

Particle Filtering & PMCMC for Integrating Wastewater Surveillance & Health System Data: Algorithmic & Mathematical Underpinnings and Pragmatics

Nathaniel Osgood

Agenda

- Retrospective
- Strengths & challenges of particle filtering for dynamic models
- Basic algorithmic specification
- Understanding the importance sampling context
- Evolving parameters & balancing model stochastics during particle filtering
- Understanding the likelihood functions
- Resampling specification
- Modeling structures to capture wastewater dynamics
 - Population shedding profile
 - Sampling from accumulations
 - Delays in wastewater propagation
 - Stormwater/wastewater coupling
- Handling presence/absence assays from wastewater
- Forecasting from present: Projecting particle-filtered models forward
- Evaluating particle filter performance
- PMCMC

Understanding of Particle Filtering with Dynamic Models

- Philosophy
- Intuition
- Concrete implementation
- Bayesian formulation
- Importance sampling

Additional Resources

- For Particle Filtering
 - Systematic walk-through of the math involved in particle filtering for dynamic models
 - Probability theory (at a distributional level) <https://youtu.be/lwEkfZTw2SU>
 - Importance sampling at the base of particle filtering https://youtu.be/bS1scuar_il
 - M.S. Arulampalam, S. Maskell, N. Gordon, & T. Clapp. A tutorial on particle filters for online nonlinear, non-Gaussian Bayesian tracking. IEEE Transactions on Signal Processing 50, 174–188, 2002.
 - Li X, Doroshenko A, Osgood ND (2018) Applying particle filtering in both aggregated and age-structured population compartmental models of pre-vaccination measles. PLoS ONE 13(11): e0206529. <https://doi.org/10.1371/journal.pone.0206529>
 - Kreier, F. 2021. The myriad ways sewage surveillance is helping fight COVID around the world. Nature. 10 May 2021. ISSN 1476-4687 <https://doi.org/10.1038/d41586-021-01234-1>
 - Li, X., 2018. Incorporating Particle Filtering and System Dynamic Modelling in Infection Transmission of Measles and Pertussis. M.Sc. Thesis. Department of Computer Science, University of Saskatchewan.
- For PMCMC
 - Andrieu C., Doucet A. Particle Markov chain Monte Carlo methods. J. R. Statist. Soc. B (2010) 72, Part 3, pp. 269–342
 - Use of MCMC with Dynamic models <https://youtu.be/F7WxC6hSXlc>
 - Intuition & use of PMCMC with Dynamic models <https://youtu.be/vJUWI3IJBpE>

Agenda

- Retrospective
- **Strengths & challenges of particle filtering for dynamic models**
- Basic algorithmic specification
- Understanding the importance sampling context
- Evolving parameters & balancing model stochastics during particle filtering
- Understanding the likelihood functions
- Resampling specification
- Modeling structures to capture wastewater dynamics
 - Population shedding profile
 - Sampling from accumulations
 - Delays in wastewater propagation
 - Stormwater/wastewater coupling
- Handling presence/absence assays from wastewater
- Forecasting from present: Projecting particle-filtered models forward
- Evaluating particle filter performance
- PMCMC

Particle Filtering with Dynamic Models: General Strengths

- Loose distributional assumptions (contrast: Kalman Filter)
- Estimation of joint probability distribution over evolving latent state (contrast: MLE estimate)
- No reliance on functional form/linearization of state equations
- Much value even when each data point is highly ambiguous
- Can simultaneously incorporate diverse observation types
- Exceptionally well suited to exploit streaming high-velocity data
- Ability to trivially project forward, ask “what if” questions

PF/PMCMC with Dynamic Models: Strengths for Wastewater

- Ready use of Particle Filtering with only sparse wastewater data
- Capacity for application at diverse levels (facilities, neighbourhood, municipal, SAG, etc.)
- Option to make use of cheaper, passive sampling over periods of time
- Potential to take advantage of both
 - Moderate-sensitivity quantitation
 - Higher-sensitivity (e.g., variant specific) dichotomous presence/absence data

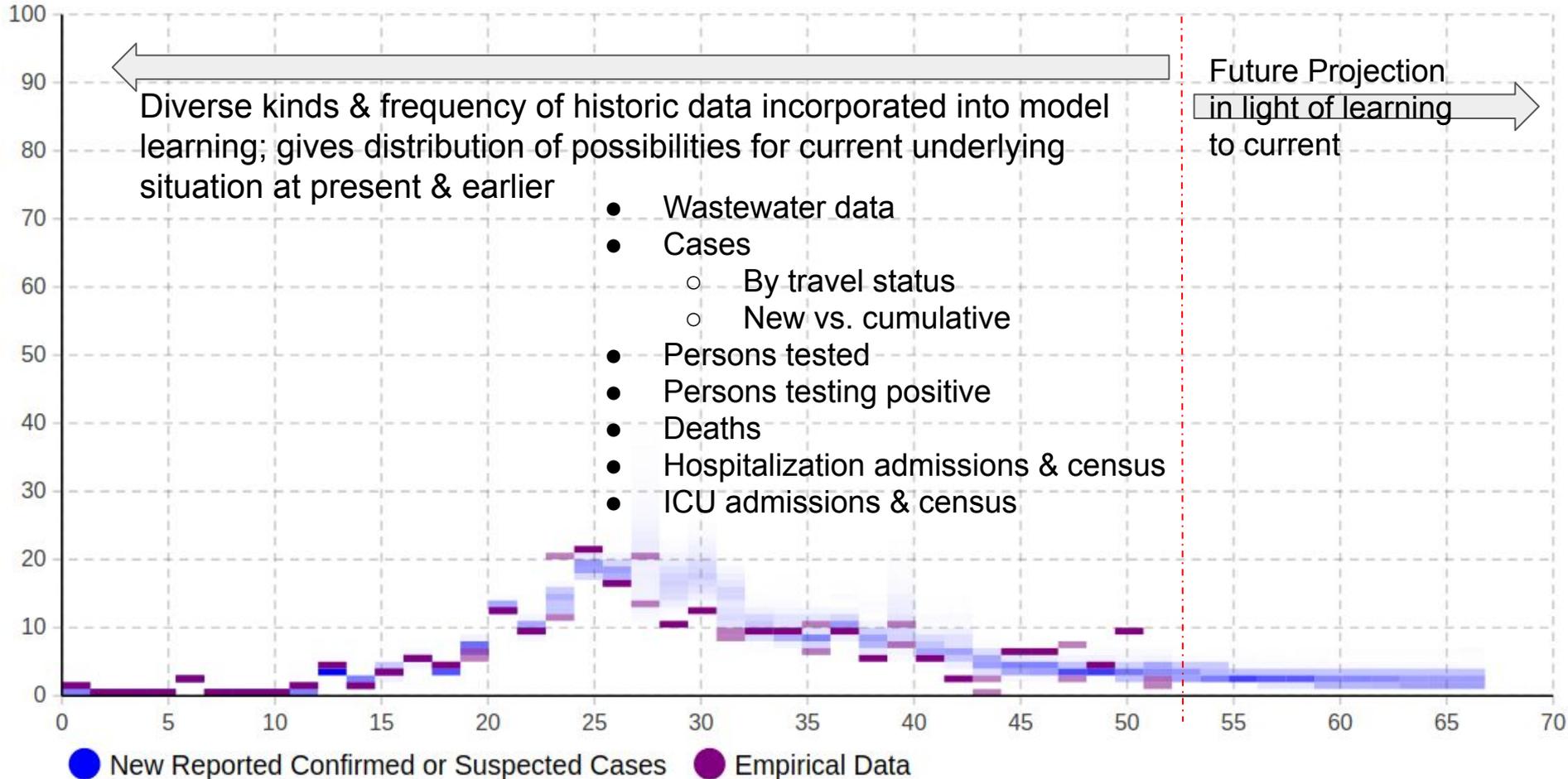
Particle Filtering Challenges

- Potential requirement of large number of particles
- Tuning required for appropriate balance of
 - Hubris: Model overconfidence & rigidness driven by too few stochastics
 - Underconfidence: Model precipitously alternates between interpretations of situation according to the latest data
- Particle impoverishment: Particle diversity becomes too small
- Risk that condensation algorithm is too naïve
 - e.g., if new data is received very frequently compared to how quickly the growth in model-related uncertainty

Agenda

- Retrospective
- Strengths & challenges of particle filtering for dynamic models
- **Basic algorithmic specification**
- Understanding the importance sampling context
- Evolving parameters & balancing model stochastics during particle filtering
- Understanding the likelihood functions
- Resampling specification
- Modeling structures to capture wastewater dynamics
 - Population shedding profile
 - Sampling from accumulations
 - Delays in wastewater propagation
 - Stormwater/wastewater coupling
- Handling presence/absence assays from wastewater
- Forecasting from present: Projecting particle-filtered models forward
- Evaluating particle filter performance
- PMCMC

Adaptive Planning -- Observing Unfolding Evidence



Bayesian Machine Learning & Dynamic Models

- **MCMC:** Sample from $p_M(\theta|y_{1:T})$: posteriors of *deterministic* dynamic model static parameters, latent states, scenario results, and incremental scenario gains.
- **Particle Filtering/SMC:** Sample from $p_{\theta,M}(x_{1:T}|y_{1:T})$: posteriors of *stochastic* dynamic model latent states stochastically evolving parameters, scenario results, and incremental scenario gains.
- **Particle MCMC (PMCMC):** Sample from $p_M(\theta, x_{1:T}|y_{1:T})$: posteriors of *stochastic* dynamic model latent states, stochastically evolving parameters, scenario results, and incremental scenario gains *and static parameters*.

Key Facts About how PF Works

The simulation model ...

- Includes ***stochastic processes***

- Runs typically during prediction step between observation points

- Entire state “corrected” to align w/empirical data after observation

Is performed ***recursively***

- Rather than re-estimate the state over all time points *de novo* when new data comes in, the estimate when new data comes in depends on earlier estimates

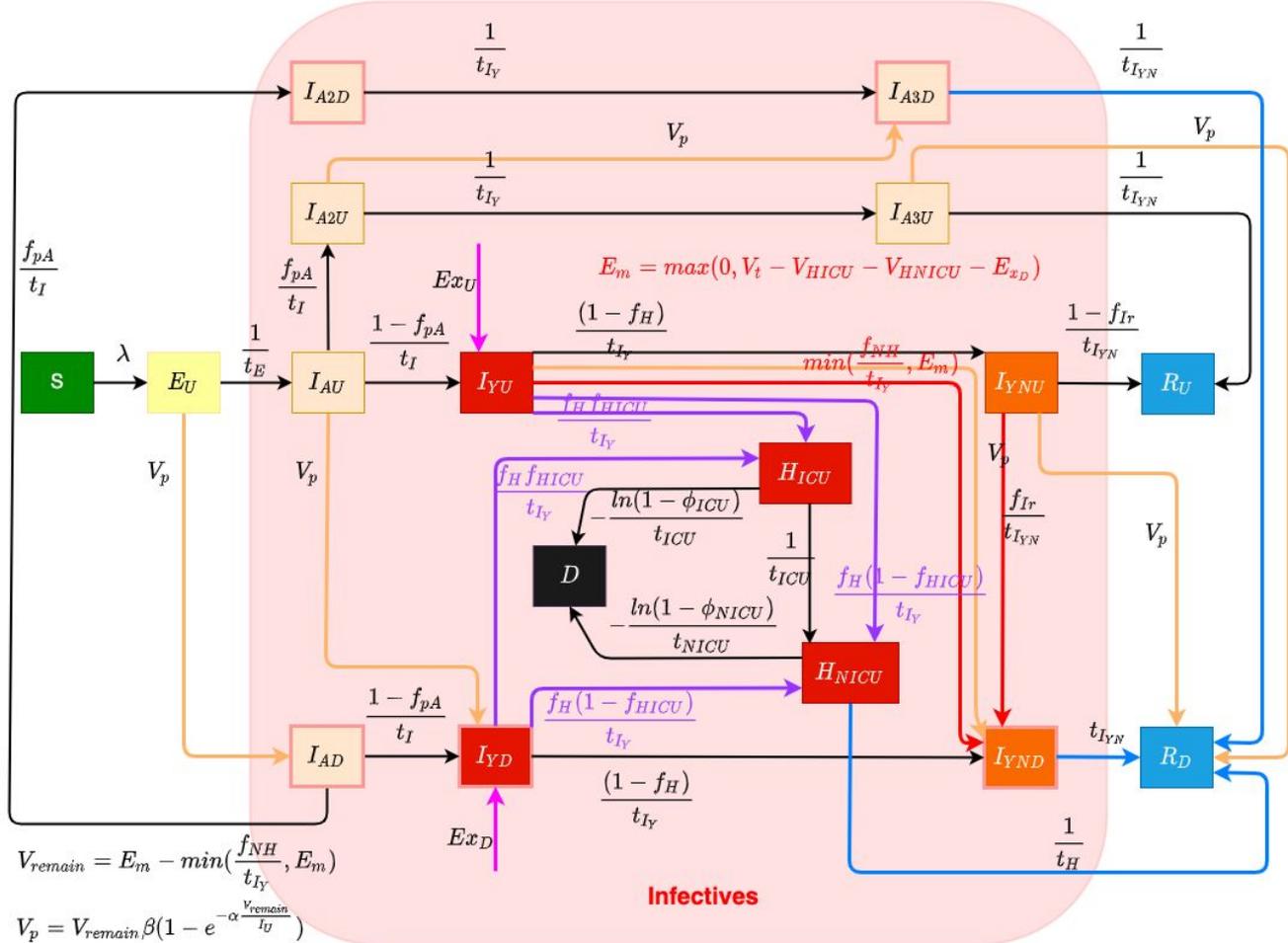
Samples from the state (and trajectory) distribution

- Each sample represented by “particle”** & contains a **complete hypothesis about state**

- Particles reflect “competing hypotheses” as to value of the current state

- Exploits **importance sampling**: distribution is sampled by associating samples from proposal dist (particles) w/**weights**

- There is a “**survival of the fittest**” of particles (hypotheses)



At any one time, each particle would hold a specific value for each state variable & is characterized by state vector

$$[S, E_U, E_D, I_{AU}, I_{AD}, I_{A2U}, I_{A2D}, I_{A3U}, I_{A3D}, I_{AU}, I_{AD}, I_{YU}, I_{YD}, I_{YNU}, I_{YND}, H_{ICU}, H_{NICU}, R_U, R_D, D, C_\beta, \alpha, f_{NH}, f_H]^T$$

How to Perform PF on Agg. Model in a Nutshell

Start with stochastic System Dynamics/Compartmental/ODE model

Subscript model by 100s to 1000s of particles

Each particle has its own full copy of model state (anything that could differ b/t realizations)

Sample from initial model state from prior distribution; set weights uniformly to 1

(Prediction phase): **Between observations**

All particles evolve according to standard model dynamics (just perform integration of each particle's state until next observation; all particles survive)

Particle weights remain invariant

(Update phase): **At observation points** (daily for SK data considered here): For each particle, multiply particle weight by likelihood of observing the empirical observation vector, given particle state

Resampling/“Survival of the fittest”: If effective sample size is too low (too much disparity in weights) following observation, particles are resampled according to their weights, and weight is reset to 1

Particles with high weights reproduce; with low weights disappear

Trajectories can be sampled by maintaining *ancestry matrix holding lineages*

1. At time $k=0$:

- (1) Sample $X_0^{N(i)}$ from $q_0(x_0^N)$;
- (2) Compute a *weight* for each particle $w_0^{(i)} = \frac{1}{N_s}$. It indicates that the weight at initial time step follows uniform distribution.

For a dynamic model, this just involves running the model forward from the previous timepoint

2. At time $k \geq 1$, perform a recursive update as follows:

- (1) Advance the sampled state by sampling $X_k^{N(i)} \sim q_k(x_k^N | y_k, X_{0:k-1}^{N(i)})$ and set $X_{0:k}^{N(i)} = (X_{0:k-1}^{N(i)}, X_k^{N(i)})$;
- (2) Update the weights to reflect the probabilistic and state update models as follows:

Likelihood function

$$w_k^{(i)} = W_{k-1}^{(i)} \frac{p(y_k^M | X_k^{N(i)}) p(X_k^{N(i)} | X_{k-1}^{N(i)})}{q(X_k^{N(i)} | X_{k-1}^{N(i)}, y_k^M)}$$

With the condensation algorithm we choose these two terms to be identical & they cancel

Image from Li, X., 2018. Incorporating Particle Filtering and System Dynamic Modelling in Infection Transmission of Measles and Pertussis. M.Sc. Thesis. Department of Computer Science, University of Saskatchewan.

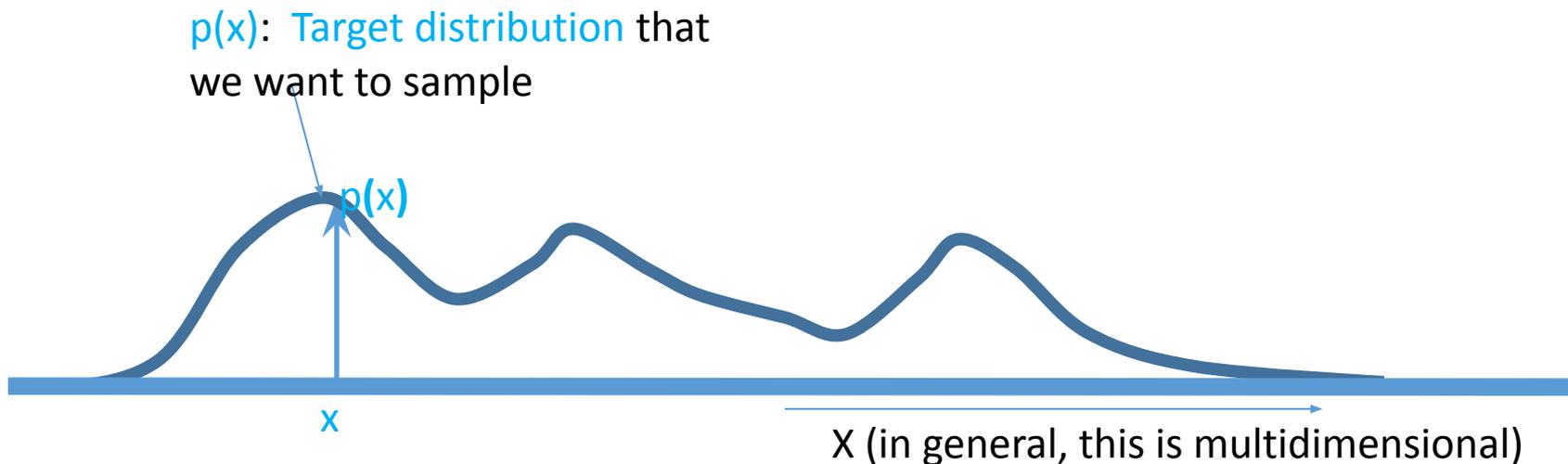
Normalize the weights $W_k^{(i)} = \frac{w_k^{(i)}}{\sum_{i=1}^{N_s} w_k^{(i)}}$

Agenda

- Retrospective
- Strengths & challenges of particle filtering for dynamic models
- Basic algorithmic specification
- **Understanding the importance sampling context**
- Understanding the likelihood functions
- Resampling specification
- Modeling structures to capture wastewater dynamics
 - Population shedding profile
 - Sampling from accumulations
 - Delays in wastewater propagation
 - Stormwater/wastewater coupling
- Handling presence/absence assays from wastewater
- Forecasting from present: Projecting particle-filtered models forward
- Evaluating particle filter performance
- PMCMC

Motivating Reflection: How to Sample from a Fully Specified Distribution?

Suppose that we want to draw samples x from a “target” distribution that given an x , has a probability (density) value $p(x)$ that can be readily computed



Central Role of Importance Sampling

- Particle Filtering uses another means of sampling – sequential importance sampling
- Here, we forsake the goal of sampling directly from the ideal (“target”) distribution via MCMC
- Rather, we use the importance sampling technique of
 - Sampling readily from a different “proposal distribution”
 - Using weighting of the samples from the proposal distribution to characterize their relative representation in the target relative to the proposal distribution
 - This weight allows us to sample more frequently from those samples that are well represented in the target distribution, and less frequently from others

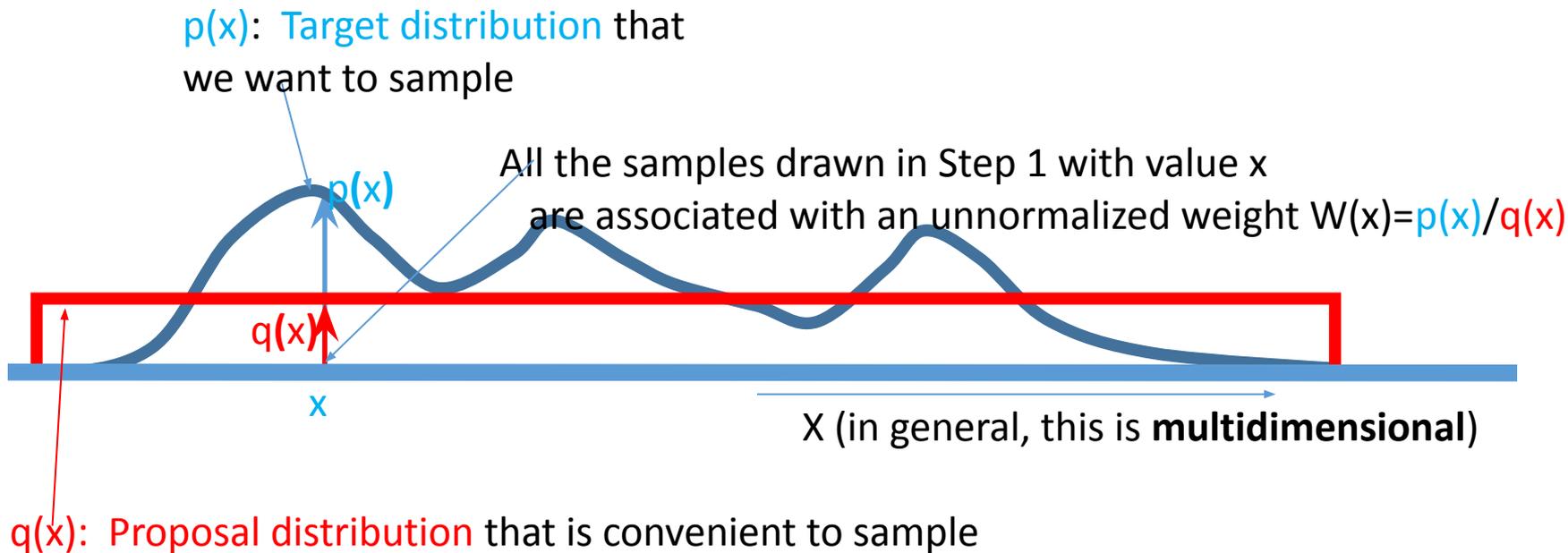
Importance Sampling 1

- Suppose that we want to draw samples x from a “target” distribution $p(x)$ that
 - Is difficult to sample from directly
 - Given an x , has a probability (density) value that can be readily computed
- Suppose that we have another “proposal” distribution $q(x)$ from which we can readily sample
 - Common example: Uniform distribution
 - Ideally this would be something like $p(x)$

Importance Sampling 2

- We can readily sample N values x from $p(x)$ by a 4 part procedure:
 - Part 1: Create a set S of N sample values x_i ($1 \leq i \leq N$) from $q(x)$
 - e.g., draw each x_i from a uniform distribution between 0 and 1
 - Part 2: Label each drawn value x_i with a “weight” $W_i = p(x_i)/q(x_i)$
 - This “weight” expresses how much more common x_i is within the target distribution $p(x)$, when compared to the distribution from which it was drawn, $q(x)$
 - Part 3: Normalize weights, labeling each x_i with weight $w_i = \frac{W_i}{\sum_{j=1}^N W_j}$
 - Part 4: Draw N samples from S , where the probability of drawing sample i each is proportional to w_i
 - For each such sample, this is readily performed by
 - Drawing a value u from $[0,1]$
 - Going through each $1 \leq i \leq N$ accumulating the value of w_i until the smallest i where $\sum_{j=1}^i W_j \geq u$

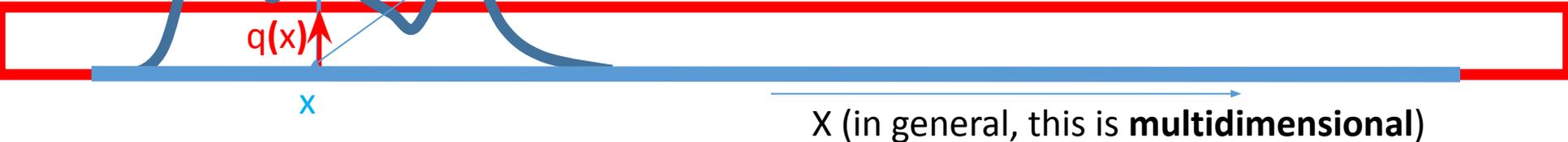
Importance Sampling



Importance Sampling: Poor Match of Proposal Distribution to Target

$p(x)$: Target distribution that we want to sample

All the samples drawn in Step 1 with value x are associated with an unnormalized weight $W(x) = p(x)/q(x)$



1. At time $k=0$:

(1) Sample $X_0^{N(i)}$ from $q_0(x_0^N)$;

(2) Compute a *weight* for each particle $w_0^{(i)} = \frac{1}{N_s}$. It indicates that the weight at initial time step follows uniform distribution.

These are importance weights -- whenever we want to use information from (e.g., compute statistics on) the particles, we need to draw from the particles with a probability given by W_k

2. At time $k \geq 1$, perform a recursive update as follows:

(1) Advance the sampled state by sampling $X_k^{N(i)} \sim q_k(x_k^N | y_k, X_{0:k-1}^{N(i)})$ and set $X_{0:k}^{N(i)} = (X_{0:k-1}^{N(i)}, X_k^{N(i)})$;

For a dynamic model, this just involves running the model forward from the previous timepoint

(2) Update the weights to reflect the probabilistic and state update models as follows:

Likelihood function

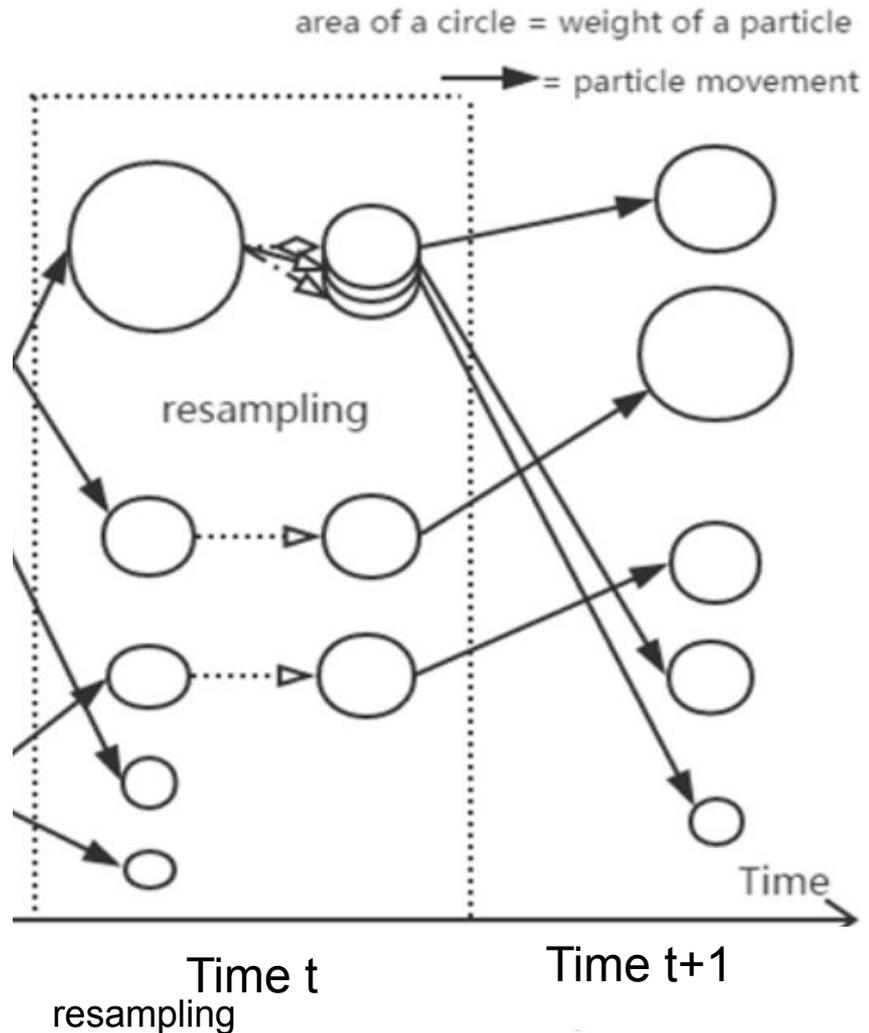
$$w_k^{(i)} = W_{k-1}^{(i)} \frac{p(y_k^M | X_k^{N(i)})p(X_k^{N(i)} | X_{k-1}^{N(i)})}{q(X_k^{N(i)} | X_{k-1}^{N(i)}, y_k^M)}$$

With the condensation algorithm we choose these two terms to be identical & they cancel

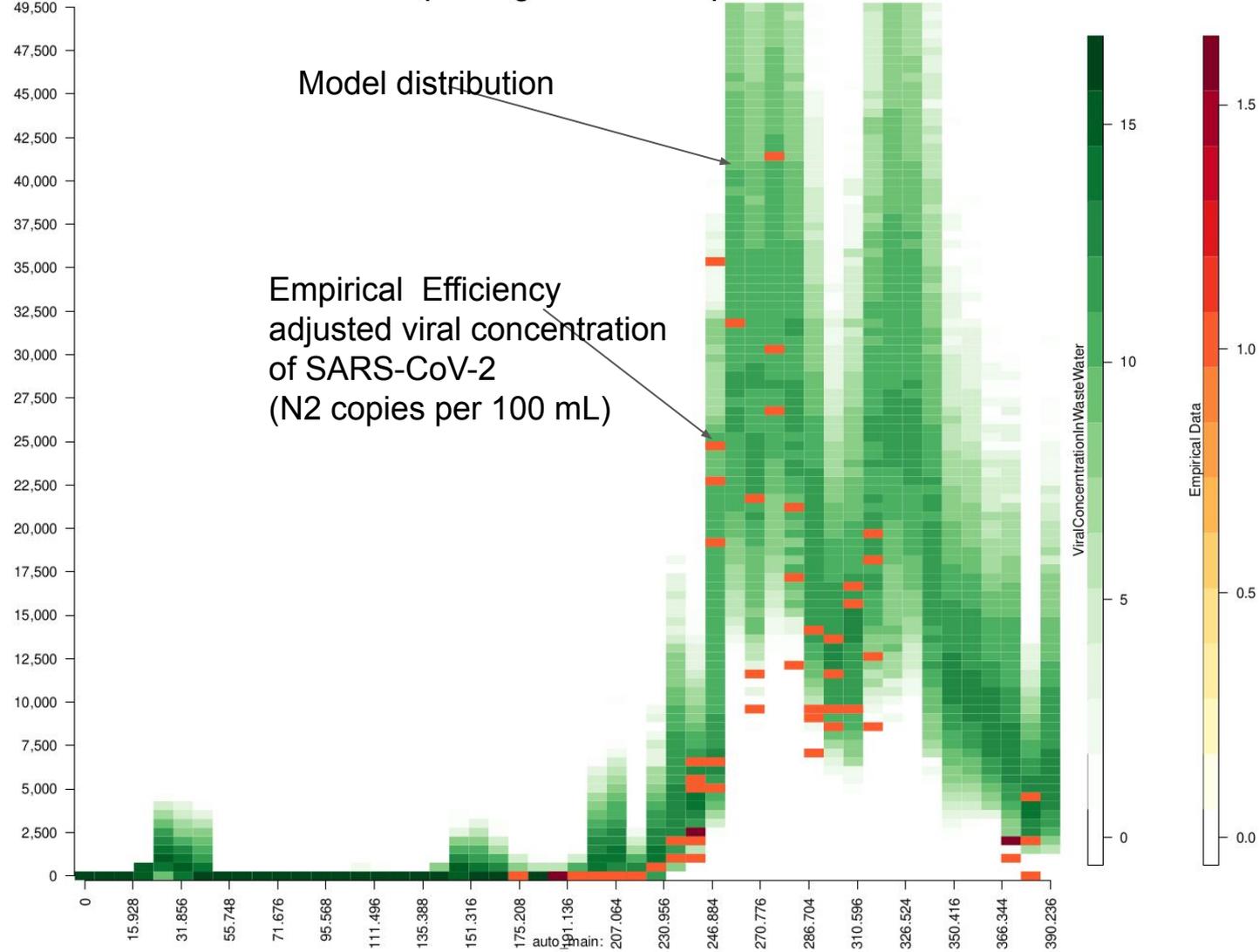
Image from Li, X., 2018. Incorporating Particle Filtering and System Dynamic Modelling in Infection Transmission of Measles and Pertussis. M.Sc. Thesis. Department of Computer Science, University of Saskatchewan.

Normalize the weights $W_k^{(i)} = \frac{w_k^{(i)}}{\sum_{i=1}^{N_s} w_k^{(i)}}$

Resampling
builds the next
generation by
(importance)
sampling from the
particles

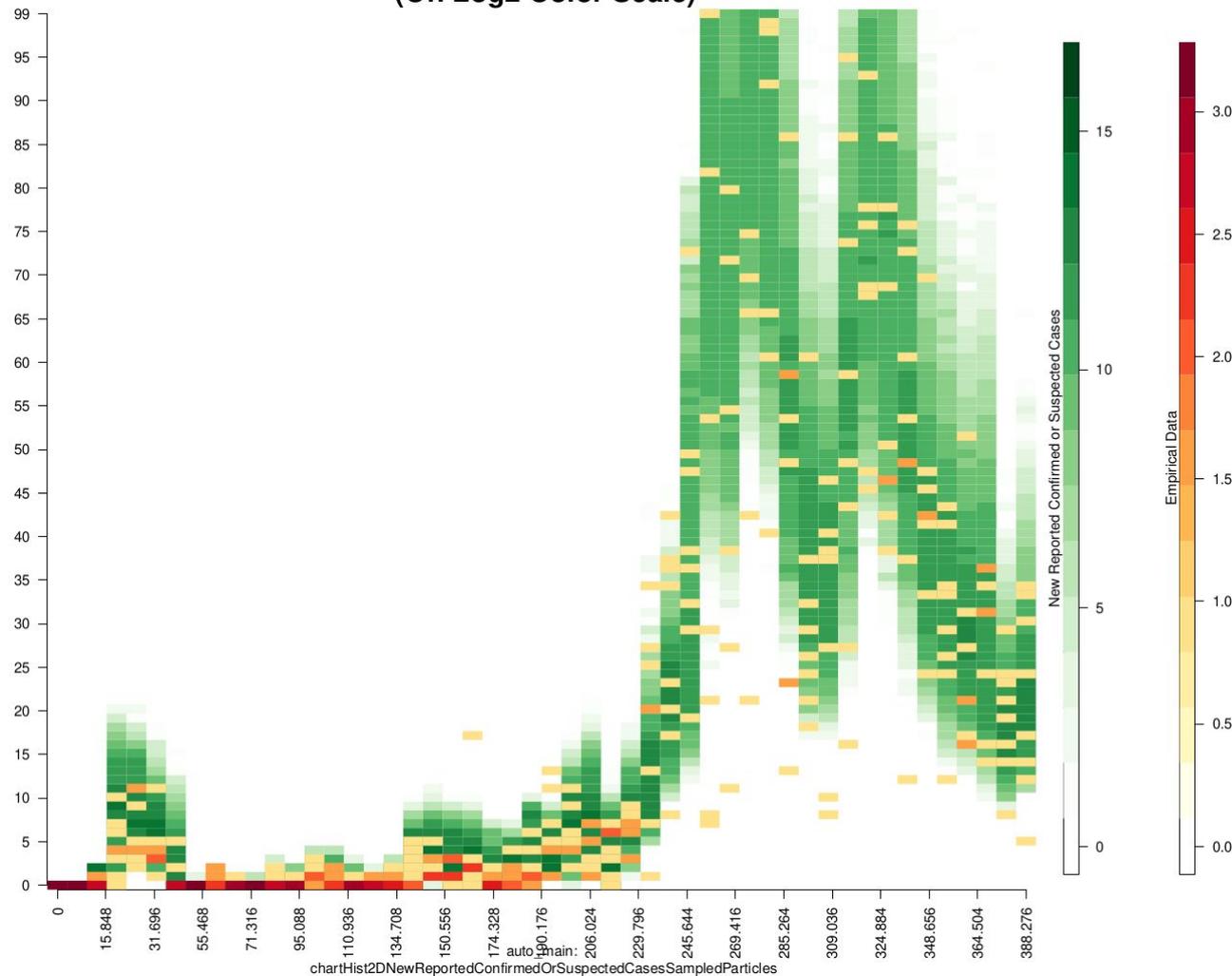


This distribution from the model is importance sampled from particles



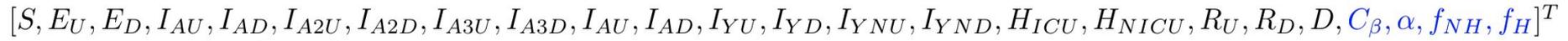
This distribution from the model is importance sampled from particles

New Reported Confirmed or Suspected Cases
(On Log2 Color Scale)



State Vector Associated with Each Particle

$[S, E_U, E_D, I_{AU}, I_{AD}, I_{A2U}, I_{A2D}, I_{A3U}, I_{A3D}, I_{AU}, I_{AD}, I_{YU}, I_{YD}, I_{YNU}, I_{YND}, H_{ICU}, H_{NICU}, R_U, R_D, D, C_\beta, \alpha, f_{NH}, f_H]^T$



Regular state variables of the System Dynamics/ODE/
Compartmental model

“Dynamic Parameters” whose (potentially transformed) values
are engaged in random walks over time

Agenda

- Retrospective
- Strengths & challenges of particle filtering for dynamic models
- Basic algorithmic specification
- **Evolving parameters & balancing model stochastics during particle filtering**
- Understanding the likelihood functions
- Resampling specification
- Modeling structures to capture wastewater dynamics
 - Population shedding profile
 - Sampling from accumulations
 - Delays in wastewater propagation
 - Stormwater/wastewater coupling
- Handling presence/absence assays from wastewater
- Forecasting from present: Projecting particle-filtered models forward
- Evaluating particle filter performance
- PMCMC

Stochastics: A Key Balance

- We need some stochastics in the model, or else all particles cloned during resampling will evolve identically, with no divergence
- Avoid overconfidence: We require enough model stochastics to allow the model to have a requisite variety to match a wide variety of different data
 - Too narrow a distribution will lead to “overconfidence” in model predictions – will not be as open to correction by “surprising” data
- Avoid underconfidence: We don’t want the model to have such pronounced stochastics that, absent data, it quickly becomes hopelessly uncertain

Two common means of introducing stochastics

- Evolving parameter as state variables, allowing them or transforms to evolve via random walks over time (e.g., reporting rate, contact rate)
- Stochastics in processes (e.g., a distribution of count of people infected, around the mean)

Tuning Model Confidence

- Particle count
- Dispersion-related parameters for likelihood distribution
- Random walks
 - Appropriate set
 - Appropriate standard deviation/volatility

Stochastically Evolving State Variables

The following variables exhibit **random walks** of variable, transformed to lie within a bounded range

- c : Contacts Per Day ($c\beta$ traditionally)
- f_H : Fraction of underlying infections that require hospitalization
- f_{NH} : Fraction of non-hospitalized that are reported as cases
- α : Efficiency of active testing

These random walks are bounded to lie within ranges

Character of Model Stochastics

- Sometimes such stochastics characterize particular known stochastic processes (e.g., evolution of reporting or contact rates)
- Sometimes stochastics seem to play mostly an instrumental role in achieving model “humility” (breadth of possibilities across particles) without characterizing specific known stochastic phenomena => We use to avoid model overconfidence

Evolving Parameters Undergo Random Walks

- Some relationships in the model can vary over time without drivers lying within the scope of the model. Common examples for communicable disease
 - Contact/transmission rate
 - Efficiency in active case finding
- Particle filtering supports inferring such values as they evolve
- The basic approach used is to
 - Implement such evolving parameters *within a given particle* as values undergoing **random walks**
 - Use the standard mechanisms of particle filtering to select for particles whose behaviour over time (as dictated by the model) is most consistent with the empirical data

Two Needs for Random Walks

- It is typical for us to have parameters that need to evolve only over certain defined ranges
 - Parameters with physical interpretation that imposes certain constraints (e.g., non-negativity)
 - Parameters empirically recognized as lying within a certain plausible range
- It is simplest to support random walks as Wiener processes over an unbounded range
- In order to perform the random walks, we need to map (transform) from a bounded to an unbounded range

Common Transformations for Random Walk

- **Non-negative values:** Perform random walk on log of quantity
- **Values bounded in range [0,1]:** Perform random walk on logit of value
- **Values bounded in range [a,b]:** Perform random walk on logit transform of normalized quantity:

$$\beta' = \frac{\beta - a}{b - a}$$

$$d(\text{logit}(\beta)) = d\left(\ln\left(\frac{\beta - a}{b - \beta}\right)\right) = s_{\beta} dW_t$$

Wiener Process characterizing an **unbounded** random walk

Implementing the Wiener Process

- During numerical integration of a given particle, we implement the Wiener Process for a state variable with a rate of change drawn from a normal distribution with mean 0 and specified **standard deviation**
- Reminder: state variable typically represents a **transformed** value of interest
- Tuning this **perturbation standard deviation** is one of the keys to effective tuning of the particle filtering via adjusting model stochastics

$$dX = s_{\beta} dW_t$$

$$\dot{X} = \frac{dX}{dt} = s_{\beta} N(0, 1) = N(0, s_{\beta})$$

Recall: How to Perform PF on Agg. Model in a Nutshell

Start with stochastic System Dynamics/Compartmental/ODE model

Subscript model by 1000s or more particles (COVID-19 provincial models: 150,000 particles is standard)

Each particle has its own **full copy of model state** (anything that could differ b/t realizations)

Sample from initial model state from prior distribution; set weights uniformly to 1

(Prediction phase): **Between observations**

All particles evolve according to standard model dynamics (just perform integration of each particle's state until next observation; all particles survive)

Particle weights remain invariant

(Update phase): **At observation points**: For each particle, multiply particle weight by likelihood of observing the empirical observation vector given that particle's state

Resampling/"**Survival of the fittest**": If effective sample size is too low (too much disparity in weights) following observation, particles are resampled according to their weights, and weight is reset to 1

Particles with high weights reproduce; those with low weights tend to disappear

Trajectories can be sampled by maintaining *ancestry matrix holding lineages*

1. At time $k=0$:

- (1) Sample $X_0^{N(i)}$ from $q_0(x_0^N)$;
- (2) Compute a *weight* for each particle $w_0^{(i)} = \frac{1}{N_s}$. It indicates that the weight at initial time step follows uniform distribution.

For a dynamic model, this just involves running the model forward from the previous timepoint

2. At time $k \geq 1$, perform a recursive update as follows:

- (1) Advance the sampled state by sampling $X_k^{N(i)} \sim q_k(x_k^N | y_k, X_{0:k-1}^{N(i)})$ and set $X_{0:k}^{N(i)} = (X_{0:k-1}^{N(i)}, X_k^{N(i)})$;
- (2) Update the weights to reflect the probabilistic and state update models as follows:

Likelihood function

$$w_k^{(i)} = W_{k-1}^{(i)} \frac{p(y_k^M | X_k^{N(i)}) p(X_k^{N(i)} | X_{k-1}^{N(i)})}{q(X_k^{N(i)} | X_{k-1}^{N(i)}, y_k^M)}$$

With the condensation algorithm we choose these two terms to be identical & they cancel

Image from Li, X., 2018. Incorporating Particle Filtering and System Dynamic Modelling in Infection Transmission of Measles and Pertussis. M.Sc. Thesis. Department of Computer Science, University of Saskatchewan.

Normalize the weights $W_k^{(i)} = \frac{w_k^{(i)}}{\sum_{i=1}^{N_s} w_k^{(i)}}$

Agenda

- Retrospective
- Strengths & challenges of particle filtering for dynamic models
- Basic algorithmic specification
- Evolving parameters & balancing model stochastics during particle filtering
- **Understanding the likelihood functions**
- Resampling specification
- Modeling structures to capture wastewater dynamics
 - Population shedding profile
 - Sampling from accumulations
 - Delays in wastewater propagation
 - Stormwater/wastewater coupling
- Handling presence/absence assays from wastewater
- Forecasting from present: Projecting particle-filtered models forward
- Evaluating particle filter performance
- PMCMC

Particle Filtering Requirements for Compartmental Model

- This approach can be readily applied to broad classes of ODE models
- Each type of data used must have corresponding element(s) in the dynamic model to which that data can be compared in a likelihood function, e.g.,
 - ICU/Non-ICU Hospital census: One or more corresponding state variable each
 - Reported cases: Diagnosis flows/transitions
 - ICU/Non-ICU hospital admissions: Hospitalization flows/transitions
 - Deaths: Hospitalization flows/transitions
 - Hospital admissions: Hospitalization flows/transitions
 - **Wastewater concentrations: One or more state variables characterizing shedding population**

Form of Likelihood Function 2: Dealing with Multiple Datasets

- Form when all data is present:

$$\begin{aligned} \mathcal{L} = & \mathcal{L}_{NewReportedEndogenousCases} \times \mathcal{L}_{CumulativeReportedEndogenousCases} \\ & \times \mathcal{L}_{CumulativeICUAdmissioncases} \times \mathcal{L}_{CumulativeNICUAdmissioncases} \\ & \times \mathcal{L}_{ICUCensusCases} \times \mathcal{L}_{NICUCensusCases} \\ & \times \mathcal{L}_{CumulativeDeathCases} \times \mathcal{L}_{ViralConcentration} \end{aligned}$$

- When any type of empirical data above is absent, that likelihood term is omitted from the calculation of the overall likelihood \mathcal{L}
- Wastewater is incorporated in the final term for those episodic days in which it is available

Likelihood Functions Explored

Likelihood functions $p(\mathbf{y}_t^M | \mathbf{x}_t^N)$ give the likelihood of the **empirical datum**, given the **particle state**

Common distributions

Binomial

Negative Binomial/Pascal

Poisson

Normal

Lognormal

Form of Likelihood Function 1

- Likelihood functions give the likelihood of the empirical datum, **given the model state for that particle**

Tempting: Binomial

$$y_t | \mathbf{x}_t^s \equiv y_t | i_t \sim \text{Binomial}(i_t, p)$$

Here, reports are some of i.i.d. Bernoulli draws, with reporting probability p

Difficulty: Stiff upper bound of i_t in reported count ($p(y_t | i_t) = 0$ for $i_t < y_t$) causes singularities when \forall particle s $i_t^s < y_t$

Selected: Negative Binomial (Pascal Distribution)

This $p(y_t | i_t)$ is non-zero for all $i_t > 0$

$$y_t | \mathbf{x}_t^s \equiv y_t | i_t \sim \text{NegativeBinomial}(i_t, r)$$

Form of Likelihood Function 2

$$y_t | \mathbf{X}_t^s \equiv y_t | i_t \sim \text{NegativeBinomial}(i_t, r)$$

$$\mathcal{L}(y_t | i_t) = \binom{y_t + r - 1}{r - 1} p^r (1 - p)^{y_t}$$

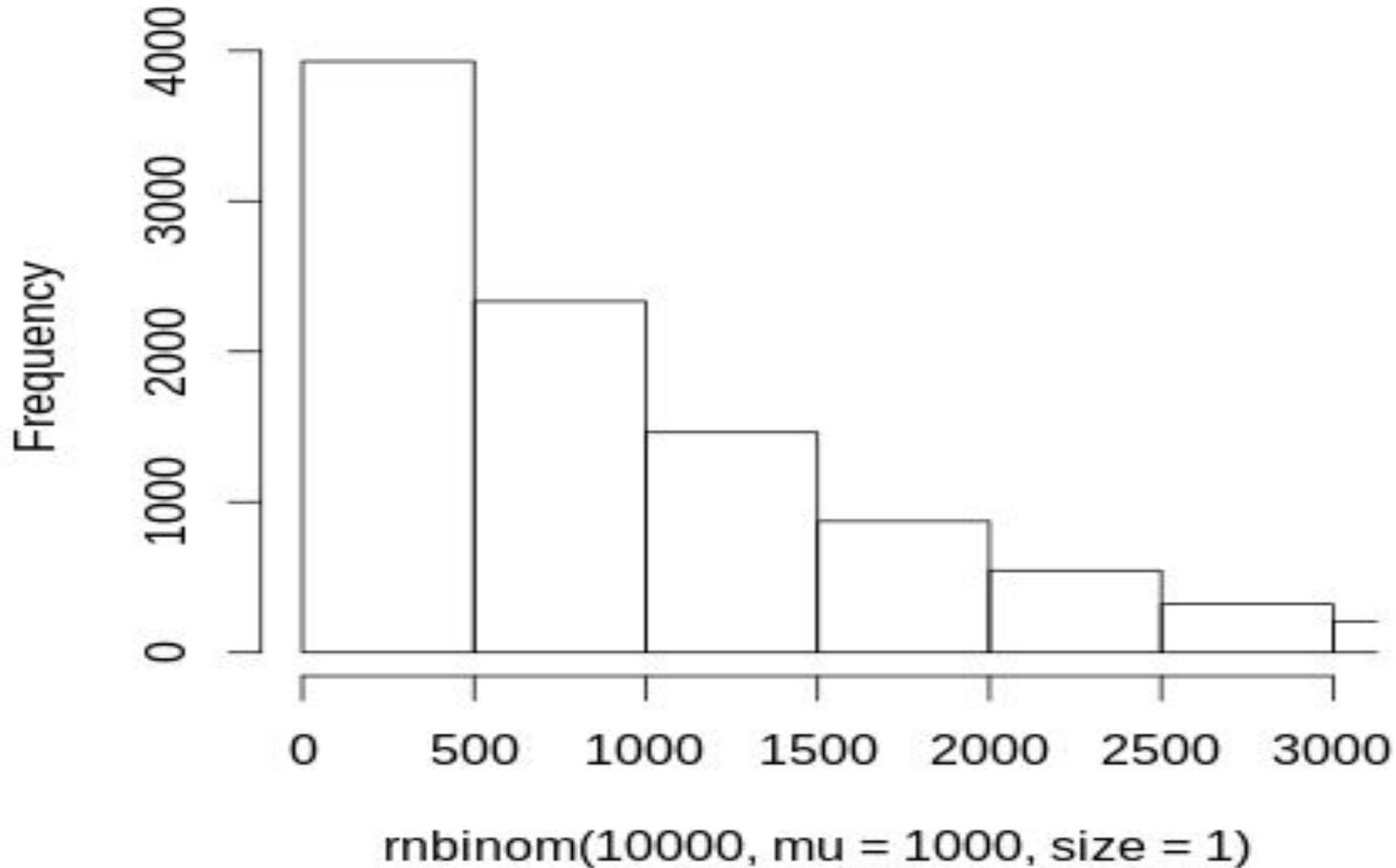
Observed cases from Empirical data

Underlying count of total cases of infectives from model

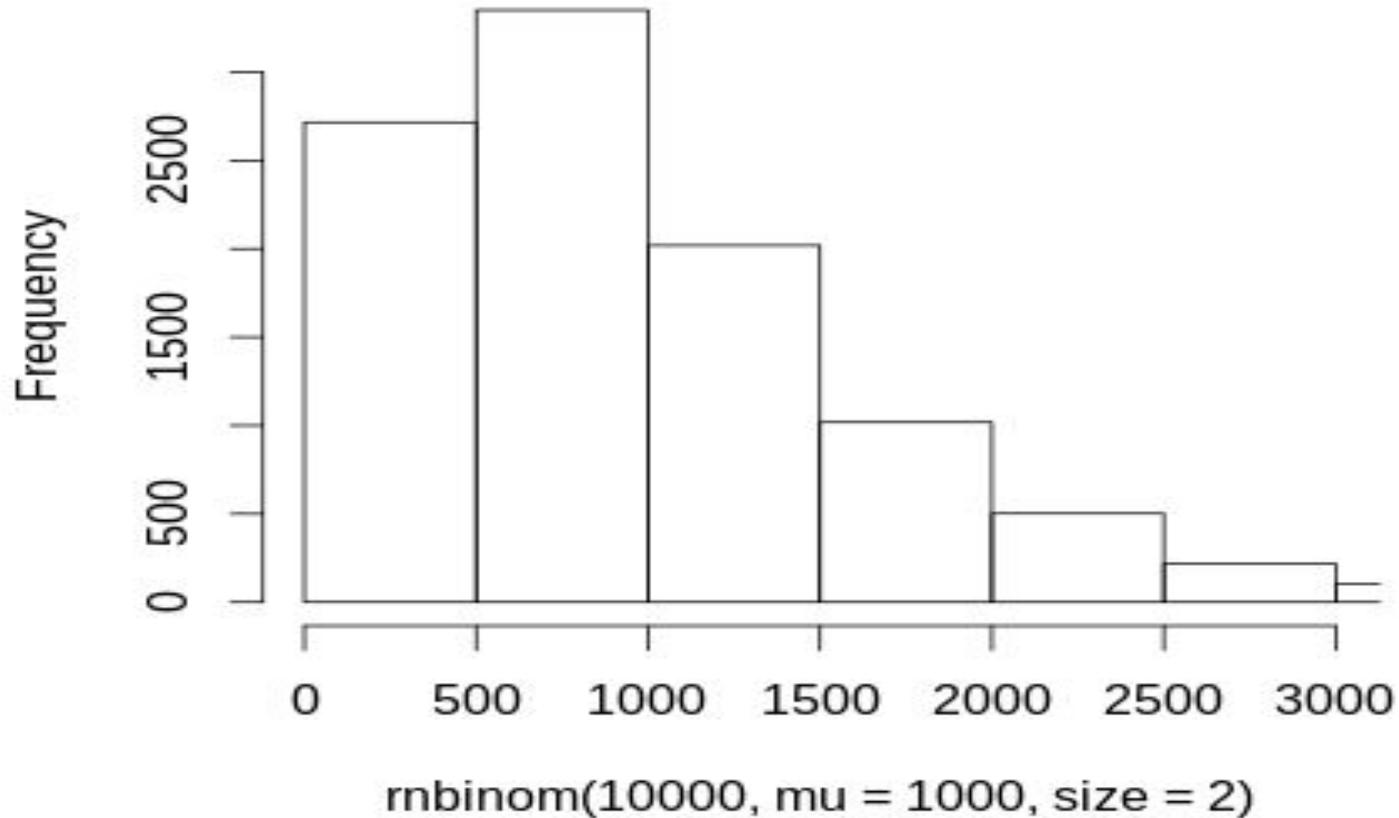
$$p = \frac{i_t}{i_t + r}$$

Dispersion Parameter (we choose!)

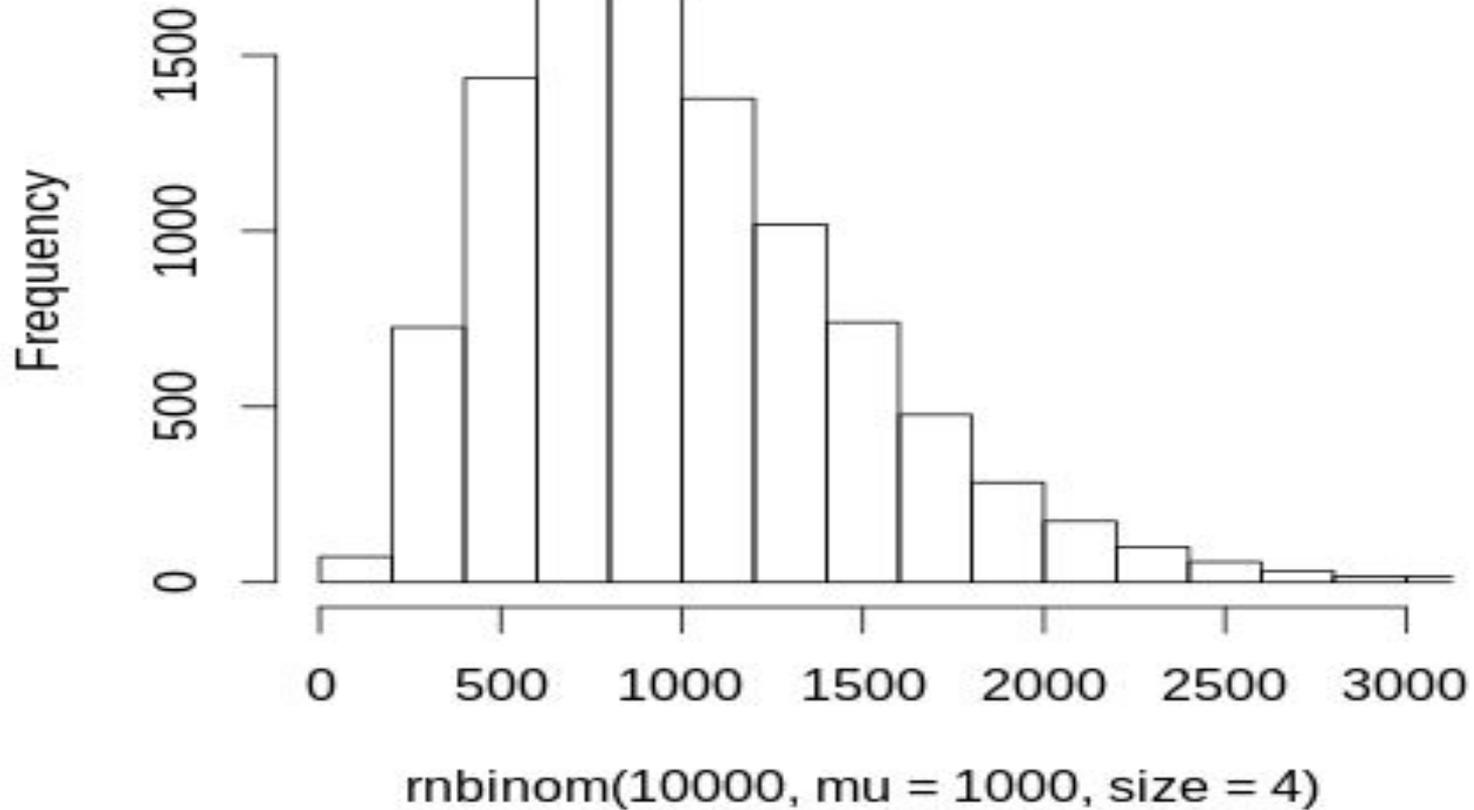
Example: Dispersion Parameter = 1



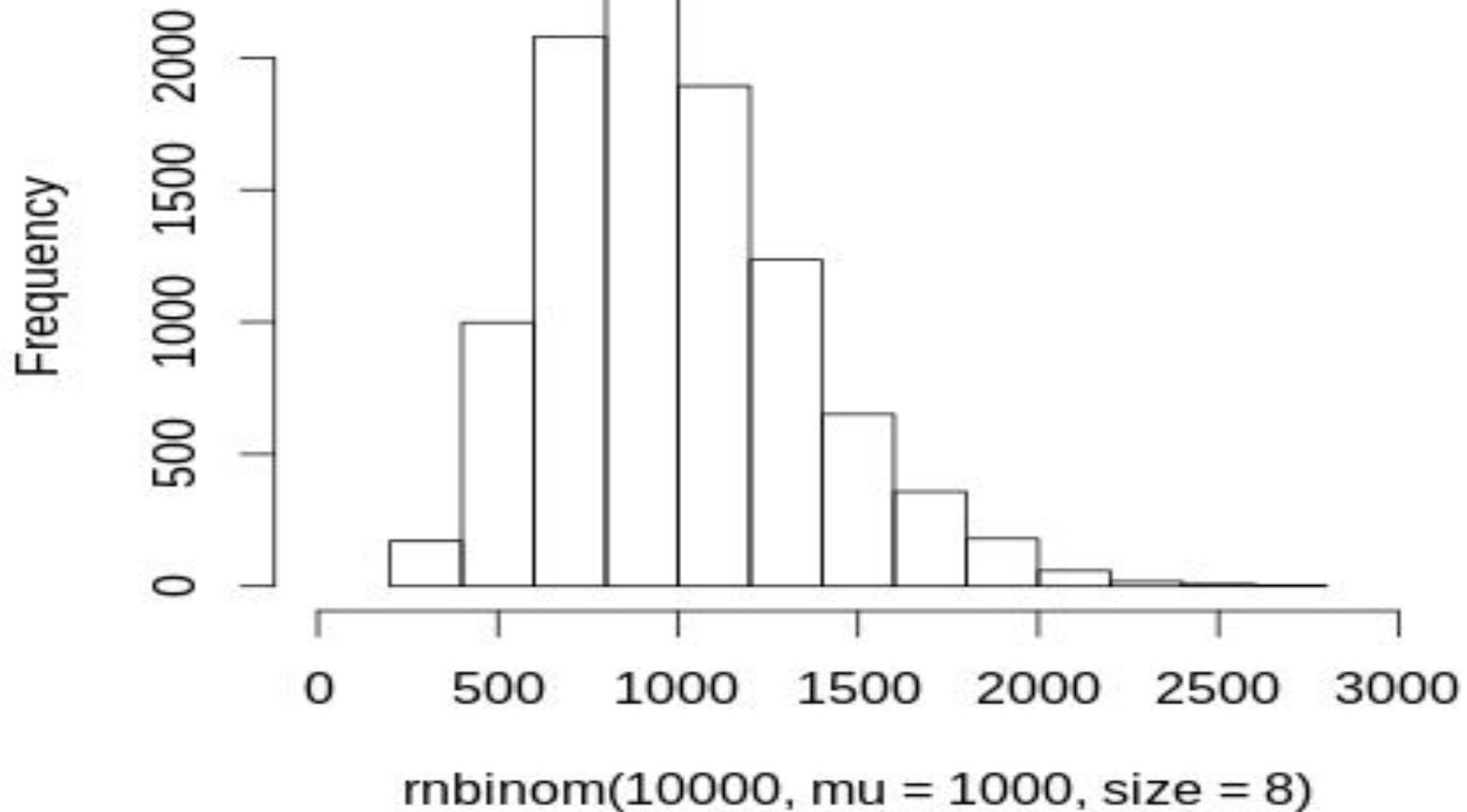
Dispersion Parameter = 2



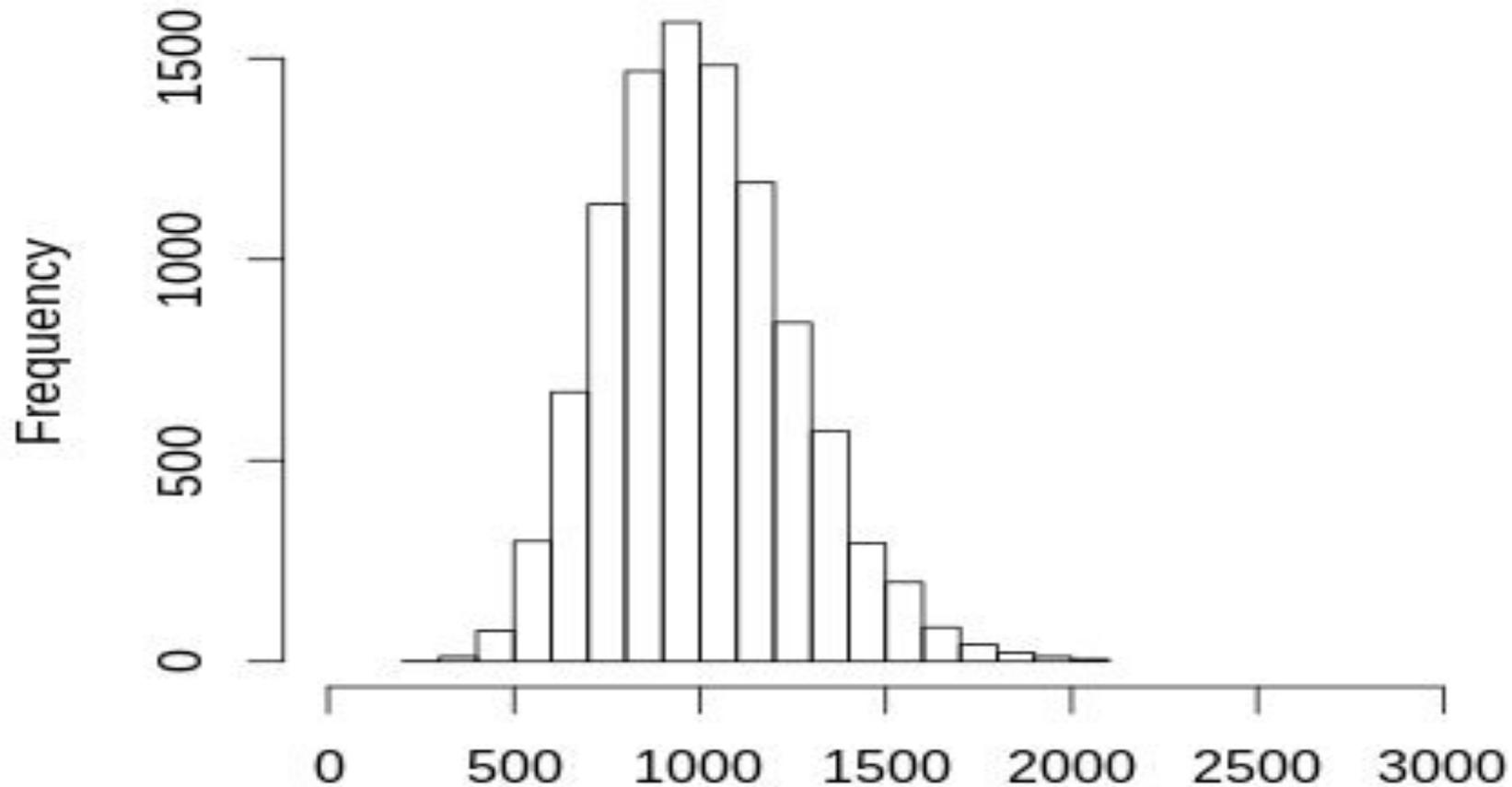
Dispersion Parameter = 4



Dispersion Parameter = 8

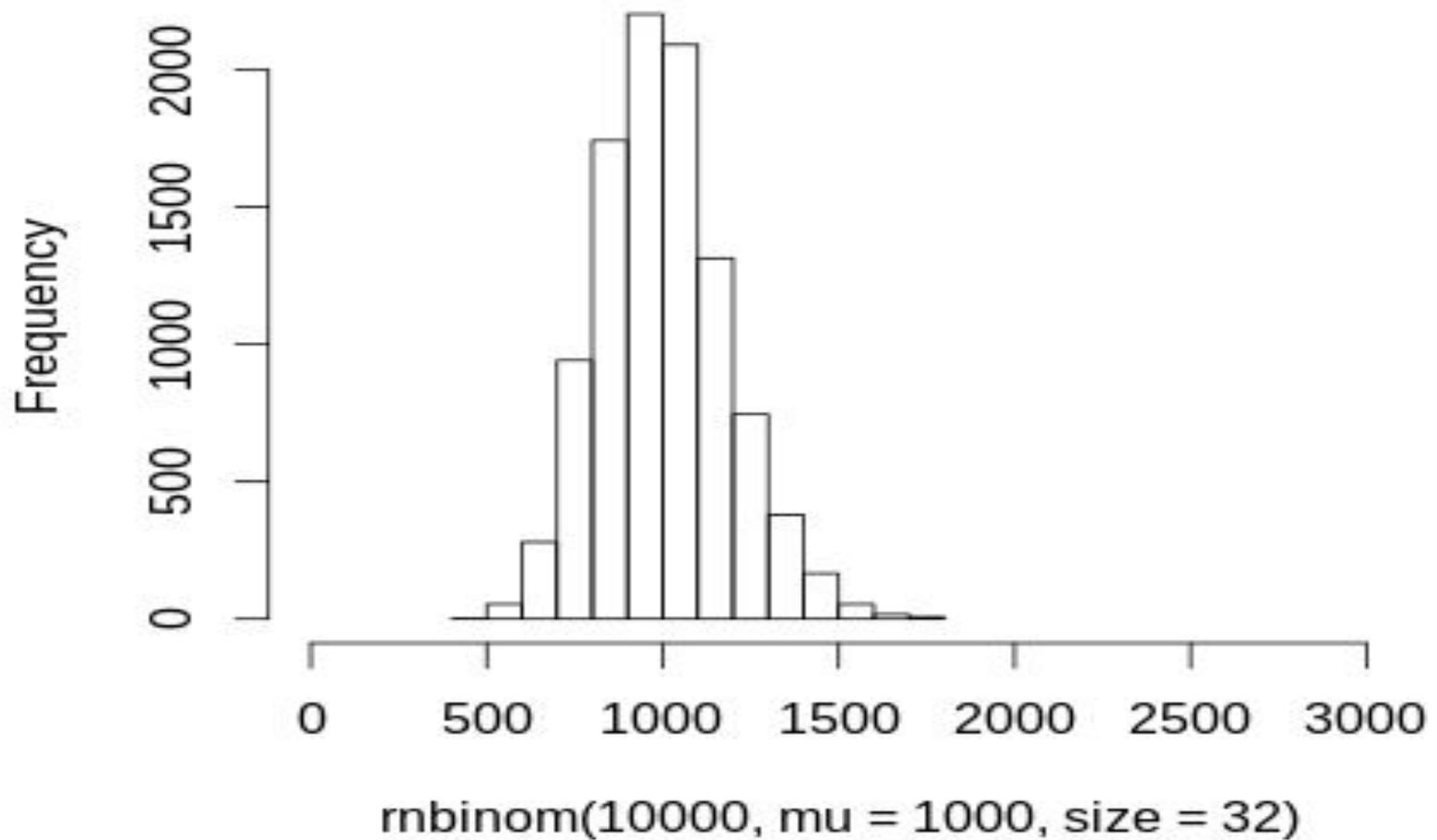


Dispersion Parameter = 16

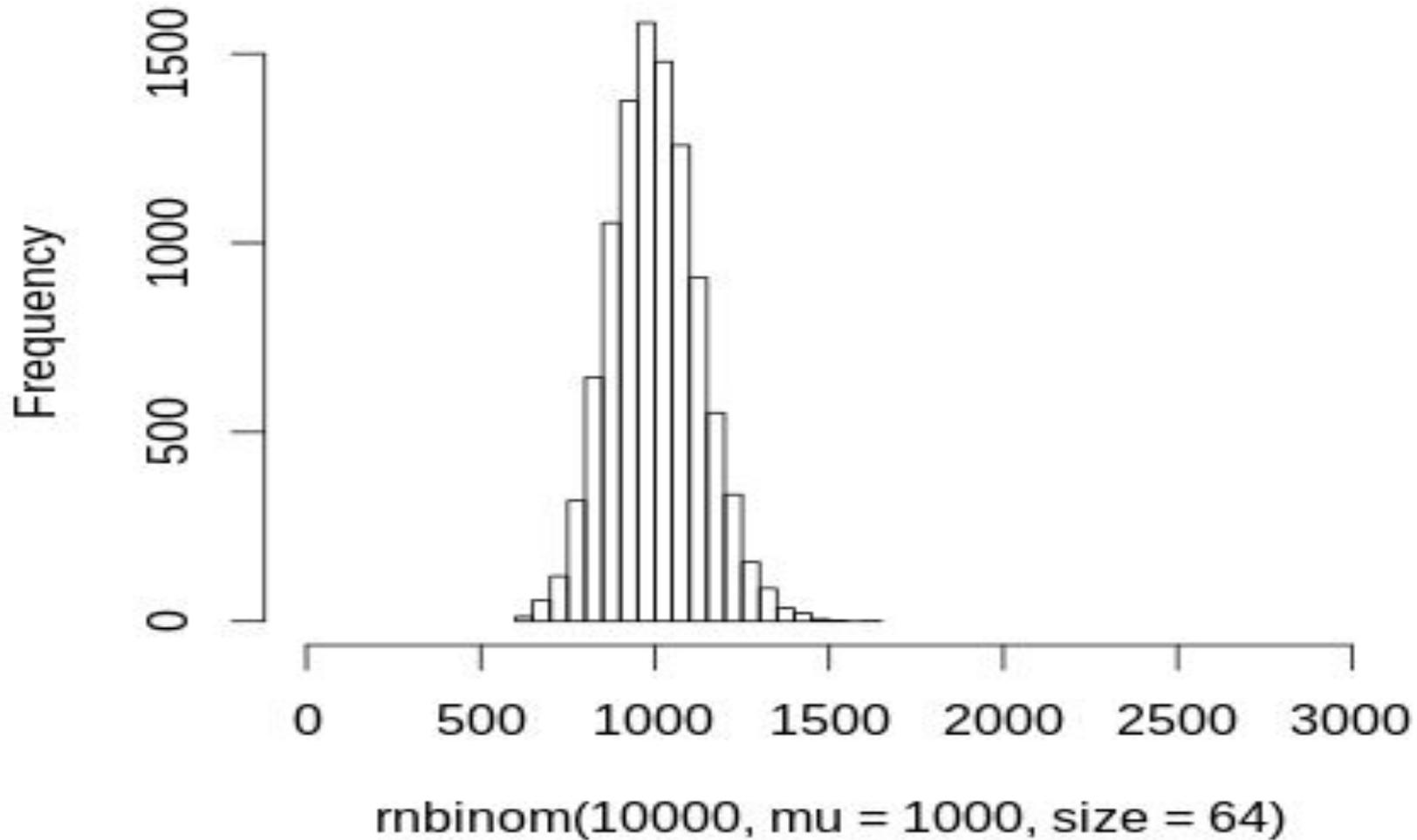


`rnbinom(10000, mu = 1000, size = 16)`

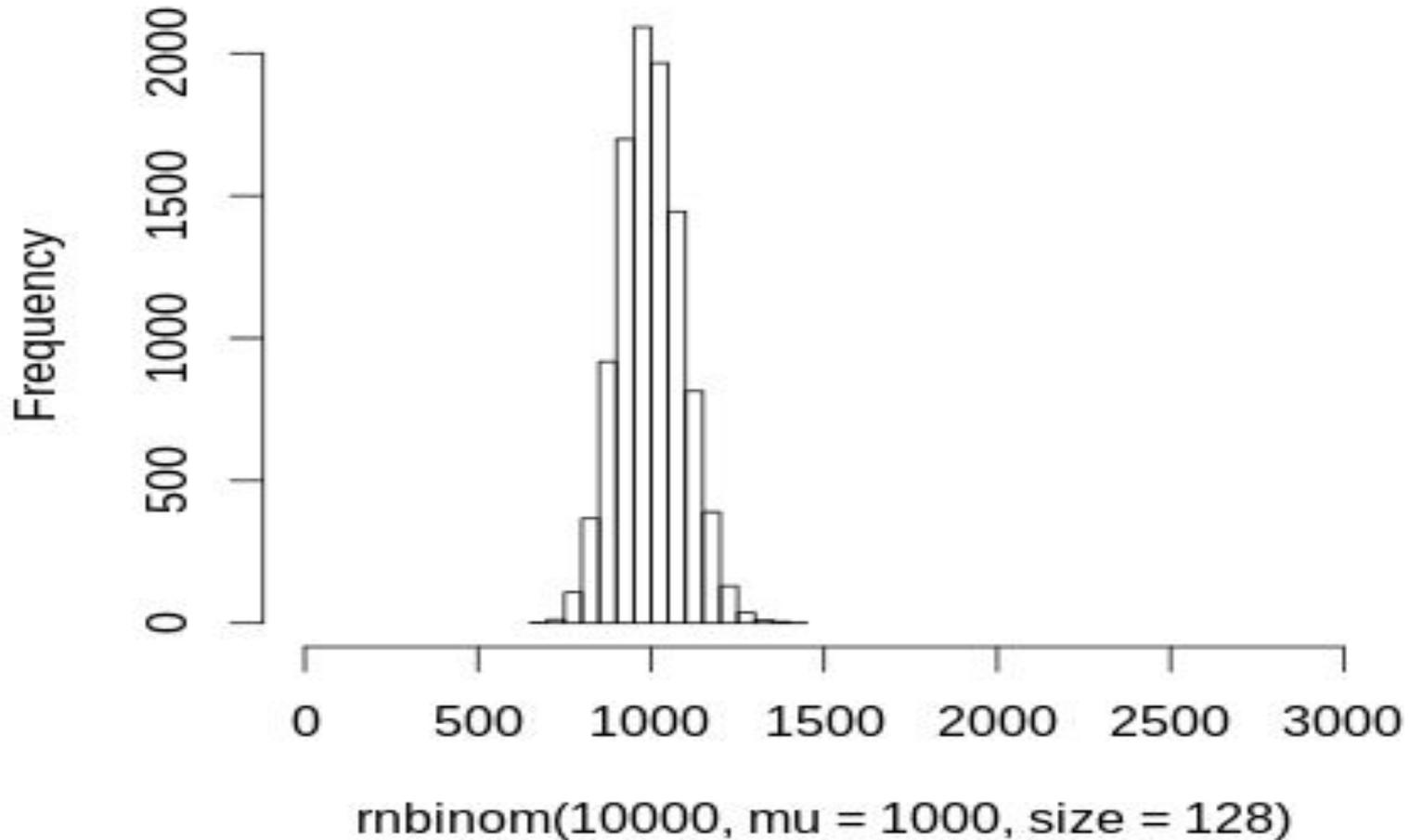
Dispersion Parameter = 32



Dispersion Parameter = 64



Dispersion Parameter = 128



Form of Likelihood Function 2: Dealing with Multiple Datasets

Two examples of (negative binomial) sub-likelihood functions:

$$\frac{i_{et}}{i_{et} + r_{er}}$$

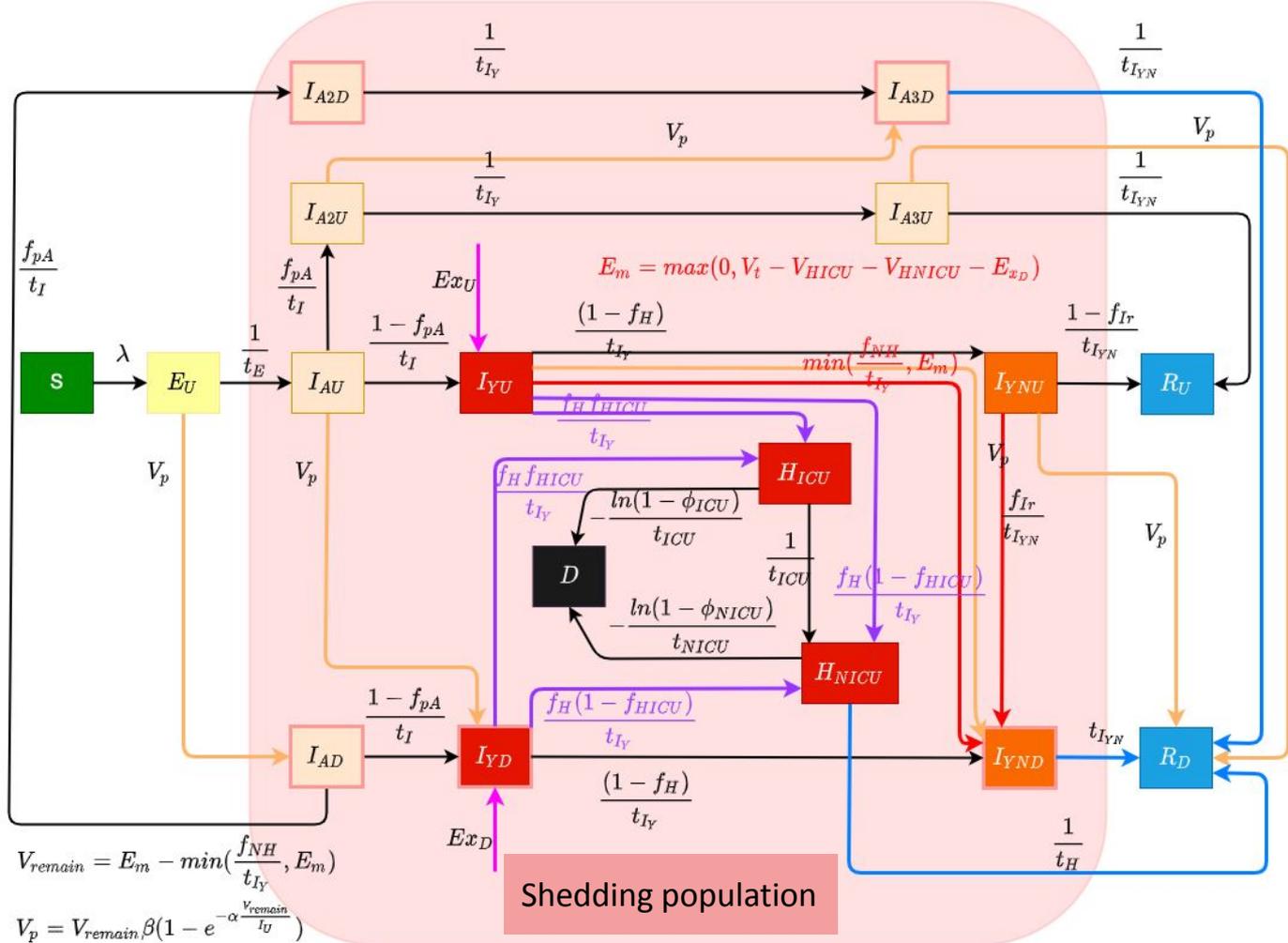
1. The sub-likelihood function of only considering new reported COVID cases:

$$\mathcal{L}_{NewReportedEndogenousCases} = \binom{y_{et} + r_{er} - 1}{r_{er} - 1} p_{er}^{r_{er}} (1 - p_{er})^{y_{et}}$$

$\frac{H_{ICU}}{H_{ICU} + r_{ICU}}$

2. The sub-likelihood function of only considering new reported COVID cases:

$$\mathcal{L}_{ICUCensusCases} = \binom{y_{ICU} + r_{ICU} - 1}{r_{ICU} - 1} p_{ICU}^{r_{ICU}} (1 - p_{ICU})^{y_{ICU}}$$



The mathematical structure of the COVID-19 dynamic model employed in particle filtering

Recall: How to Perform PF on Agg. Model in a Nutshell

Start with stochastic System Dynamics/Compartmental/ODE model

Subscript model by 1000s or more particles (COVID-19 provincial models: 150,000 particles is standard)

Each particle has its own **full copy of model state** (anything that could differ b/t realizations)

Sample from initial model state from prior distribution; set weights uniformly to 1

(Prediction phase): **Between observations**

All particles evolve according to standard model dynamics (just perform integration of each particle's state until next observation; all particles survive)

Particle weights remain invariant

(Update phase): **At observation points**: For each particle, multiply particle weight by likelihood of observing the empirical observation vector given that particle's state

Resampling/"**Survival of the fittest**": If effective sample size is too low (too much disparity in weights) following observation, particles are resampled according to their weights, and weight is reset to 1

Particles with high weights reproduce; those with low weights tend to disappear

Trajectories can be sampled by maintaining *ancestry matrix holding lineages*

1. At time $k=0$:

- (1) Sample $X_0^{N(i)}$ from $q_0(x_0^N)$;
- (2) Compute a *weight* for each particle $w_0^{(i)} = \frac{1}{N_s}$. It indicates that the weight at initial time step follows uniform distribution.

For a dynamic model, this just involves running the model forward from the previous timepoint

2. At time $k \geq 1$, perform a recursive update as follows:

- (1) Advance the sampled state by sampling $X_k^{N(i)} \sim q_k(x_k^N | y_k, X_{0:k-1}^{N(i)})$ and set $X_{0:k}^{N(i)} = (X_{0:k-1}^{N(i)}, X_k^{N(i)})$;
- (2) Update the weights to reflect the probabilistic and state update models as follows:

Likelihood function

$$w_k^{(i)} = W_{k-1}^{(i)} \frac{p(y_k^M | X_k^{N(i)}) p(X_k^{N(i)} | X_{k-1}^{N(i)})}{q(X_k^{N(i)} | X_{k-1}^{N(i)}, y_k^M)}$$

With the condensation algorithm we choose these two terms to be identical & they cancel

Image from Li, X., 2018. Incorporating Particle Filtering and System Dynamic Modelling in Infection Transmission of Measles and Pertussis. M.Sc. Thesis. Department of Computer Science, University of Saskatchewan.

Normalize the weights $W_k^{(i)} = \frac{w_k^{(i)}}{\sum_{i=1}^{N_s} w_k^{(i)}}$

Form of Likelihood Function 2: Dealing with Multiple Datasets

Two examples of the sub-likelihood function:

1. The sub-likelihood function only considering new reported COVID cases $\propto \frac{i_{et}}{i_{et} + r_{er}}$

$$\mathcal{L}_{NewReportedEndogenousCases} = \binom{y_{et} + r_{er} - 1}{r_{er} - 1} p_{er}^{r_{er}} (1 - p_{er})^{y_{et}}$$

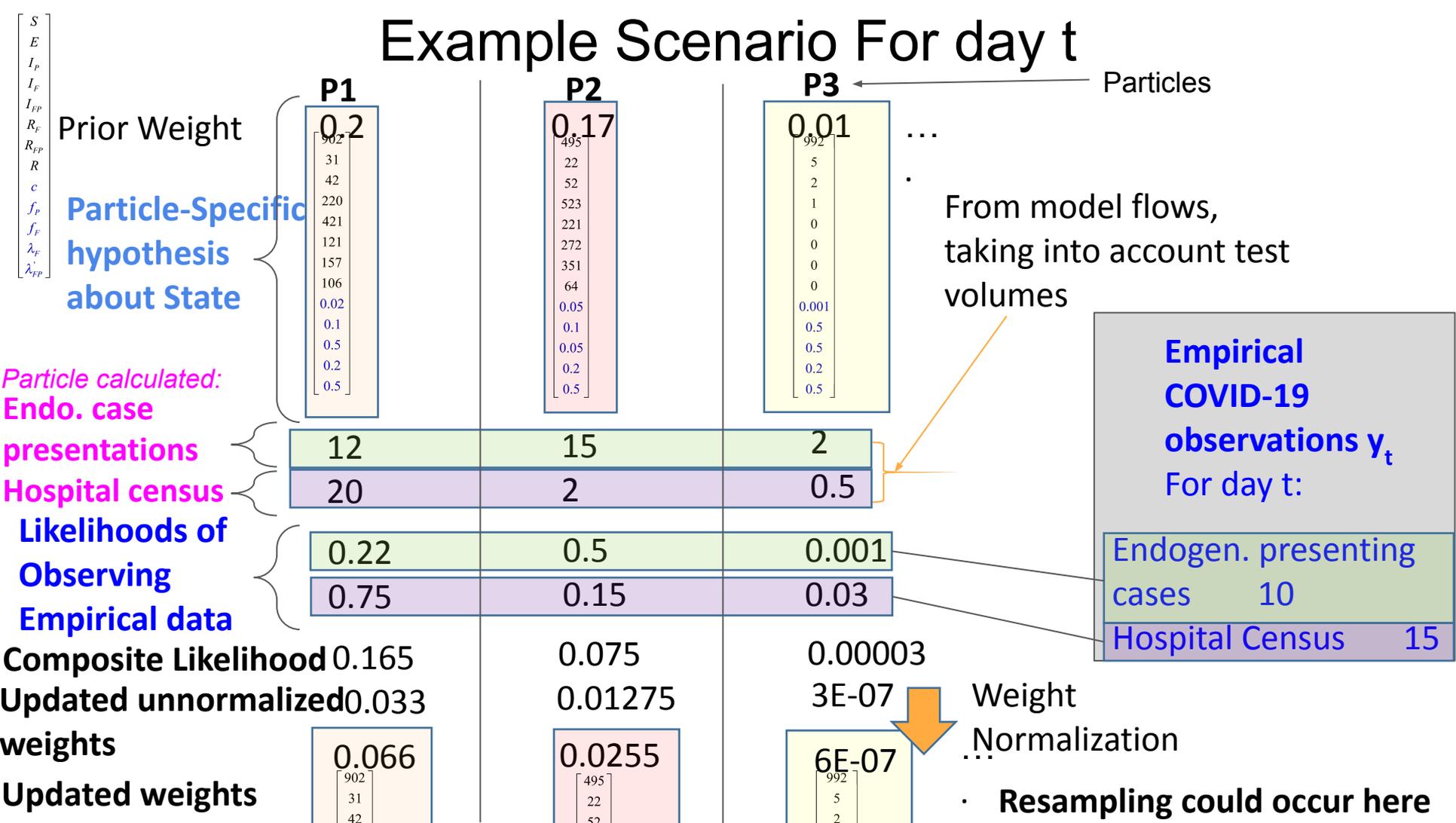
2. The sub-likelihood function only considering wastewater data (when available)

$$\mathcal{L}_{ViralConcentration} = \binom{y_{VC} + r_{VC} - 1}{r_{VC} - 1} p_{VC}^{r_{VC}} (1 - p_{VC})^{y_{VC}}$$

$$p_{VC} = \frac{x_{VC}}{x_{VC} + r_{VC}}$$

$$x_{VC} = \gamma(I_{AU} + I_{AD} + I_{A2U} + I_{A2D} + I_{A3U} + I_{A3D} + I_{YU} + I_{YD} + I_{YNU} + I_{YND} + H_{ICU} + H_{NICU})$$

Example Scenario For day t



Recall: How to Perform PF on Agg. Model in a Nutshell

Start with stochastic System Dynamics/Compartmental/ODE model

Subscript model by 1000s or more particles (COVID-19 provincial models: 150,000 particles is standard)

Each particle has its own **full copy of model state** (anything that could differ b/t realizations)

Sample from initial model state from prior distribution; set weights uniformly to 1

(Prediction phase): **Between observations**

All particles evolve according to standard model dynamics (just perform integration of each particle's state until next observation; all particles survive)

Particle weights remain invariant

(Update phase): **At observation points**: For each particle, multiply particle weight by likelihood of observing the empirical observation vector given that particle's state

Resampling/“Survival of the fittest”: If effective sample size is too low (too much disparity in weights) following observation, particles are resampled according to their weights, and weight is reset to 1

Particles with high weights reproduce; those with low weights tend to disappear

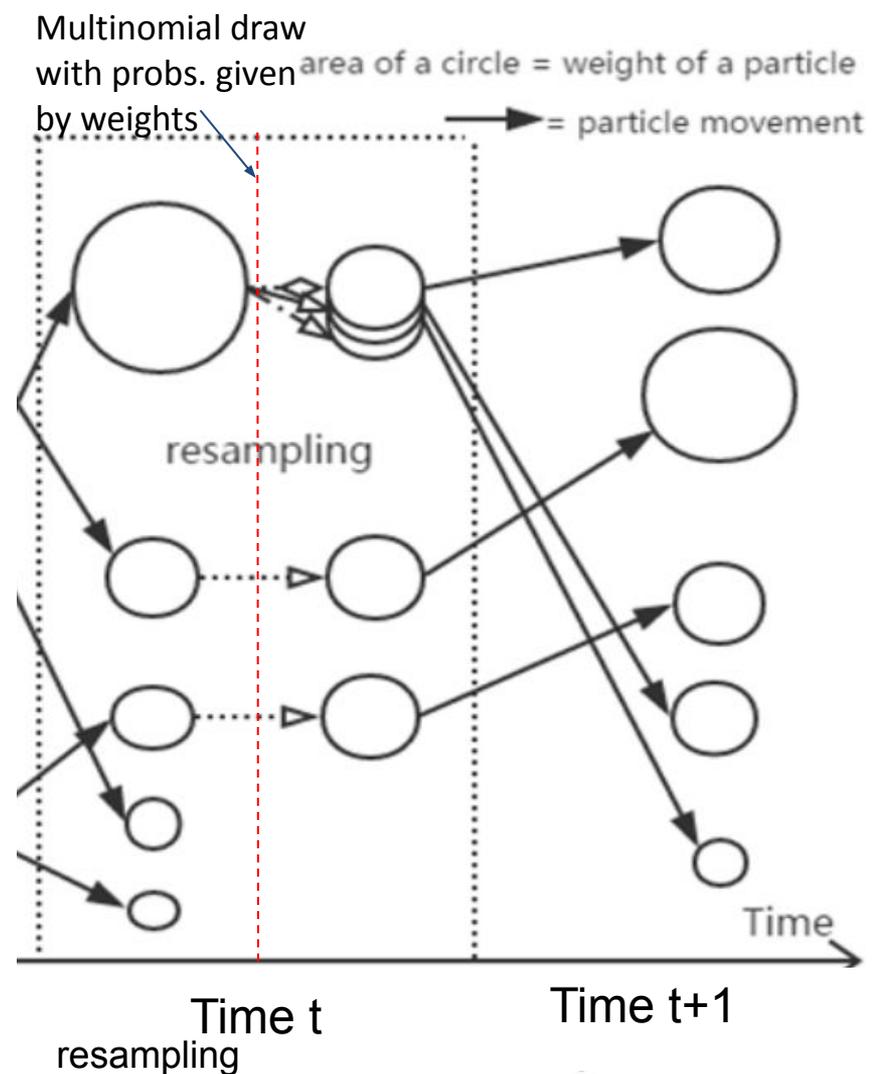
Trajectories can be sampled by maintaining *ancestry matrix holding lineages*

Agenda

- Retrospective
- Strengths & challenges of particle filtering for dynamic models
- Basic algorithmic specification
- Understanding the importance sampling context
- Evolving parameters & balancing model stochastics during particle filtering
- Understanding the likelihood functions
- **Resampling specification**
- Modeling structures to capture wastewater dynamics
 - Population shedding profile
 - Sampling from accumulations
 - Delays in wastewater propagation
 - Stormwater/wastewater coupling
- Handling presence/absence assays from wastewater
- Forecasting from present: Projecting particle-filtered models forward
- Evaluating particle filter performance
- PMCMC

Resampling

Triggered after weight updates & normalization if effective sample size lies below a certain threshold (or a certain fraction of the total number of particles)



Resampling Step

- When there is too large a diversity of particle weights following the updating of weights, we *perform a (weighted) resampling* from the particles
 - We draw a new set of particles from the set of particles (reflecting the updated weights), where the chance of selecting a given particle is proportional to its weight
 - A given particle may disappear or be duplicated many times
 - NB: If it is duplicated several times, note that the resulting particles have a complete copy of the state of the original particle (such that this state can then evolve independently)
 - The resampled particles are assigned a weight of 1

Practical Problem: Reduced Effective Sample Size

- Performed naively, the algorithm above can lead to a situation where
 - Most particles have very low weight
 - Only a few particles have significant weight
- This situation gives a low “effective sample size” in that with sequential importance sampling, the high weight particles will be overwhelmingly overrepresented
- We can recognize this situation by monitoring the variance in weights, using the second moment of the weights

$$S_{eff} = \frac{1}{\sum_{i=1}^N (w_k^{(i)})^2}$$

- If $S_{eff} < S_T$, we view the samples as suffering too low a sample size, and perform resampling

Resampling Step

- Suppose we have particles $X_k^{N(i)}$ (call these $X_{k-}^{N(i)}$) whose weight $W_{k-}^{(i)}$ has just been updated by an observation;
- Let $S_{eff} = \frac{1}{\sum_{i=1}^N (W_k^{(i)})^2}$
- If $S_{eff} < S_T$, then $\forall i, 1 \leq i \leq N$
 - $X_{k+}^{N(i)} \sim X_{k-}^{N(\text{multinomial}(W_{k-}^{(1..N)}))}$
 - $W_{k+}^{(i)} = 1/N$
- Please note that $\text{multinomial}(W_{k-}^{(1..N)})$ represents a sample from the multinomial distribution, taking as arguments N parameters representing the probabilities of returning each value (here $W_{k-}^{(1..N)}$) and returning an index of the chosen value

Agenda

- Retrospective
- Strengths & challenges of particle filtering for dynamic models
- Basic algorithmic specification
- Understanding the importance sampling context
- Evolving parameters & balancing model stochastics during particle filtering
- Understanding the likelihood functions
- Resampling specification
- **Modeling structures to capture wastewater dynamics**
 - Population shedding profile
 - Sampling from accumulations
 - Delays in wastewater propagation
 - Stormwater/wastewater coupling
- Handling presence/absence assays from wastewater
- Forecasting from present: Projecting particle-filtered models forward
- Evaluating particle filter performance
- PMCMC

Wastewater: Additional Texture

- Characterizing shedding population
- Delays
- Accumulations
- Stormwater event cross-over

Wastewater: Additional Texture

- Characterizing shedding population
- Accumulations
- Delays
- Stormwater event cross-over

Shedding Population

- To be used in particle filtering, wastewater concentrations need to be related to the disease status of the population
- Characterizing viral concentration levels in wastewater benefits from judicious consideration of the shedding population
- Appropriate consideration of variability in shedding over the natural history of infection can support better estimation of the distribution of the population across disease states
- Recommended: Using a weighted sum of values of state variables of infected parties

Wastewater Data Examined Here: Sampled Storage & Laboratory Context

- Wastewater plant freezes samples & stores throughout week
- Samples are shipped to Toxicology centre at end of sampling week
- Laboratory samples from a given week are analyzed in a block
- Tox. Centre provides efficiency-adjusted viral concentration estimates (N2 copies/100 mL)
- Results are reported following normalization with respect to efficiency of RNA extraction, qPCR (judged via a reference surrogate armed virus) and recovery ratio

Key Parameter: Ratio Between Reported Normalized Viral Concentration & Shedding Population (γ)

- Based on feedback from laboratory partners, we assumed a linear scaling relationship between shedding population & reported viral concentrations
- The scaling term is γ
- We sought to bound possible values for coefficient γ based on estimates of shedding population and empirical concentration values
- Wastewater infrastructure will affect the extent of other model structures required

Early Stage Infection Dominates COVID-19 Shedding

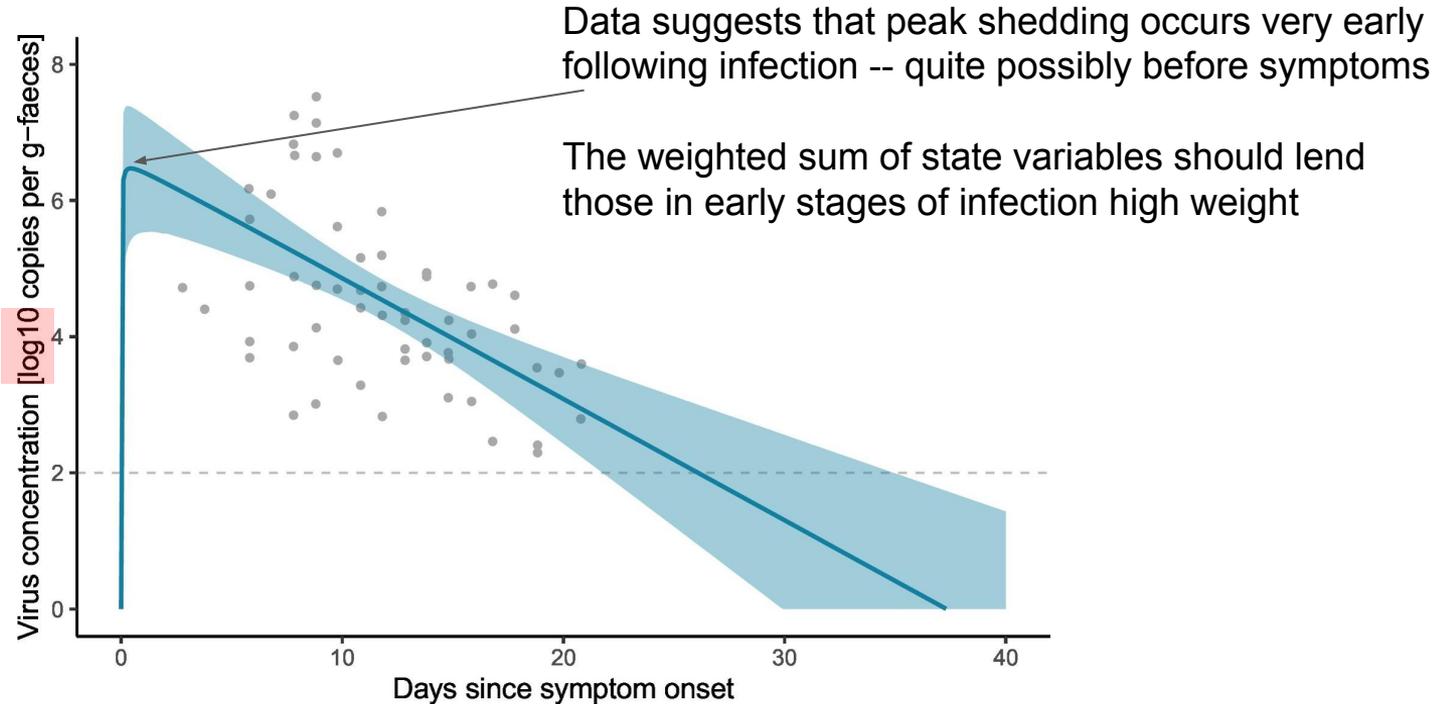


Image from Miura F, Kitajima M, Omori R. Duration of SARS-CoV-2 viral shedding in faeces as a parameter for wastewater-based epidemiology: Re-analysis of patient data using a shedding dynamics model. *Science of The Total Environment*. 2021 May 15;769:144549.

Fecal Shedding over Time: Data Gaps & Variability

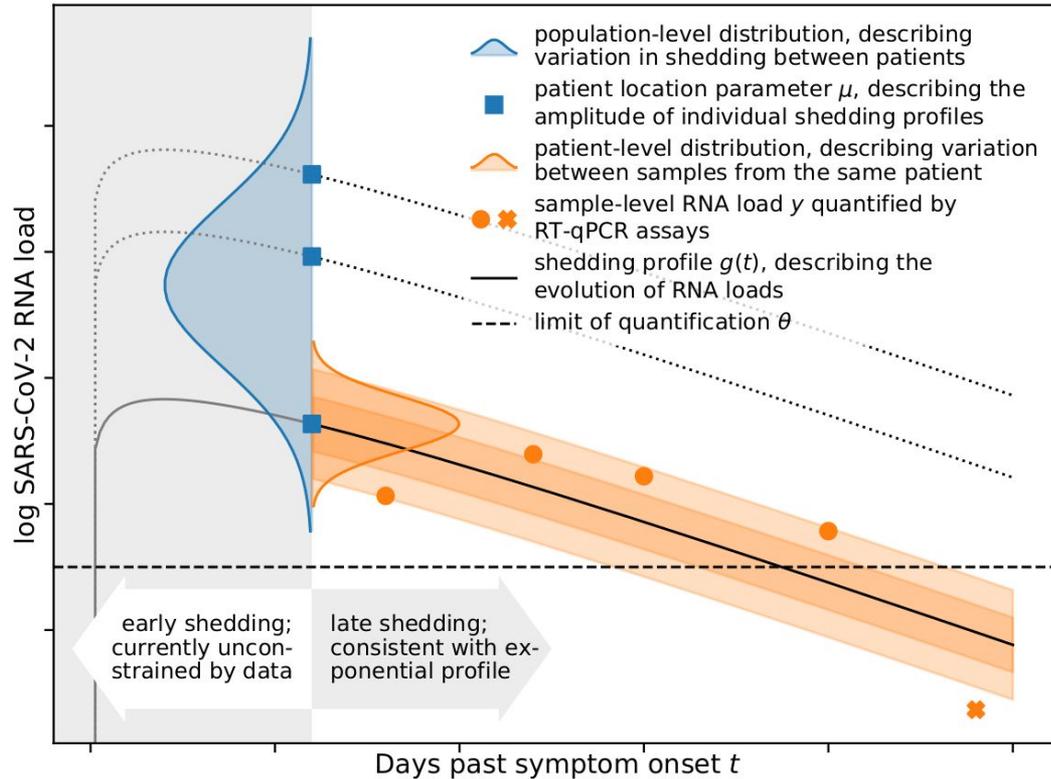


Image from Hoffmann T, Alsing J. Faecal shedding models for SARS-CoV-2 RNA amongst hospitalised patients and implications for wastewater-based epidemiology. medRxiv. 2021 Jan 1.

Population Heterogeneity in Duration of Shedding

- Somewhat longer shedding fecally than via respiratory
- Longer overall shedding in
 - Symptomatic cases
 - Adults vs. children
 - Those receiving corticosteroidal treatment
 - Those with chronic disease

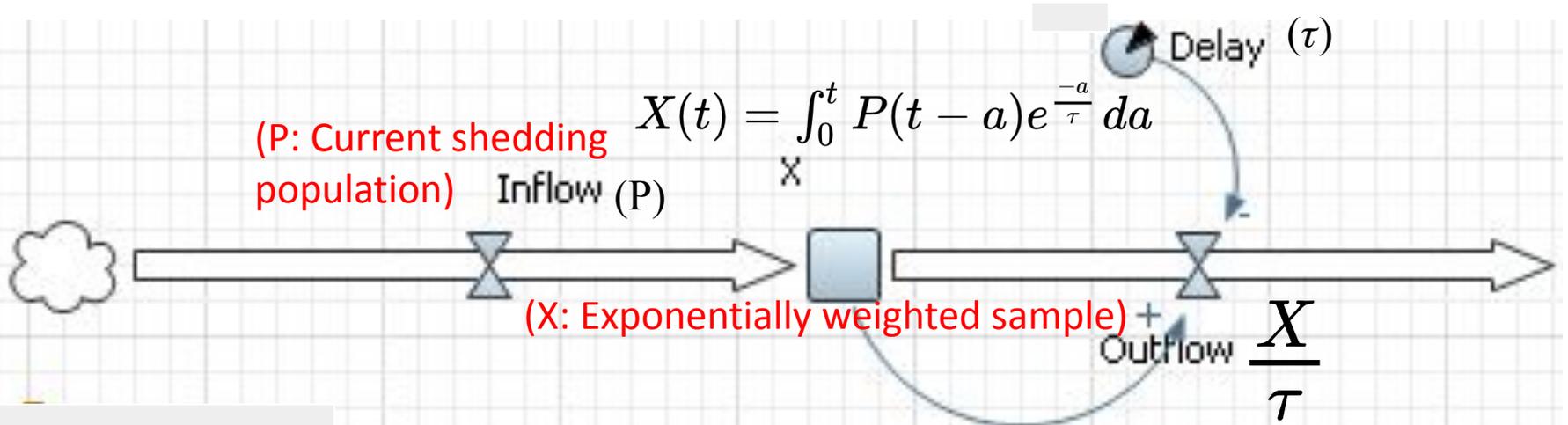
Yan D, Zhang X, Chen C, Jiang D, Liu X, Zhou Y, Huang C, Zhou Y, Guan Z, Ding C, Chen L. Characteristics of viral shedding time in SARS-CoV-2 infections: A systematic review and meta-analysis. *Frontiers in public health*. 2021 Mar 19;9:209.

Wastewater: Additional Texture

- Characterizing shedding population
- **Accumulations**
- Delays
- Stormwater event cross-over

Capturing Sampling from Accumulations

- There are sometimes sampling from accumulations that mix together signal from shedding occurring different lengths of time in the past
- One solution: Taking an exponentially weighted average of shedding using a “first order delay” $\frac{dX}{dt} = \text{Current Shedding population} - \frac{X}{\tau}$



Wastewater Data Examined Here: Infrastructure Context

- All wastewater & health data considered here is specifically from Saskatoon
- Saskatoon has almost entirely separate wastewater & stormwater infrastructures
 - Daily flow is not affected by precipitation, first flush or snowmelt
 - Minimal risk of pH&salinity changes
 - Small amounts of infrastructural coupling could apply during extreme weather events not experienced during the sampling timeframe
- Wastewater data is from municipal-wide primary effluent wastewater (~80ML/Day)
- To normalize with respect to within-day effects with respect to effluent flow, autosampling is conducted throughout day from a moving incoming effluent stream
- U of S Toxicology Centre estimates a transit time from homes to the wastewater plant primary effluent of **below 12 hours & possibly half that** ⇒ less degradation
- Wastewater transition to the wastewater treatment plant is made by sewer several meters below ground, which preserve temperature in the 12-18C range

Wastewater: Additional Texture

- Characterizing shedding population
- Accumulations
- **Delays**
- Stormwater event cross-over

Capturing Delays in Wastewater Propagation

- In large municipalities, it commonly takes large fraction of a day to many days for wastewater to move from homes/workplaces to the sampling location
- These delays can mean that wastewater signal is providing understanding of population health state not about now, but about the recent past
- When we are using wastewater measurements to help estimate model state, large delays must be taken into account
 - This lets Particle Filtering select for particles whose population state at the time of wastewater production (shedding a few days back) is suggested by the current wastewater signal
- If there are markedly different delays for different areas of a city, we can represent several such delays within model, with different population subsets

3rd Order Delay

For 3rd Order Case:

$$\frac{dS_1}{dt} = \text{Current Shedding population} - \frac{S_1}{\tau}$$

$$\frac{dS_2}{dt} = \frac{S_1}{\tau} - \frac{S_2}{\tau}$$

$$\frac{dS_3}{dt} = \frac{S_2}{\tau} - \frac{S_3}{\tau}$$

General Form:

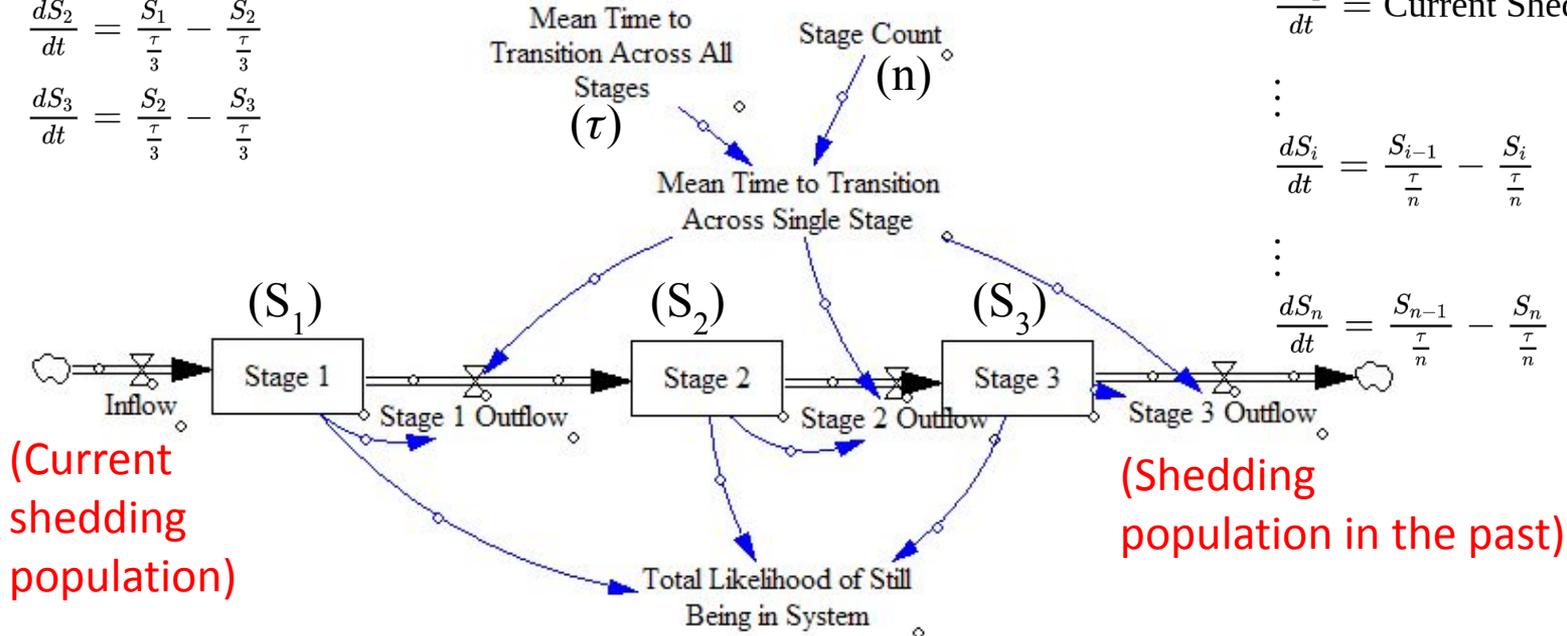
$$\frac{dS_1}{dt} = \text{Current Shedding population} - \frac{S_1}{\tau}$$

⋮

$$\frac{dS_i}{dt} = \frac{S_{i-1}}{\tau} - \frac{S_i}{\tau}$$

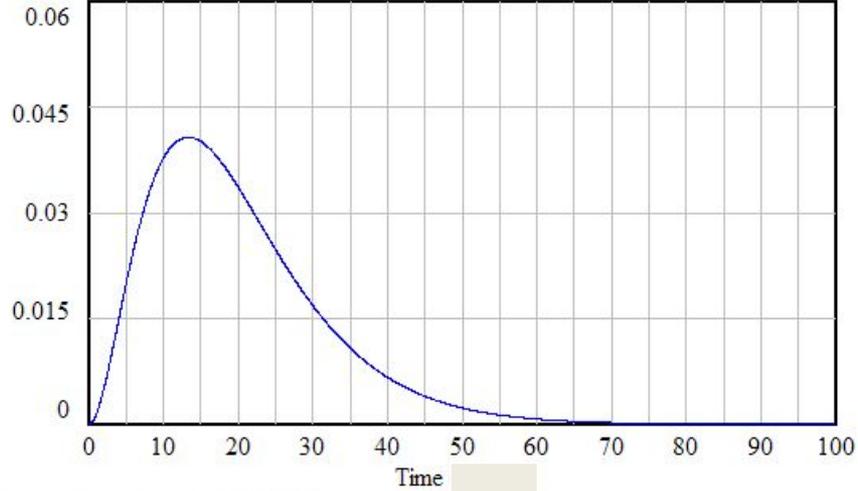
⋮

$$\frac{dS_n}{dt} = \frac{S_{n-1}}{\tau} - \frac{S_n}{\tau}$$



3rd Order Delay

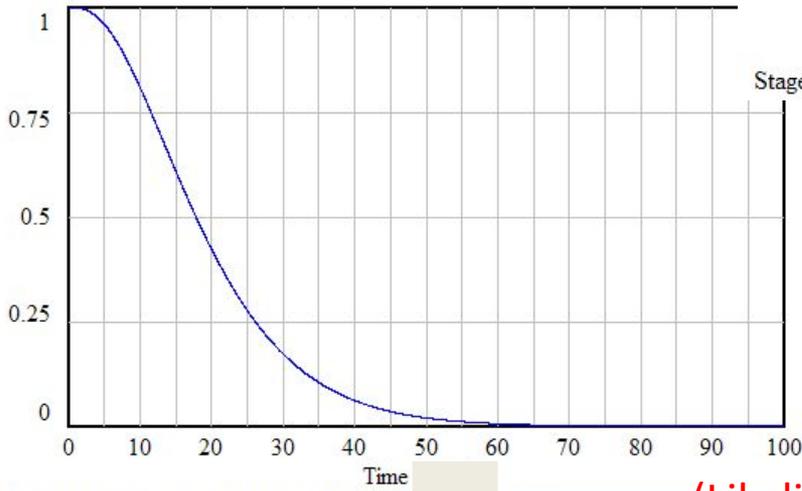
Stage 3 Outflow



Stage 3 Outflow : 3rd Order Delay

(Per-unit-time chance of transitioning out over time)

Total Likelihood of Still Being in System

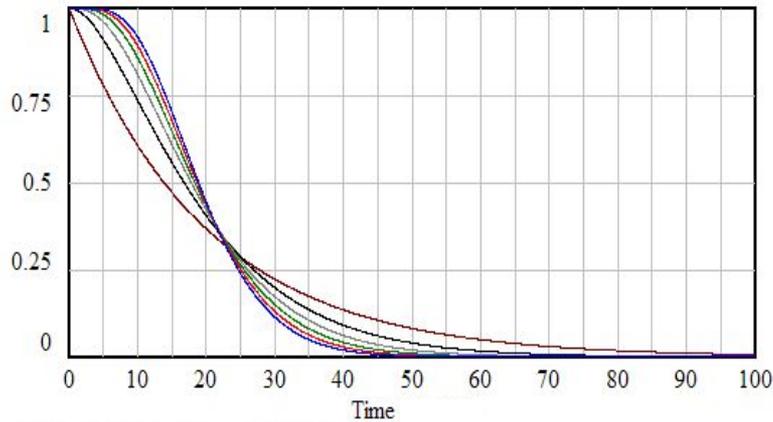


Total Likelihood of Still Being in System : 3rd Order Delay

(Likelihood of still being in transit)

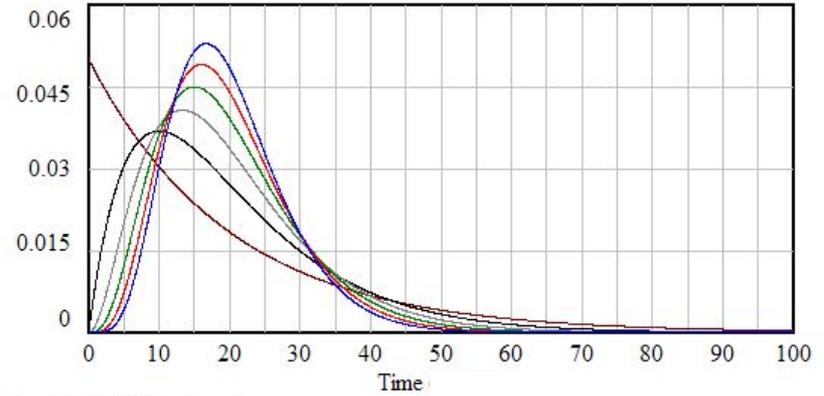
1st through 6th OrderDelays

Total Likelihood of Still Being in System



Total Likelihood of Still Being in System : 6th Order Delay For Comparison
 Total Likelihood of Still Being in System : 5th Order Delay For Comparison
 Total Likelihood of Still Being in System : 4th Order Delay For Comparison
 Total Likelihood of Still Being in System : 3rd Order Delay For Comparison
 Total Likelihood of Still Being in System : 2nd Order Delay For Comparison
 Total Likelihood of Still Being in System : 1st Order Delay For Comparison

Outflow



Outflow : 6th Order Delay For Comparison
 Outflow : 5th Order Delay For Comparison
 Outflow : 4th Order Delay For Comparison
 Outflow : 3rd Order Delay For Comparison
 Outflow : 2nd Order Delay For Comparison
 Outflow : 1st Order Delay For Comparison

(Per-unit-time chance of
transitioning out over time)

(Likelihood of still being in transit)

Wastewater: Additional Texture

- Characterizing shedding population
- Delays
- Accumulations
- Stormwater event cross-over

Capturing Dynamics of Stormwater-Wastewater Coupling

- The episodic or frequent intrusion of stormwater into wastewater systems can affect measurement in several ways, including
 - Diluting the wastewater
 - Introducing temperature & salinity changes that affect laboratory assay efficiency
- Lab processes commonly seek to normalize for changes in nominal concentration levels, mediating & leading to less direct impacts on concentration measurements
 - Be sure to consult with those knowledgeable in laboratory methods & normalization procedures to understand likely impact of stormwater intrusion on wastewater data
- One area where this is likely to still have a marked impact on inference from wastewater concentration is in the reliability & variability in measurement
 - Use of time-varying dispersion parameters for likelihood functions may be appropriate

Agenda

- Retrospective
- Strengths & challenges of particle filtering for dynamic models
- Basic algorithmic specification
- Understanding the importance sampling context
- Evolving parameters & balancing model stochastics during particle filtering
- Understanding the likelihood functions
- Resampling specification
- Modeling structures to capture wastewater dynamics
 - Population shedding profile
 - Sampling from accumulations
 - Delays in wastewater propagation
 - Stormwater/wastewater coupling
- **Handling presence/absence assays from wastewater**
- Forecasting from present: Projecting particle-filtered models forward
- Evaluating particle filter performance
- PMCMC

Handling Presence-Absence Detection

- Presence-absence data is of great value in certain contexts
 - Arrival of pathogen in vulnerable communities (e.g., remote communities, crowded housing, homeless shelters, incarceration facilities, aged care, dormitories)
 - Arrival of a new variant of concern in any community or region
- One successful strategy: Stochastic arrivals that are systematically selected against if absent, but otherwise selected for
 - Low-rate stochastic transition always allows some small chance of pathogen/variant arrival
 - Particle filtering will select **against** particles positing premature arrival after a few observations
 - Particle filtering will select **for** particles that presciently posit arrival substantiated by subsequent observations including consistent reports

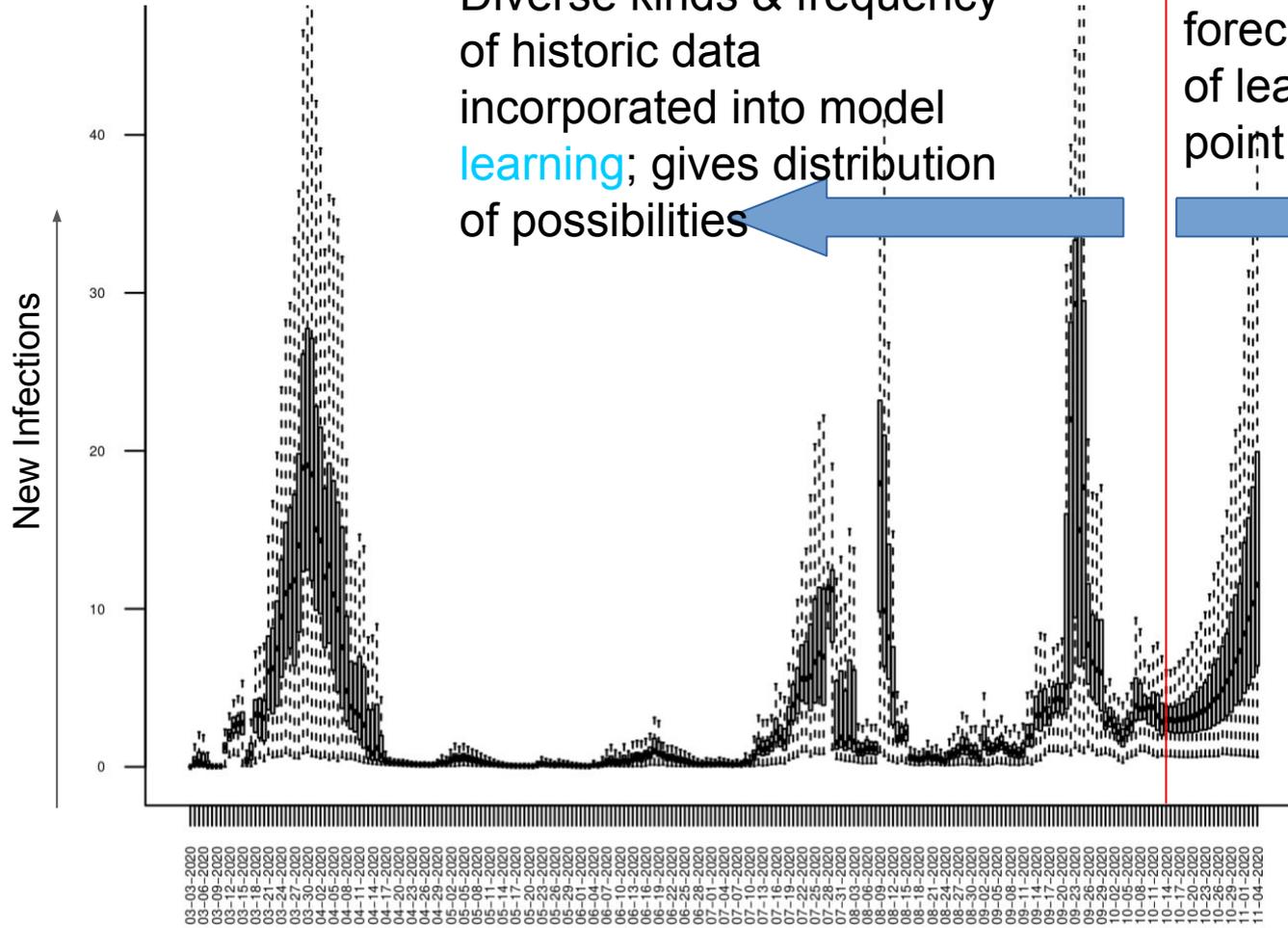
Agenda

- Retrospective
- Strengths & challenges of particle filtering for dynamic models
- Basic algorithmic specification
- Understanding the importance sampling context
- Evolving parameters & balancing model stochastics during particle filtering
- Understanding the likelihood functions
- Resampling specification
- Modeling structures to capture wastewater dynamics
 - Population shedding profile
 - Sampling from accumulations
 - Delays in wastewater propagation
 - Stormwater/wastewater coupling
- Handling presence/absence assays from wastewater
- **Forecasting from present: Projecting particle-filtered models forward**
- Evaluating particle filter performance
- PMCMC

Projecting Forward

- The model structure is posited to incorporate the “physics” of the underlying causal situation
- Particle filtering cannot continue after the time horizon of the data, but the ODE model can still be simulated forward
- For the projection regime, there is a need to determine how to handle the evolving parameters undergoing random walks
 - “Status quo”: Project forward at the current value of those evolving parameter
 - Allow continued random walks by the parameters

Projection Discrepancies



Diverse kinds & frequency
of historic data
incorporated into model
learning; gives distribution
of possibilities

Model
forecasts in light
of learning to this
point



Agenda

- Retrospective
- Strengths & challenges of particle filtering for dynamic models
- Basic algorithmic specification
- Understanding the importance sampling context
- Evolving parameters & balancing model stochastics during particle filtering
- Understanding the likelihood functions
- Resampling specification
- Modeling structures to capture wastewater dynamics
 - Population shedding profile
 - Sampling from accumulations
 - Delays in wastewater propagation
 - Stormwater/wastewater coupling
- Handling presence/absence assays from wastewater
- Forecasting from present: Projecting particle-filtered models forward
- **Evaluating particle filter performance**
- PMCMC

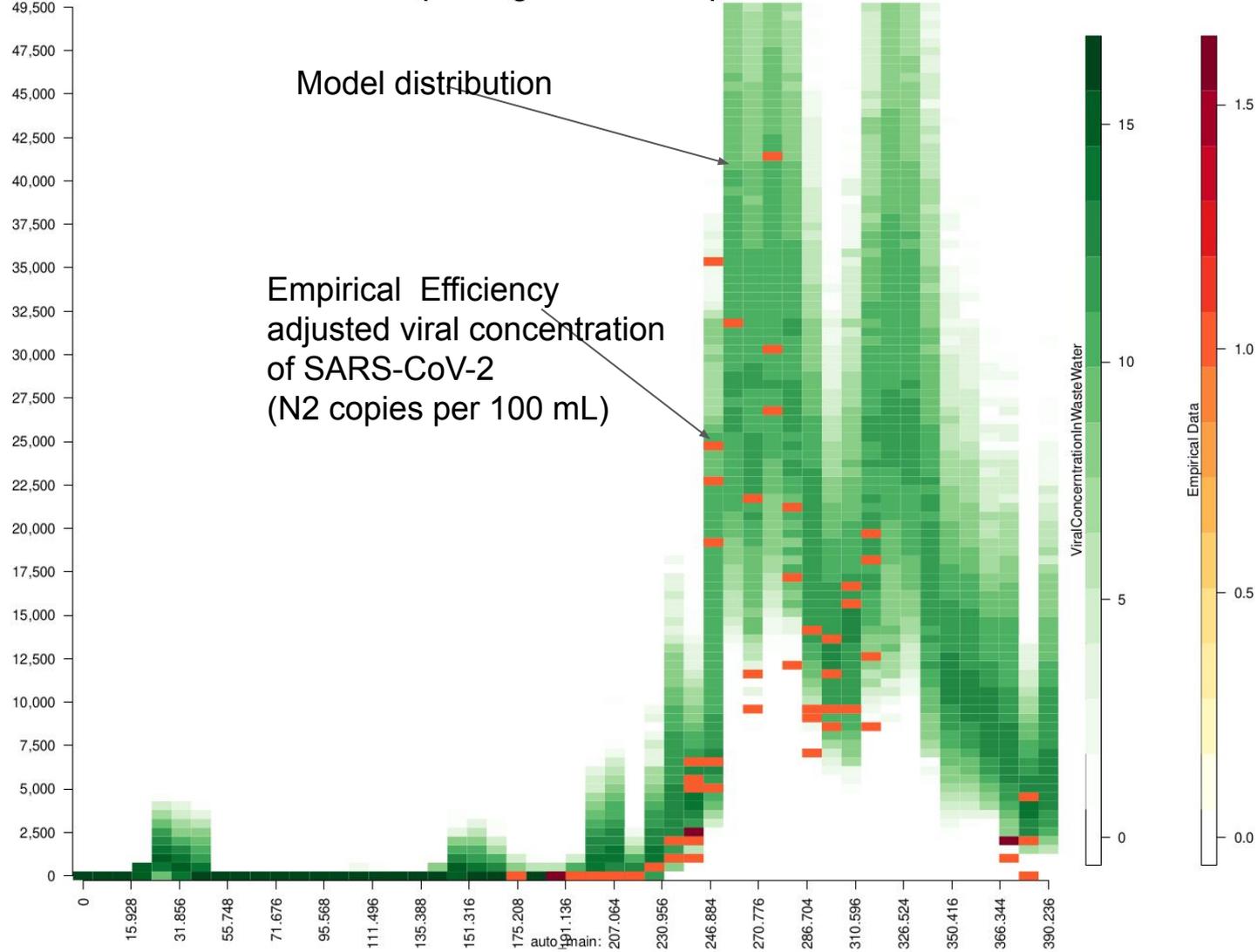
Evaluation Options Explored

- Posterior discrepancy -- compare posterior predicted and empirical observables
 - Sample particles 1,000s of times to compute the average per-unit-time L2 norm (mean root sum squared discrepancy) between empirical data and corresponding particle predictions
- Projective accuracy -- Train on data points in a prefix time horizon, test evolution against later data points: Over several times t in historic time horizon
 - Perform particle filtering until time t
 - Project some collection of samples forward (without particle filtering) until time $t + \Delta t$ (e.g., $t + 2$ weeks)
 - Sample particles 1,000s of times to compute the average per-unit-time L2 norm (mean root sum squared discrepancy) between empirical data and corresponding particle predictions

Evaluation Options Explored

- **Posterior discrepancy -- compare posterior predicted and empirical observables**
 - Sample particles 1,000s of times to compute the average per-unit-time L2 norm (mean root sum squared discrepancy) between empirical data and corresponding particle predictions
- Projective accuracy -- Train on data points in a prefix time horizon, test evolution against later data points: Over several times t in historic time horizon
 - Perform particle filtering until time t
 - Project some collection of samples forward (without particle filtering) until time $t + \Delta t$ (e.g., $t + 2$ weeks)
 - Sample particles 1,000s of times to compute the average per-unit-time L2 norm (mean root sum squared discrepancy) between empirical data and corresponding particle predictions

Discrepancy
from
Wastewater
Data:
Considering
Wastewater
in Likelihood
with
Constant
coefficient γ
(Method B)



Posterior Discrepancy from All Data

40K Particles, Time horizon Feb 23, 2020 - Mar 26, 2021

Model Scenario within Scenario Set 1	Discrepancy from Health System Data over entire model time horizon	Discrepancy from both Health System & Wastewater Data over entire model time horizon
No wastewater data considered in likelihood	3.73	N/A (No discrepancy from WW data assessed)
Wastewater data with temporally varying coefficient relating concentration & infectives	3.70	3.82
Wastewater data with constant coefficient relating concentration & infectives	3.74	3.82

Posterior Discrepancy from Empirical Health System Data 75K & 150K Particles; Time horizon Feb 23, 2020 - Mar 31, 2021

Model Scenario	75K Particles 2 Realizations Each Scenario Set 2	150K Particles 1 Realization Each Scenario Set 3
No wastewater data considered in likelihood	3.713	3.722
Wastewater data with constant coefficient relating concentration & infectives	3.710	3.694

Impact of Wastewater Data Frequency & Smoothing on Posterior Discrepancy

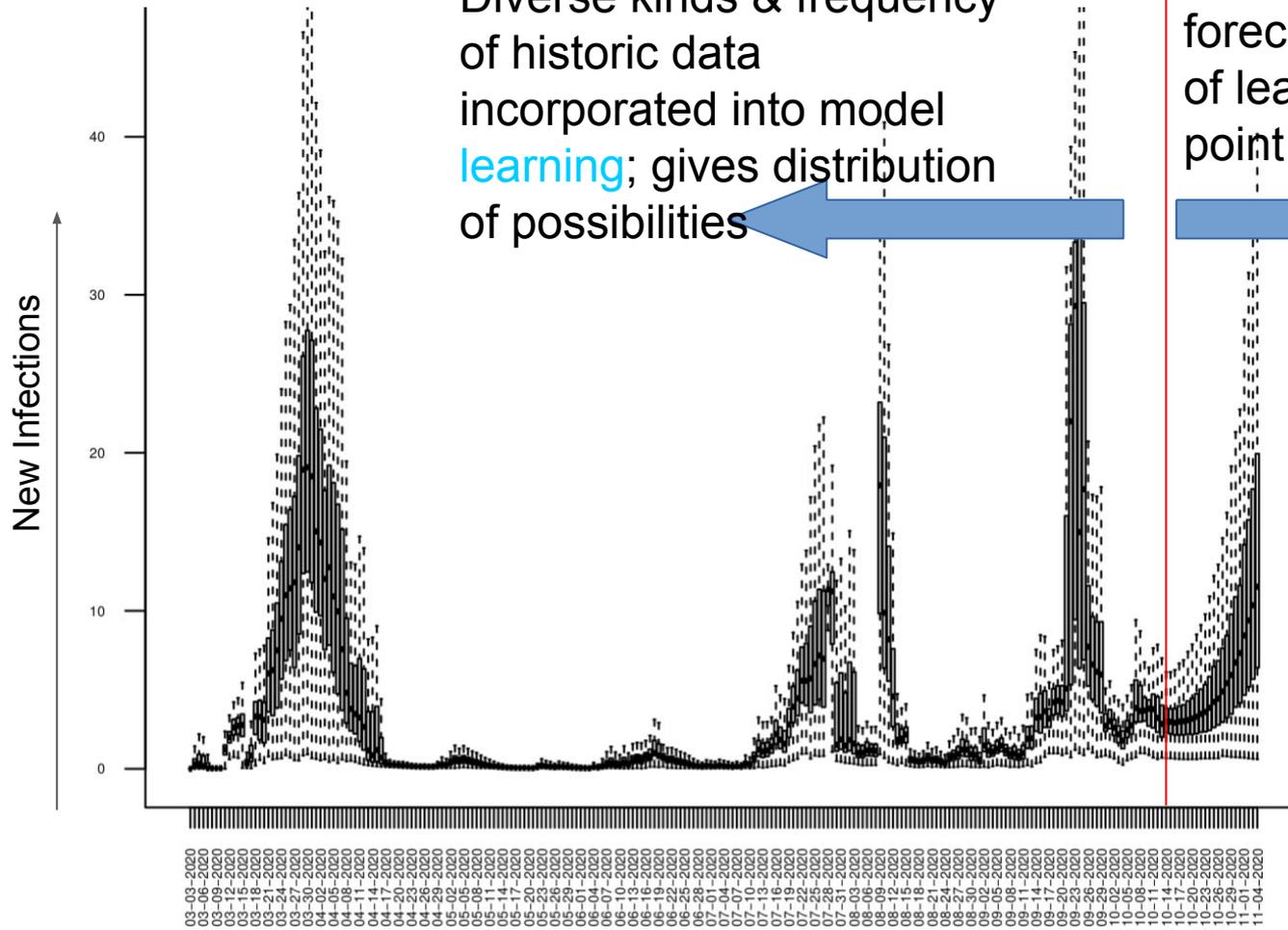


Model Scenario within Scenario Set 1	Discrepancy from Health System Data over entire model time horizon
No wastewater data considered in likelihood	
Wastewater data sampled 1 day per week	3.79
Wastewater data sampled 3 days per week	3.85
Wastewater data sampled 3 days per week & averaged	3.78

Evaluation Options Explored

- Posterior discrepancy -- compare posterior predicted and empirical observables
 - Sample particles 1,000s of times to compute the average per-unit-time L2 norm (mean root sum squared discrepancy) between empirical data and corresponding particle predictions
- Projective accuracy -- Train on data points in a prefix time horizon, test evolution against later data points: Over several times t in historic time horizon
 - Perform particle filtering until time t
 - Project some collection of samples forward (without particle filtering) until time $t + \Delta t$ (e.g., $t + 2$ weeks)
 - Sample particles 1,000s of times to compute the average per-unit-time L2 norm (mean root sum squared discrepancy) between empirical data and corresponding particle predictions

Projection Discrepancies



Diverse kinds & frequency
of historic data
incorporated into model
learning; gives distribution
of possibilities

Model
forecasts in light
of learning to this
point



Projection Discrepancy from Empirical Health System Data

150K Particles; health system data time horizon Feb 23, 2020 - Mar 24, 2021; projection period March 25, 2021 - April 8, 2021

Model Scenario	2 Realizations of 150,000 particles for each base scenario, with each realization being extended with 10 projection realizations utilizing 1000 particles
No wastewater data considered in likelihood	1.663
Wastewater data with constant coefficient relating concentration & infectives	1.596

Mann-Whitney one-way U-Test
exact p-value: 0.04296

The consideration of wastewater data in the likelihood employed during particle filtering yields a statistically significant reduction in **projection discrepancy** of the particle filtered estimates of current state and the empirical data during the projection time period.

Agenda

- Retrospective
- Strengths & challenges of particle filtering for dynamic models
- Basic algorithmic specification
- Understanding the importance sampling context
- Evolving parameters & balancing model stochastics during particle filtering
- Understanding the likelihood functions
- Resampling specification
- Modeling structures to capture wastewater dynamics
 - Population shedding profile
 - Sampling from accumulations
 - Delays in wastewater propagation
 - Stormwater/wastewater coupling
- Handling presence/absence assays from wastewater
- Forecasting from present: Projecting particle-filtered models forward
- Evaluating particle filter performance
- **PMCMC**

PMCMC

- PMCMC basics
 - Ancestral lineages
 - Key Influences on acceptance rate: Transition SD, particle count, dispersion parameter, # of data points
- PMCMC pragmatics
 - number of data points -- need for assessment
 - tuning acceptance rate: Particle count key
 - show graph of acceptance rate with particle count

PMCMC-Based Analytics for Wastewater

Bayesian Machine Learning & Dynamic Models

- **MCMC:** Sample from $p_M(\theta|y_{1:T})$: posteriors of *deterministic* dynamic model static parameters, latent states, scenario results, and incremental scenario gains.
- **Particle Filtering/SMC:** Sample from $p_{\theta,M}(x_{1:T}|y_{1:T})$: posteriors of *stochastic* dynamic model latent states stochastically evolving parameters, scenario results, and incremental scenario gains.
- **Particle MCMC (PMCMC):** Sample from $p_M(\theta, x_{1:T}|y_{1:T})$: posteriors of *stochastic* dynamic model latent states, stochastically evolving parameters, scenario results, and incremental scenario gains *and static parameters*.

PMCMC Basics

- Supports estimating (via sampling from) **joint** distributions of
 - Parameters (static and evolving)
 - System state over time
- High computational expense
 - 10,000+ MCMC iteration (burn-in period required)
 - Each MCMC iteration requires running particle filtering to sample *trajectory* of latent states
 - Each particle in particle filtering must run model
 - Often seek to run multiple MCMC walkers (chains)
 - Silver lining: Highly parallelizable
- MCMC is actually a *family* of algorithms

Family of Algorithms

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

See

Andrieu C., Doucet A. *Particle Markov chain Monte Carlo methods*. J. R. Statist. Soc. B (2010) 72, Part 3, pp. 269–342

Basic Approach to Sampling from

$$p(\theta, x_{1:T} | y_{1:T})$$

Samples jointly
From parameters
& latent state

- Find initial parameters θ by looping to find non-zero posterior value
- MCMC (Metropolis-Hastings) iteration, sampling parameters via MH-algorithm, by considering candidates θ^*
 - **Particle filter**, sampling from **trajectories conditional on candidate θ^***
 - This particle filtering assumes current candidate parameter value θ^* as parameter values
 - Every observation triggers resampling following weight update
 - **Entire trajectory** is sampled, according to **final weight**
 - This is part of sample if θ^* accepted
 - An ancestry matrix is maintained
 - Weights across successive observations across all trajectories support calculating the **posterior for the candidate parameter value θ^*** (NB: Can be sensitive to particle count!)
 - For each sampled value, emit sampled values (new or existing)
 - If accepted candidate θ^* , emit θ^* and sampled trajectory of latent state
 - If did not accept candidate, state in current place, and (re-)emit those (pre-existing) parameter values **and** the latent state (previously) sampled from them

Particle Filtering with ABMs

- Guidelines for effective particle filtering with ABMs have yet to be elucidated
- Given high nominal (& likely moderately high intrinsic) dimensionality of state space, non-sparse coverage requires high # of particles
- Exceptionally weighty computational resource demand
 - High dimensionality => High number of particles
 - Per-ensemble high because each particle is associated with a ...
 - Complete model state representation High memory need
 - ABM: Large populations & inter-agent interactions High computational burden
- Our lines of research
 - Case studies of tradeoffs
 - Creation of ABM-specialized platforms
 - Large-scale parallel implementation (GPU, future: FPGA)
 - Distributed computation

Acknowledgements

- Doctoral student Xiaoyan Li provided essential leadership in integrating wastewater data into particle filtering, evaluating & refining formulation
- Postdoctoral fellow Dr. Jeremy Eng integrated wastewater data into the PMCMC model and evaluated the results
- Dr. John Giesy, Dr. Yuwei Xie, Dr. Markus Brinkmann and Dr. Kerry McPhedran of the U Sask. Toxicology Centre provided the wastewater used, key background understanding concerning wastewater analysis, lab processes, wastewater infrastructure, shedding, and related areas
- The Saskatchewan Health Authority & Saskatchewan Ministry of Health made possible and realized provision of health system data, funded the development of the model, shaped its requirements, the infrastructure (special thanks to Dr. J. Basran & Jennifer Zerff)
- PHAC for the wastewater contract that accelerated this work, helped provide trainee support, as well as PHAC & FNIHB for supporting wastewater infras.
- NDO wants to extend special gratitude for SYK for inspiring this & related work

Shifting Gears: From “What” to “How”

- The previous section indicated the “target distribution” from which we have to sample
- This is a distribution $p(x_{0:k}^N | y_{1:k}^M)$ over trajectories $x_{0:k}^N$ of latent states of the system, given observations $y_{1:k}^M$
- This section examines the mathematics underlying the scheme for *how we actually go about sampling from that distribution*

How: Particle Dynamics: Two Components

- Recall: Each particle has its own full copy of model state (anything that could differ b/t realizations)
- **Prediction:** Between observations
 - Particles evolve according to standard [stochastic] model dynamics (just run the model on each particle's state)
 - Particle weights remain invariant
 - There is no filtering out of particles
- **Update:** At observation point
 - Particle weights are updated to reflect likelihood of observing the empirical data, given the particle state
 - If too much disparity in weights, particles are resampled according to their weights; tendency:
 - Particles w/ high weights reproduce; w/low weights disappear

Overall Algorithm

At time $t \geq 2$, perform a recursive update as follows

(1) Advance the sampled state by sampling $X_t^{(i)} \sim q_t(x_t | y_t, X_{1:t-1}^{(i)})$ and set $X_{1:t}^{(i)} = (X_{1:t-1}^{(i)}, X_t^{(i)})$;

(2) Update the weights to reflect the probabilistic and state update models $w_t^{(i)} =$

$$w_t^{(i)} = w_{t-1}^{(i)} \frac{p(X_t^{(i)} | X_{t-1}^{(i)}) g(y_t | X_t^{(i)})}{p(X_t^{(i)} | X_{t-1}^{(i)})}$$

g is the likelihood function: Gives likelihood of the empirical datum y_+ , given the model state X_+

Normalize the weights $W_t^{(i)} = \frac{w_t^{(i)}}{\sum_{i=1}^N w_t^{(i)}}$.

For any time t , if the effective sample size is too small (i.e., the variance of the weights is too high, $\frac{1}{\sum_{i=1}^N (W_t^{(i)})^2} < K$), resample $X_t^{(i)}$ and set $W_t^{(i)} = \frac{1}{N}$. Here K is a threshold value for the variation of the weights.

How: Prediction

(Dynamics Between Observations)

- Recall: Each particle has its own full copy of model state (anything that could differ b/t realizations)
- Each particle runs the model forward until the next observation
 - Originally identical particles diverge because of
 - Model stochastics
 - Distribution over some parameters
 - For our model, the particle evolution is governed by a stochastic differential equation
- Weights remain unchanged

Update Phase: Dynamics at Observations

- N: Count of state variables (stocks)
- M: Count of observations
- Because we can't easily draw a sample from the posterior distribution over trajectories $p(x_k^N | y_{1:k}^M, x_{k-1}^N)$, we use importance sampling
 - We actually capture the posterior by updating particle weights, and possibly resampling (see later)
 - This involves choosing a proposal distribution $q(x_k^N | y_t, x_{k-1}^N)$
- Please note that we will often go beyond the above and sample from the full trajectories $p(x_{0:k}^N | y_{1:k}^M)$ rather than simply from the value at time k

Sequential Importance Sampling for Our Case

- Importance-weighted samples are maintained over time
 - Such samples are termed “particles”
- Successive observations at integer times are made, with each such observation updating the underlying distribution $p\left(x_{0:k}^{N(i)} \mid y_{1:k}^M\right)$
 - Direct sampling from this distribution is not generally possible
- To enable successively sampling from $p\left(x_{0:k}^{N(i)} \mid y_{1:k}^M\right)$ as each observations k arrives, we draw instead from a proposal distribution $q(x)$ & successively update samples $x_{0:k-1}^{N(i)}$ & weights $w_{k-1}^{(i)}$ to reflect the observation, yielding $x_{0:k}^{N(i)}$ and $w_k^{(i)}$

Importance Sampling to Draw From Target

- Target distribution is $p \left(x_{0:k-1}^{N(i)} \mid y_{1:k-1}^M \right)$
- $x_{1:k-1}^{N(i)}$ is drawn from proposal distribution $q \left(x_{0:k-1}^{N(i)} \mid y_{1:k-1}^M \right)$
- Here we are seeking to choose and maintain a *proposal distribution* that can be readily sampled from over time
 - Per importance sampling principles, these samples are then weighted as

$$w_{k-1}^{(i)} = \frac{p \left(x_{0:k-1}^{N(i)} \mid y_{1:k-1}^M \right)}{q \left(x_{0:k-1}^{N(i)} \mid y_{1:k-1}^M \right)}$$

← Target distribution

← Proposal distribution

Convenient Choice in Proposal Distribution

• *For convenience, we assume that*

$$\triangle q(x_{0:k}^N | y_{1:k}^M) = q(x_k^N | x_{0:k-1}^N, y_{1:k}^M) q(x_{0:k-1}^N | y_{1:k-1}^M) \triangle$$

• These terms represent

- $q(x_{0:k-1}^N | y_{1:k-1}^M)$ is just the value for the proposal distribution at the time of the previous observations
- $x_k^{N(i)} \sim q(x_k^N | x_{0:k-1}^N, y_{1:k}^M)$: the probability (density) of the updated state taking into account the new measurement measurements at time k

Key Need: Weight Updates

- Given: A sample (particle) drawn with a weight $w_{k-1}^{(i)}$ from the importance-sampling approximated distribution $p(x_{0:k-1}^N | y_{1:k-1}^M)$ at time $k-1$
- Need: Formulate an equation to *update the weight of that particle* to draw from the target distribution $p(x_{0:k}^{N(i)} | y_{1:k}^M)$ for time. This update would take into account both
 - The model-borne dynamics mapping from x_{k-1}^N to x_k^N
 - The observation y_k^M

- Recall from earlier that we have

$$w_k^{(i)} = \frac{p(x_{0:k}^{N(i)} | y_{1:k}^M)}{q(x_{0:k}^{N(i)} | y_{1:k}^M)}$$

- Recall from earlier slide “Full Trajectory Distribution” that

$$p(x_{0:k}^N | y_{1:k}^M) \propto p(y_k^M | x_k^N) p(x_k^N | x_{k-1}^N) p(x_{0:k-1}^N | y_{1:k-1}^M)$$

- Recall further that we chose a proposal distribution of the form

$$q(x_{0:k}^N | y_{1:k}^M) = q(x_k^N | x_{0:k-1}^N, y_{1:k}^M) q(x_{0:k-1}^N | y_{1:k-1}^M)$$

- We thus have

$$w_k^{(i)} = \frac{p(y_k^M | x_k^N) p(x_k^N | x_{k-1}^N) p(x_{0:k-1}^N | y_{1:k-1}^M)}{q(x_k^N | x_{0:k-1}^N, y_{1:k}^M) q(x_{0:k-1}^N | y_{1:k-1}^M)}$$

Recursive Weight Updates

- Recall from the previous slide that

$$w_k^{(i)} = \frac{p(y_k^M | x_k^N) p(x_k^N | x_{k-1}^N)}{q(x_k^N | x_{0:k-1}^N, y_{1:k}^M)} \frac{p(x_{0:k-1}^N | y_{1:k-1}^M)}{q(x_{0:k-1}^N | y_{1:k-1}^M)}$$

- Recalling from a few slides earlier that

$$w_{k-1}^{(i)} = \frac{p(x_{0:k-1}^{N(i)} | y_{1:k-1}^M)}{q(x_{0:k-1}^{N(i)} | y_{1:k-1}^M)}$$

- We can recognize the ratio of the final terms in the numerator & denominator as $w_{k-1}^{(i)}$, and thus rewrite the equation as:

$$w_k^{(i)} = \frac{p(y_k^M | x_k^N) p(x_k^N | x_{k-1}^N)}{q(x_k^N | x_{0:k-1}^N, y_{1:k}^M)} w_{k-1}^{(i)}$$

Recursive Weight Updates Redux

- From the previous slide, we have

$$w_k^{(i)} = \frac{p(y_k^M | x_k^N) p(x_k^N | x_{k-1}^N)}{q(x_k^N | x_{0:k-1}^N, y_{1:k}^M)} w_{k-1}^{(i)}$$

- Because the process g_k is first-order Markovian, we can further simplify the denominator by recognizing that all impact of past state is captured in the previous state $x_{k-1}^{N(i)}$, and by assuming that all observations prior to k have already been reflected in the distribution, thus:

$$q(x_k^{N(i)} | x_{0:k-1}^{N(i)}, y_{1:k}^M) = q(x_k^{N(i)} | x_{k-1}^{N(i)}, y_k^M)$$

and (rearranging slightly) we have

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(y_k^M | x_k^{N(i)}) p(x_k^{N(i)} | x_{k-1}^{N(i)})}{q(x_k^{N(i)} | x_{k-1}^{N(i)}, y_k^M)}$$

Convenient Choice in Proposal Distribution

- *For convenience, we assume that*

$$\triangle q(x_{0:k}^N | y_{1:k}^M) = q(x_k^N | x_{0:k-1}^N, y_{1:k}^M) q(x_{0:k-1}^N | y_{1:k-1}^M)$$

- These terms represent

- $q(x_{0:k-1}^N | y_{1:k-1}^M)$ is just the value for the proposal distribution **at the time of the previous observations**
- $x_k^{N(i)} \sim q(x_k^N | x_{0:k-1}^N, y_{1:k}^M)$: the probability (density) of the updated state taking into account the new measurement measurements at time k

Recursive Weight Updates Redux

- We have $w_{k-1}^{(i)} = \frac{p(x_{0:k-1}^{N(i)} | y_{1:k-1}^M)}{q(x_{0:k-1}^{N(i)} | y_{1:k-1}^M)}$
- As demonstrated in the supplemental slides, we can then take advantage of several factors
 - The assumed form of the proposal distribution 
 - The formula for the probability of the trajectories $p(x_{0:k}^N | y_{1:k}^M)$
 - Cancellation
 - The fact that the process g_k is first-order Markovian (and thus all influences of past states come via the current state)
- The formula for the weight can then be formulated recursively as:

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(y_k^M | x_k^{N(i)}) p(x_k^{N(i)} | x_{k-1}^{N(i)})}{q(x_k^{N(i)} | x_{k-1}^{N(i)}, y_k^M)}$$

Recursive Weight Updates Redux

- Supplemental slides demonstrate that:

$$w_k^{(i)} = \frac{p(y_k^M | x_k^N) p(x_k^N | x_{k-1}^N)}{q(x_k^N | x_{1:k-1}^N, y_{1:k}^M)} w_{k-1}^{(i)}$$

- Because the process g_k is first-order Markovian, we can further simplify the denominator by recognizing that all impact of past state is captured in the previous state $x_{k-1}^{N(i)}$, and by assuming that all observations prior to k have already been reflected in the distribution, thus:

$$q(x_k^{N(i)} | x_{0:k-1}^{N(i)}, y_{1:k}^M) = q(x_k^{N(i)} | x_{k-1}^{N(i)}, y_k^M)$$

and (rearranging slightly) we have

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(y_k^M | x_k^{N(i)}) p(x_k^{N(i)} | x_{k-1}^{N(i)})}{q(x_k^{N(i)} | x_{1:k-1}^{N(i)}, y_k^M)}$$

Naïve Algorithm

Initialization (k=0)

- Sample $X_0^{N(i)}$ from $q_0(x_0^N | y_0^M)$
- For each particle i
 - $w_k^{(i)} = \frac{p(X_0^{N(i)})p(y_0^M | X_0^{N(i)})}{q(X_0^{N(i)} | y_0^M)}$
- For each particle i
 - $W_0^{(i)} = \frac{w_0^{(i)}}{\sum_{i=1}^{N_s} w_0^{(i)}}$

Ongoing Observations (k>0)

- For each particle i
 - Advance state via sampling $X_k^{N(i)} \sim q(x_k^N | y_k, X_{0:k-1}^{N(i)})$
 - Supplement trajectory $X_{0:k}^{N(i)} = (X_{0:k-1}^{N(i)}, X_k^{N(i)})$
 - Update weight $w_k^{(i)} = w_{k-1}^{(i)} \frac{p(y_k^M | x_k^{N(i)}) p(x_k^{N(i)} | x_{k-1}^{N(i)})}{q(x_k^{N(i)} | x_{k-1}^{N(i)}, y_k^M)}$
- For each particle i
 - $W_k^{(i)} = \frac{w_k^{(i)}}{\sum_{i=1}^{N_s} w_k^{(i)}}$

Choice of Proposal Distribution

- The algorithms above leave open to the implementer the choice of a proposal distribution

$$q(x_k^N | y_k, X_{0:k-1}^{N(i)})$$

- The choice of proposal distribution can have a sizeable impact on
 - The practical performance of the particle filtering
 - The complexity of the implementation
- Many practitioners make use of the “condensation algorithm” of Isard & Blake (1998), which employs a particularly simple proposal distribution

Condensation Algorithm

- Goes by additional names in the computational statistics literature
- Involves just using the prior distribution until the next point of observation, and then updating at that time using the likelihood
- In operational terms, this means that we simply
 - Run the model forward from the last data point (this samples from “the prior”) until the next observation point
 - At the time of the observation, multiply the weight of a particle by the value of the likelihood of observing the datapoint given the model state hypothesized by that particle
- It is not yet clear to the instructor if there is an effective

The “condensation algorithm”

(Isard & Blake 1998)

- Here, one chooses $q(x_k^{N(i)} | x_{k-1}^{N(i)}, y_k^M) = p(x_k^{N(i)} | x_{k-1}^{N(i)})$
 - In other words, the **proposal** distribution simply uses the simulation induced probability distribution and ignores the data

- Recall that, in general, for particle i

$$w_k^{(i)} = w_{k-1}^{(i)} \frac{p(y_k^M | x_k^{N(i)}) p(x_k^{N(i)} | x_{k-1}^{N(i)})}{q(x_k^{N(i)} | x_{k-1}^{N(i)}, y_k^M)}$$

We are just multiplying the old weight by the likelihood to get the new weight!

- For $q(x_k^{N(i)} | x_{k-1}^{N(i)}, y_k^M) = p(x_k^{N(i)} | x_{k-1}^{N(i)})$:

$$w_k^{(i)} = w_{k-1}^{(i)} \frac{p(y_k^M | x_k^{N(i)}) p(x_k^{N(i)} | x_{k-1}^{N(i)})}{p(x_k^{N(i)} | x_{k-1}^{N(i)})} = w_{k-1}^{(i)} p(y_k^M | x_k^{N(i)})$$

- Recall further that $p(y_k^M | x_k^{N(i)})$ is the **likelihood** of observing the empirical data at that time (y_t) given the particle state $x_k^{N(i)}$ at that time