hastings
 - Entire trajectory is sampled, according to final weight
 - An ancestry matrix is maintained
 - Resampling occurs upon each observation
 - Weights across successive observations across all trajectories support calculating the posterior for the candidate parameter value θ^*
- For each sampled value, emit sampled values (new or existing)
 - If accepted candidate, emit those parameter values and sampled latent state
 - If did not accepted candidate, state in current place, and (re-)emit those (pre-existing) parameter values and the latent state (previously) sampled from them

Basic Approach

Find θ giving non-zero value for (relative) posterior density $p_{\theta}(y_{1:T})p(\theta) = p(y_{1:T}|\theta)p(\theta)$

Metropolis hastings step to sample next value of θ and a value of full trajectories of latent variables $x_{1:T}$ for a given θ

Generate candidate θ^* based on a multivariate normal perturbation to the existing θ (either transforming the coordinates or resampling as required to deal with boundaries)

The acceptance ratio is $\frac{p_{\theta^*}(y_{1:T})p(\theta^*)q(\theta|\theta^*)}{p_{\theta}(y_{1:T})p(\theta)q(\theta^*|\theta)} = \frac{p_{\theta^*}(y_{1:T})p(\theta^*)}{p_{\theta}(y_{1:T})p(\theta)}$ (equivalence for symmetric q)
denominator was calculated in previous step (initial or prev. trial)

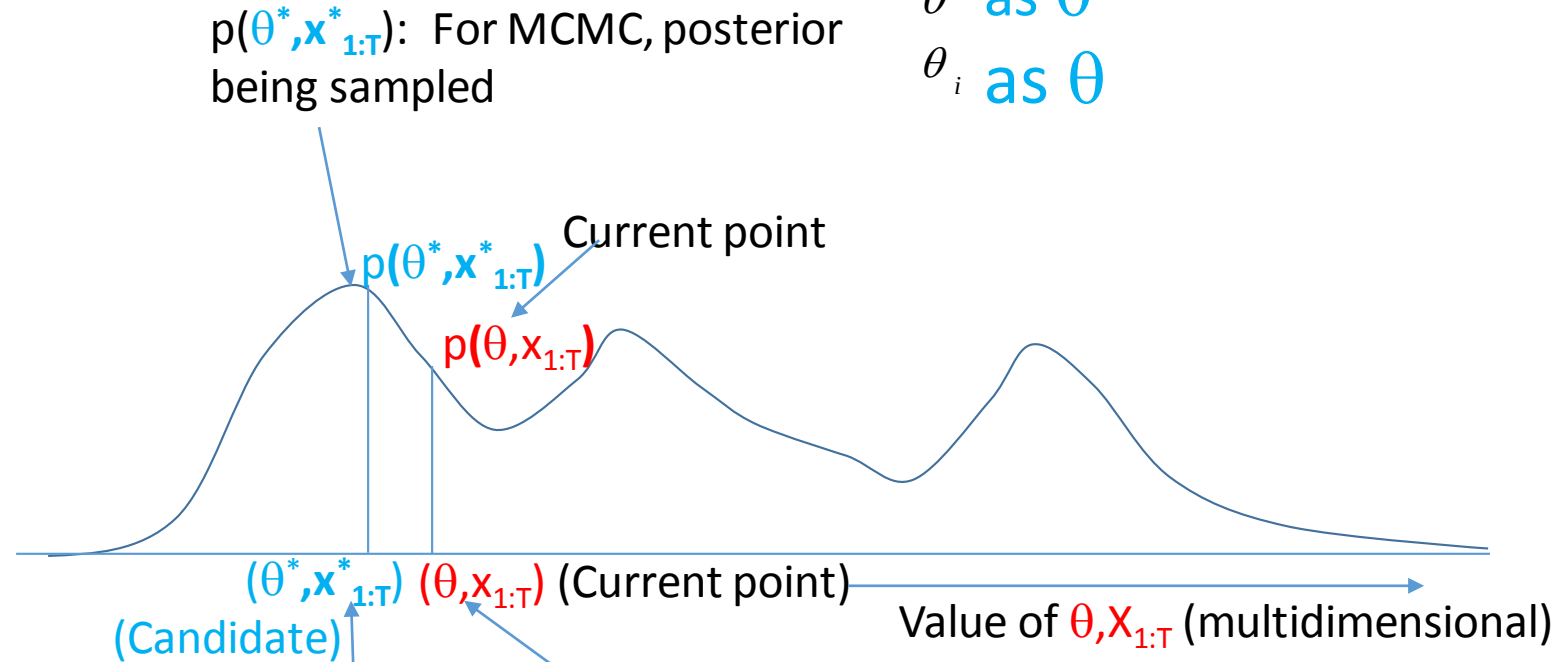
In the process of computing approx. to $p_{\theta^*}(y_{1:T})$, perform SMC

We accept and adopt as current sample (location) $(\theta^*, x_{1:T}^*)$ with probability $\min(1, \frac{p_{\theta^*}(y_{1:T})p(\theta^*)}{p_{\theta}(y_{1:T})p(\theta)})$
Otherwise, we re-emit $(\theta, x_{1:T})$ sampled during previous iteration

Elements of Why (Random Walk M-H) MCMC Works

For simplicity, we denote

θ as θ^*
 θ_i as θ



We move from $(\theta, \mathbf{x}_{1:T})$ to $(\theta^*, \mathbf{x}_{1:T}^*)$ with probability

$$\min\left(1, \frac{p_{\theta^*}(\mathbf{y}_{1:T})p(\theta^*)}{p_{\theta}(\mathbf{y}_{1:T})p(\theta)}\right)$$

NB: Even if $p(\theta^*, \mathbf{x}_{1:T}^*) < p(\theta, \mathbf{x}_{1:T})$,
May still transition current pos. from $\theta, \mathbf{x}_{1:T}$ to (candidate) $(\theta^*, \mathbf{x}_{1:T}^*)$
 $\Rightarrow$ aids repeated explore entire distribution

Performing Particle Filtering

- To sample from a trajectory of latent variables $x_{1:T}$ for a given θ and calculate estimate $\hat{p}_{\theta}(y_{1:T})$ of $p_{\theta}(y_{1:T})$: use SMC (particle filtering)
- Perform particle filtering, resampling at every observation, accum. ancestry info
- We compute the value of $p_{\theta}(y_{1:T})$ marginal distribution using weight and state information accumulated across **all** particles over all timesteps

$$\hat{p}_{\theta}(y_{1:T}) = \hat{p}_{\theta}(y_1) \prod_{n=2}^T \hat{p}_{\theta}(y_n | y_{1:(n-1)})$$

$$\hat{p}_{\theta}(y_n | y_{1:(n-1)}) = \frac{1}{N} \sum_{k=1}^N w_k \left(X_{1:n}^k \right)$$


- This equation takes an average (over all of the particles) of the weights obtaining in the model at a certain point in time n – this equation could in fact be computed incrementally, with
 - Full weights remembered for the current time only
 - With just this summary of weights remembered across all earlier timepoints

Sampling Trajectories

- Sampling a trajectory involves sampling from a full path of state of the model over time
- There is a marked difference between sampling cross-sectionally, and sampling longitudinally (via trajectories)
 - Sampling particles cross-sectionally at a certain time t chooses a particle (at time t) based on a weight informed by data observed up to the current point (in light of model structure)
 - Sampling an entire trajectory will choose a particle at time t based on a weight at a later time (e.g., following the most recent observation) – this chooses the trajectory based on a weight informed by all data observed up to the current point (in light of model structure) – typically including much data after time t !
- *Sampling from trajectories lets us infer what was going on at an earlier time on the basis of hindsight – i.e., in light of much data since that point*
 - For example, if there was a large outbreak shortly following time t , it may strongly suggest that there remained a high level of susceptibles at time t .

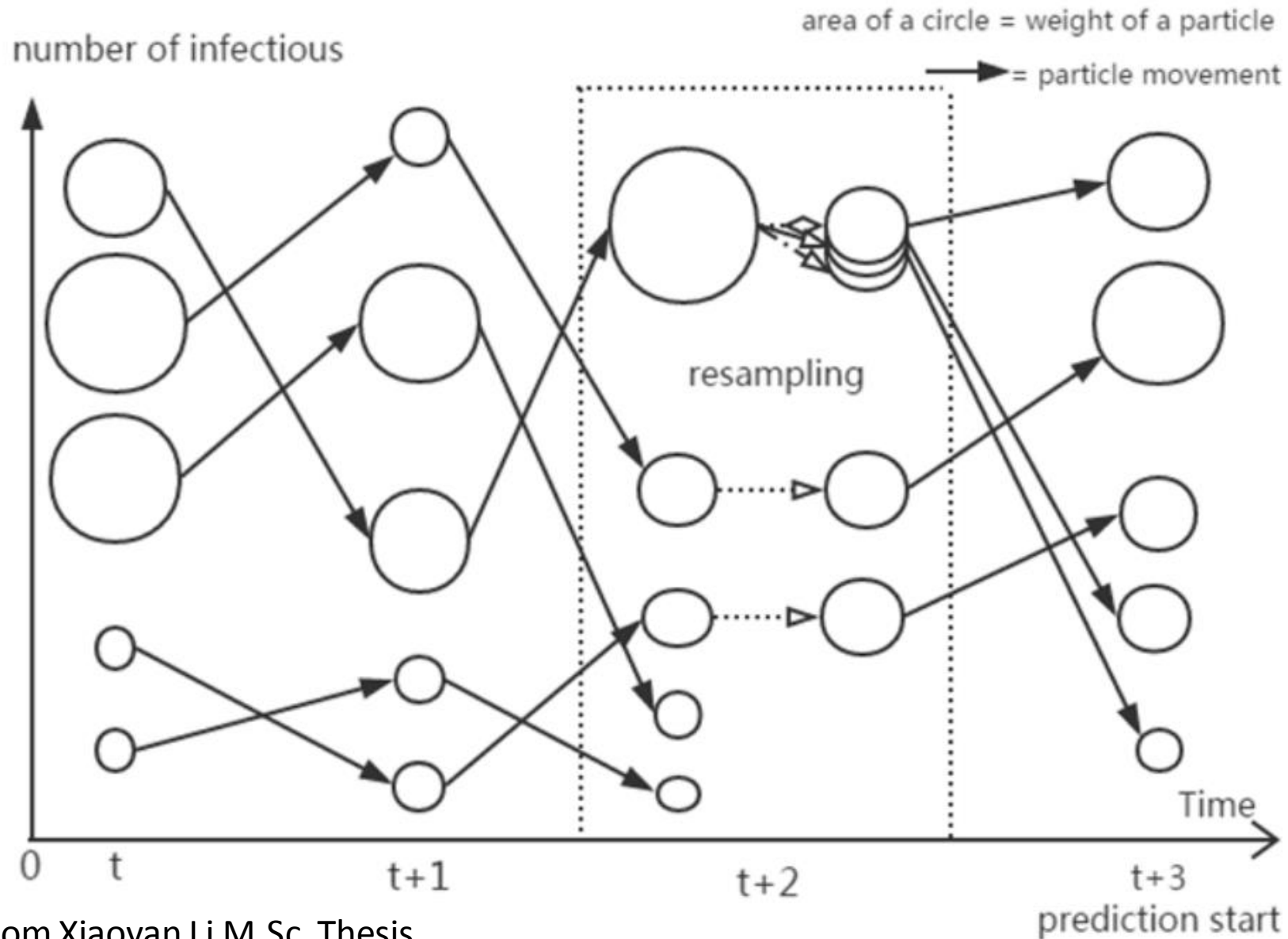
Sampling the Trajectories

- Sampling from particles at current time samples from the posterior distribution for the joint state of the model in light of *all data to present*
- As with HMMs
 - When operating *retrospectively*, we can sharpen our estimate as to what was going on back at an earlier time by *considering data received since that time*
 - We can reconstruct full *trajectories* of states following measurement points by keeping track of the “lineage” (ancestry) of a particle
- The particularly powerful use of particle filtering – and PMCMC – is to sample such trajectories
- We sample trajectories by sampling the particles (by weight) at the ***final timepoint***

Tracking Lineage: The Ancestry Matrices

- In order to sample trajectories, we construct an “ancestry matrix” that records the index of the ancestors of each particle at previous observation points
- We further need to keep track of the *value* of each particle at that earlier time
- A given particle will have a particular lineage – its ancestors were located at certain indices at points in the past (prior to particular observation points)
- Looking back, many current particles will often share common ancestor and lineages (typically higher weight at that time)

Tracking Linage: The Ancestry Matrices 2



Tracking Lineage: The Ancestry Matrices 2

- When we want to sample a trajectory, we sample a particle at that time, trace back successive ancestors, obtaining the value of each
- Maintaining ancestral information requires considerable extra space – but is often well worth it in terms of giving us a picture as to what was happening at earlier timepoints
- Sampled trajectories can reveal plausible hypotheses as to what has occurred over time

High Computational Cost

- Compared to other methods covered, PMCMC is accompanied by a very high computational cost
- We must perform a particle filter (likely with thousands of particles) for each MCMC iteration
- It is essential that PMCMC be performed in an efficient fashion
- There are many parallelizable elements of MCMC
 - Particle evolution between observation points
 - Distinct Monte Carlo walkers (chains)

Additional Components

- The probability of acceptance of a candidate MCMC step can be notably affected by the chance sampled trajectory following the associated particle filtering taking place
- This can cause a low acceptance rates
- Things that can elevate acceptance rates
 - Broader likelihood functions (not rejecting distant values)
 - Higher particle count (imposes less variability in particle filtering sampling results from trajectories)
- From our experience, effective PMCMC still requires quite technical programming

Additional Components

