

Bayesian Parameter Estimation of System Dynamics Models Using Markov Chain Monte Carlo (MCMC) Methods: An Informal Introduction

Nathaniel Osgood¹ osgood@cs.usask.ca

Juxin Liu² liu@math.usask.ca

¹Departments of Computer Science and Community
Health & Epidemiology; Laboratory for Computational
Epidemiology and Public Health Informatics

²Department of Mathematics and Statistics

University of Saskatchewan

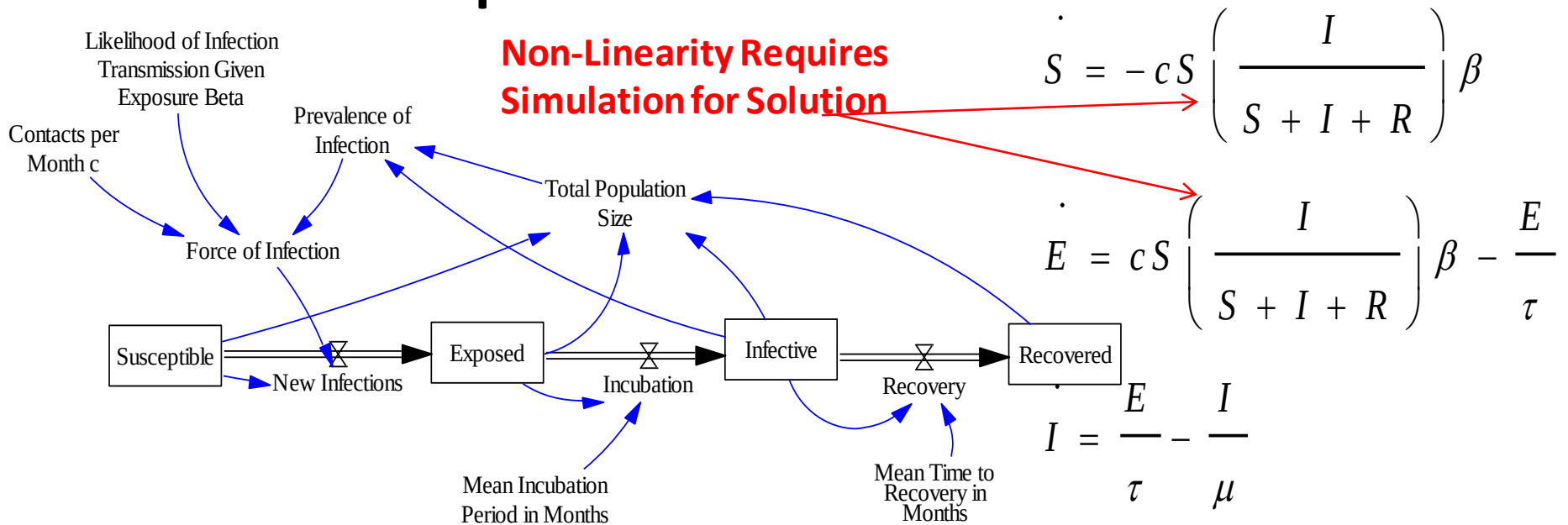
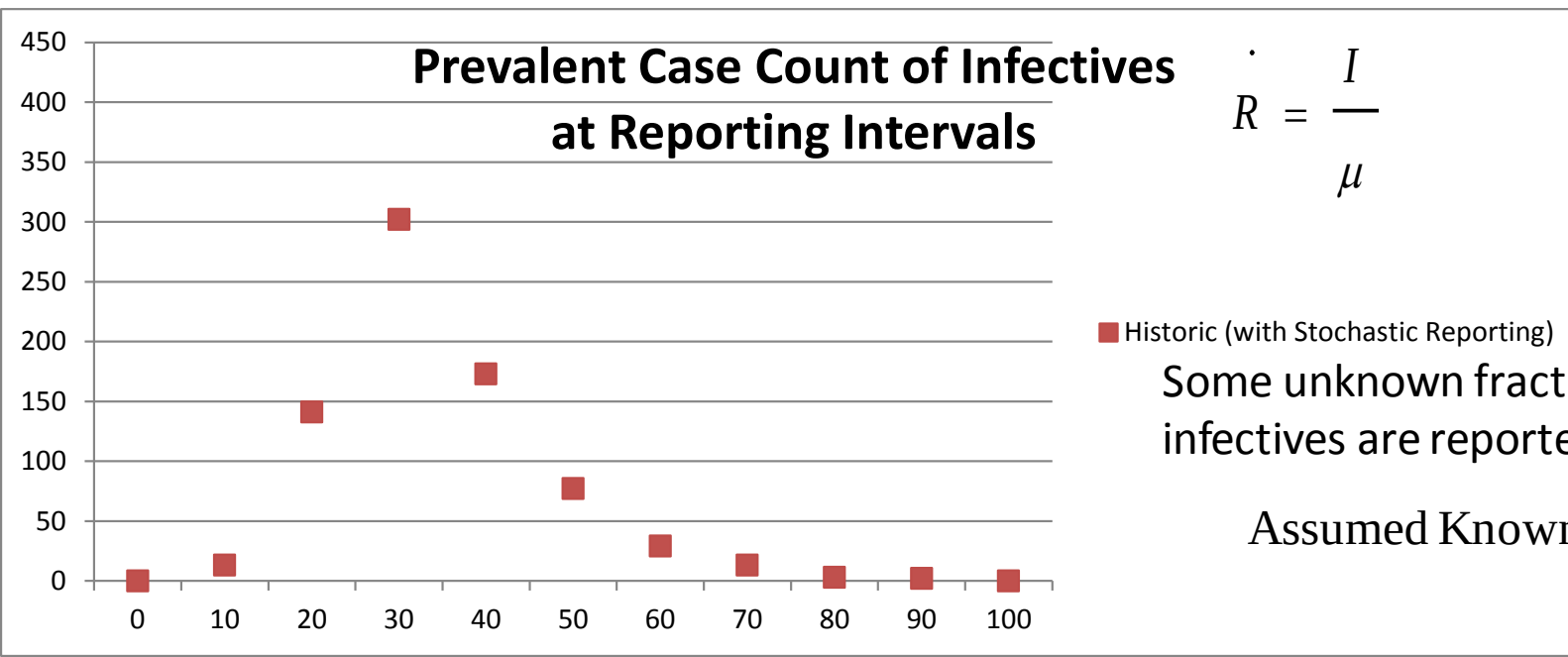
Calibration: “Triangulating” from Diverse Data Sources

- Often we don’t have reliable information on *some* parameters, but do have other data on the emergent behavior of system
 - Doesn’t relate to any one parameter in isolation, but to a combination influences
- Calibration involves “tuning” values of less well known parameters to best match observed data
- Frequently we learn from this that our model structure just can’t produce the patterns!

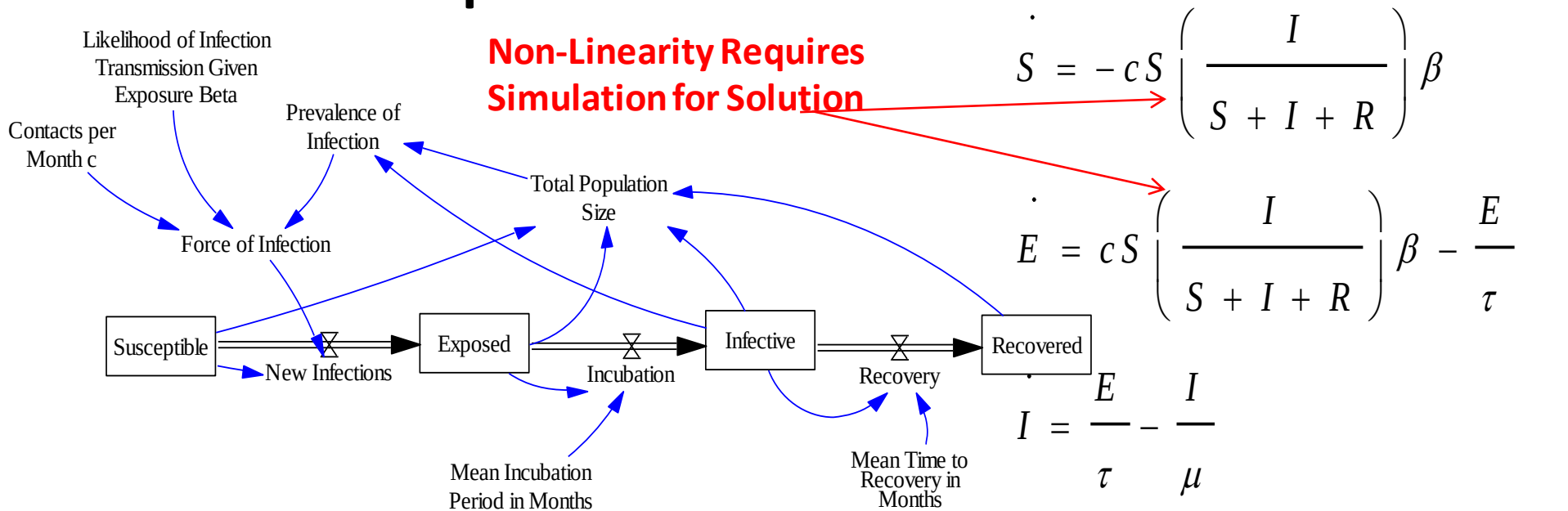
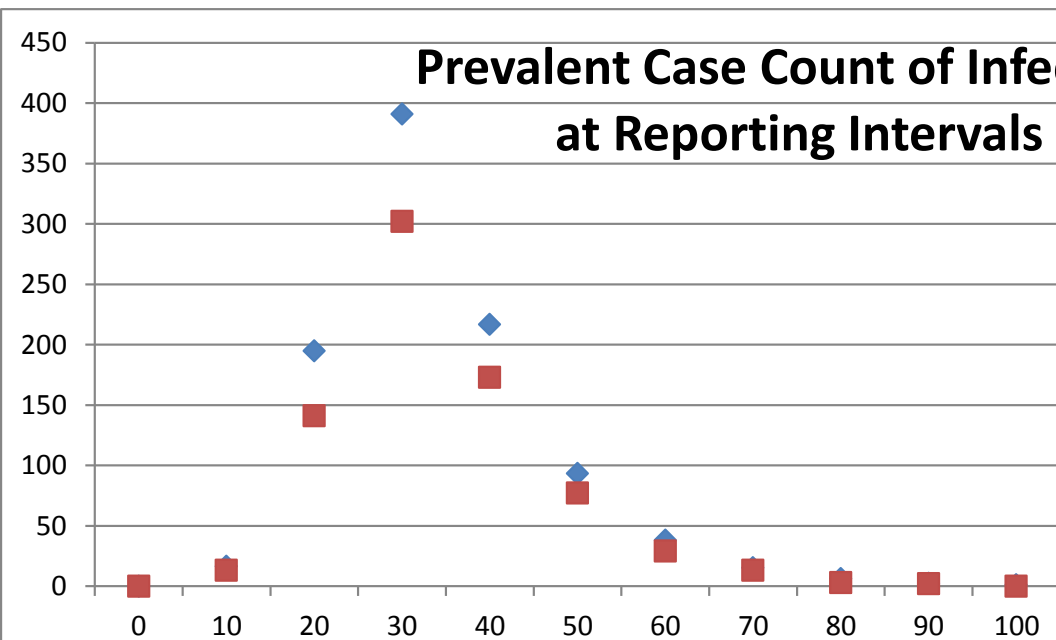
Traditional Parameter Estimation via Calibration in (Dynamic) Simulation Models

- Point estimation of parameter values
- Optional: Confidence intervals/covariance

Example: Classic SEIR Model



Example: Classic SEIR Model



Traditional calibration provides point estimates of parameters (potentially with covariance matrix and/or confidence intervals)

Some unknown fraction p of infectives are reported

Assumed Known: β, τ, μ
Assumed Unknown: $p=0.8, c=10$

Traditional Parameter Estimation via Calibration in (Dynamic) Simulation Models

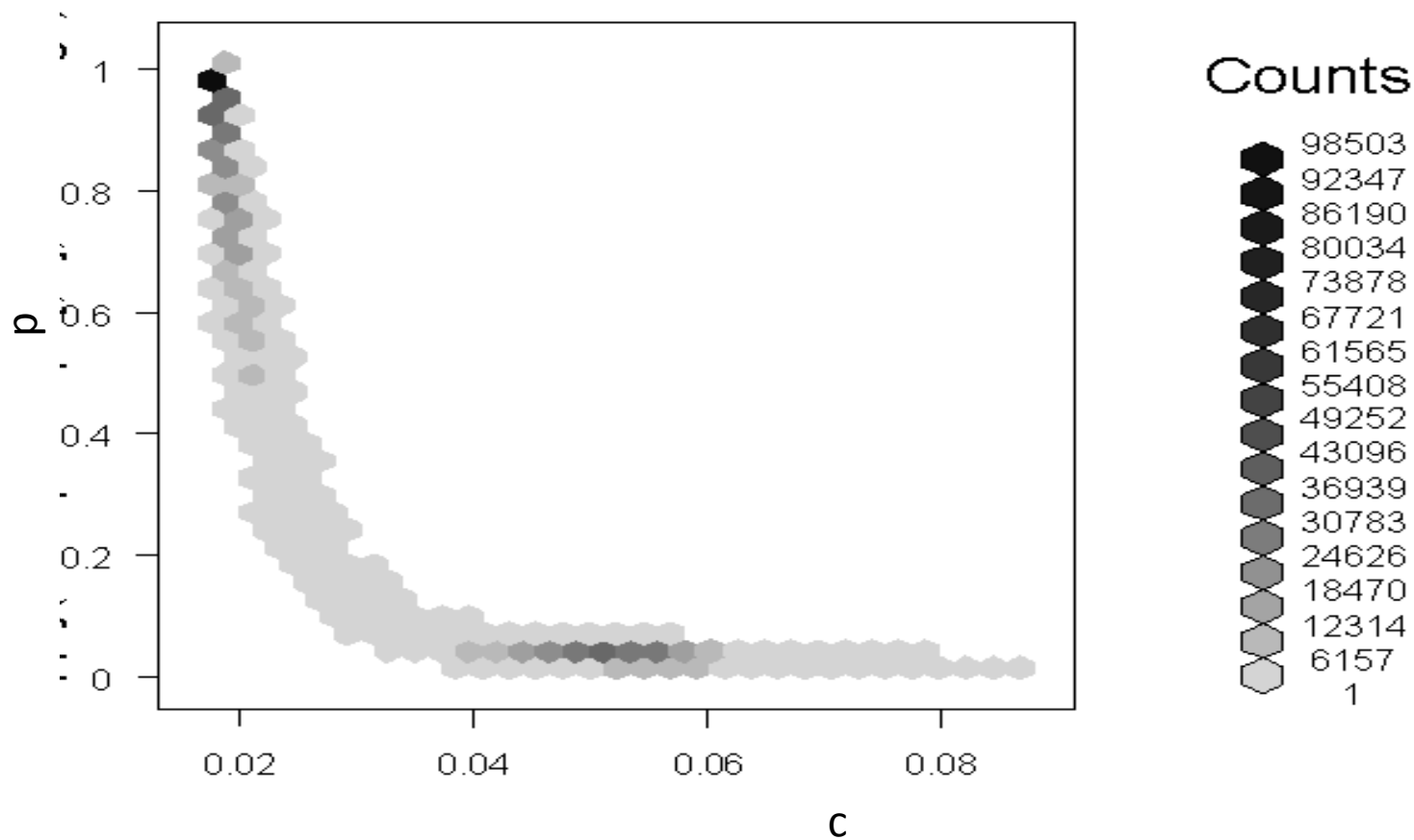
- Point estimation of parameter values
- Optional: Confidence intervals/covariance
- Limitations
 - Single point estimate gives no sense of variability
 - Even given estimate of dispersion (e.g. confidence intervals, covariance matrix), strong limitations
 - Unimodality assumption
 - Doesn't make apparent how this yields dispersion across certain output measures e.g. model variables, comparison b/t interventions
 - Poor understanding of global shape of parameter distribution
 - Error weighting (e.g. least squares) imposes distributional assumptions on errors
 - Single dynamic model assumed

Posterior Distributions for Parameters:

Moving Beyond Local Estimates

- A posterior distribution over parameters specifies the relative probability (density) of parameter vals in light of
 - The observed data
 - One or more models under consideration
 - Any pre-existing expectations of the relative likelihood of different parameter values (a prior distribution)
- If can sample from posterior, can derive many things, e.g.
 - Point (MLE, mean, Minimum var, Max *a posteriori*, etc.) or dispersion estimates of parameters & other model variables
 - Likelihood intervention A is “better” than intervention B
 - “Credibility intervals” for diverse model output
 - Visualizations of parameter space

Example of an asymmetric, multi-modal and non-linear posterior (different context)

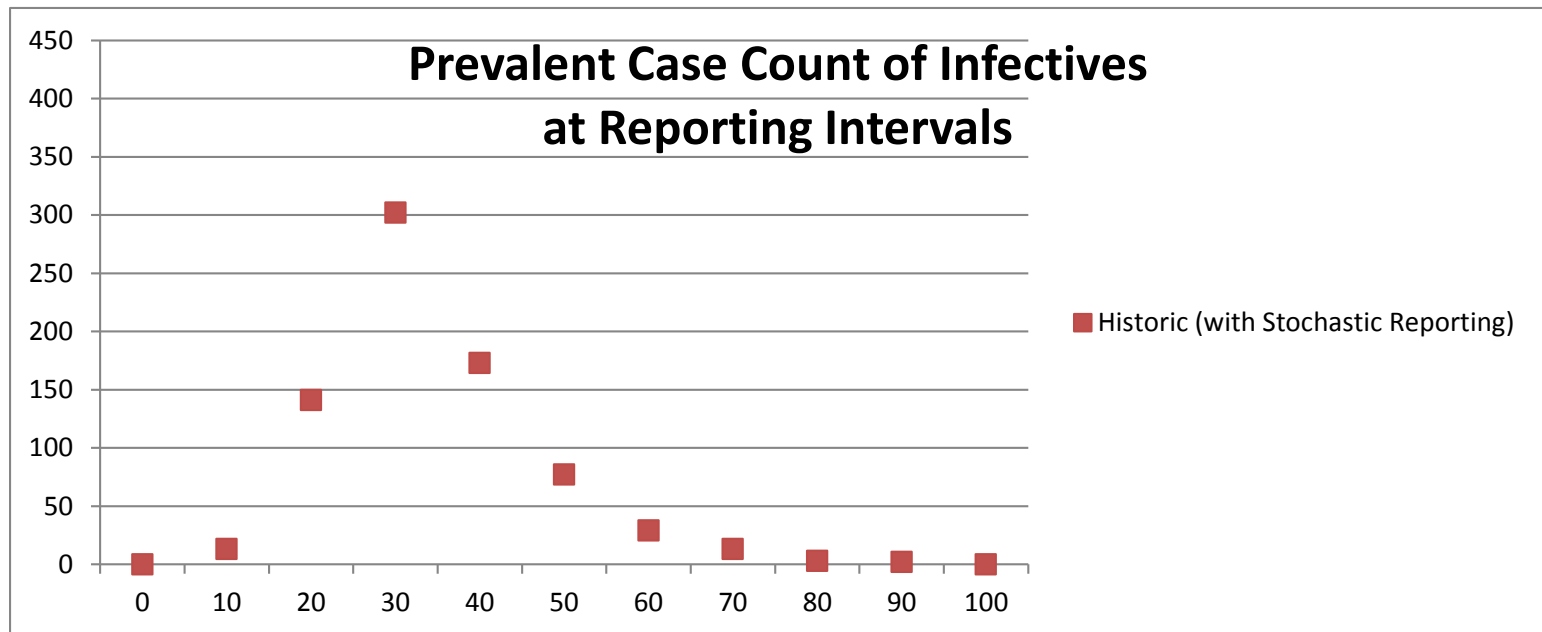


Notation

- θ : Vector of parameter values (i.e., a particular mapping between parameters and values)

– Here, $\vec{\theta} \equiv \begin{bmatrix} c \\ p \end{bmatrix}$

- y : Observed data



Markov Chain Monte Carlo Helps us Reliably Sample Parameter Posterior Distributions

- MCMC is a principled way of generating samples from θ according to a posterior distribution over θ
- These samples from the posterior distribution of θ are generated over and over again $\Rightarrow$ Monte Carlo
 - For each such sample, we can e.g.,
 - Accumulate statistics on functions of this parameter set (θ)
 - θ itself
 - Model outputs (applying model to parameters – i.e. to θ)
 - Intervention gains over baseline for output (subtract value of baseline scenario for parameter values θ from that of intervention for same θ)
 - Record the sample for later use (as a valid draw from the posterior distribution)
- A “Markov Chain” is used to generate the θ samples

High-Level Operation of MCMC for Parameter Estimation

- Elements to specify before algorithm
 - Assign prior distributions (over parameters, models, etc.)
 - May or may not be parametric (if so, via ‘hyperparameters’)
 - Formulate likelihood formulae that indicate relative likelihood of seeing particular values of empirical data (y) in light of values of parameters (θ) & simulation model output
 - These could be either parametric or non-parametric
 - Apply Bayes’ rule to get rule for probability (density) of parameter θ in terms of data (y) and priors
- MCMC: A Markov Chain is used to generate samples of parameters (θ), with (asymptotic stationary) distribution proportional to posterior distribution
 - For each generated parameter sample, we sample output measures of interest & accumulate statistics, etc.

High-Level Operation of MCMC for Parameter Estimation

- Elements to specify before algorithm
 - Assign prior distributions (over parameters, models, etc.)
 - May or may not be parametric (if so, via ‘hyperparameters’)
 - Formulate likelihood formulae that indicate relative likelihood of seeing particular values of empirical data (y) in light of values of parameters (θ) & simulation model output
 - These could be either parametric or non-parametric
 - Apply Bayes’ rule to get rule for probability (density) of parameter θ in terms of data (y) and priors
- MCMC: A Markov Chain is used to generate samples of parameters (θ), with (asymptotic stationary) distribution proportional to posterior distribution
 - For each generated parameter sample, we sample output measures of interest & accumulate statistics, etc.

Prior Parameter Distributions

- Before calibration/MCMC estimates, we often have some sense as to likely ranges of parameter values
 - We can specify this with a “prior” distribution
 - This is called a “prior” because we can formulate it prior to observing the data
- Examples
 - Minimally informative (e.g. Uniform) Prior
 - Parametrically expressed prior
- Aside: It is possible to use MCMC in a non-Bayesian fashion absent a prior (here we sample from the sampling distribution)

Our Prior Parameter Distributions

- Here, we have two parameters of interest: $\vec{\theta} \equiv \begin{bmatrix} c \\ p \end{bmatrix}$
- For our example, we compare consequences on the posterior of using 2 priors

– **Uninformative:** Minimally informative (Uniform) prior

$$p(\vec{\theta}) = \begin{cases} \frac{1}{50}, & 0 \leq c \leq 50, 0 \leq p \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

– **Misleading:** Highly biased parametrically expressed prior

$$p(\vec{\theta}) \sim \text{LogNormal}(c; \ln(5), \ln(2)) \text{Triangle}(p; 0, 1)$$

High-Level Operation of MCMC for Parameter Estimation

- Elements to specify before algorithm
 - Assign prior distributions (over parameters, models, etc.)
 - May or may not be parametric (if so, via ‘hyperparameters’)
 - Formulate likelihood formulae that indicate relative likelihood of seeing particular values of empirical data (y) in light of values of parameters (θ) & simulation model output
 - These could be either parametric or non-parametric
 - Apply Bayes’ rule to get rule for probability (density) of parameter θ in terms of data (y) and priors
- MCMC: A Markov Chain is used to generate samples of parameters (θ), with (asymptotic stationary) distribution proportional to posterior distribution
 - For each generated parameter sample, we sample output measures of interest & accumulate statistics, etc.

Likelihood Function Formulae

- A likelihood function $P(y|\theta)$ must be specified by MCMC model creator to assess likelihood of seeing particular empirical data (y) values in light of parameters values (θ)
- This function represents a key factor in judging the relative posterior probability of different possible sets of parameter values (θ) for a model
 - Different formulations than product should also be considered
- For our example, we treat there as being some probability p that a given case will be reported an a given reporting period (for simplicity here, we assume that this chance is independent period-to-period) $\Rightarrow$ Assume a binomial distr.

For our case, this is from the simulation model results for θ (i.e., for $\theta_1=c$)

$$p \left(\vec{y} \middle| \vec{\theta} \right) = \prod_{i=0}^{10} \text{BinP} \left(y_{10i}; I_{10i}, p \right) = \prod_{i=0}^{10} \binom{I_{10i}}{y_{10i}} p^{y_{10i}} (1-p)^{I_{10i}-y_{10i}}$$

High-Level Operation of MCMC for Parameter Estimation

- Elements to specify before algorithm
 - Assign prior distributions (over parameters, models, etc.)
 - May or may not be parametric (if so, via ‘hyperparameters’)
 - Formulate likelihood formulae that indicate relative likelihood of seeing particular values of empirical data (y) in light of values of parameters (θ) & simulation model output
 - These could be either parametric or non-parametric
 - Apply Bayes’ rule to get rule for probability (density) of parameter θ in terms of data (y) and priors
- MCMC: A Markov Chain is used to generate samples of parameters (θ), with (asymptotic stationary) distribution proportional to posterior distribution
 - For each generated parameter sample, we sample output measures of interest & accumulate statistics, etc.

Prior & Likelihood $\Rightarrow$ Posterior Value

- By Bayes' Rule $p(\vec{\theta} | \vec{y}) = \frac{p(\vec{y} | \vec{\theta}) p(\vec{\theta})}{p(\vec{y})}$

This is the *likelihood function*

This is the *prior distribution*

- If we consider this as a function of $\vec{\theta}$ (with respect to a fixed $\vec{y}$), we have

$$p(\vec{\theta} | \vec{y}) \propto p(\vec{y} | \vec{\theta}) p(\vec{\theta})$$

- Reminder:

$$p(\vec{y} | \vec{\theta}) = \prod_{i=0}^{10} \text{BinP}(y_{10i}; I_{10i}, p) = \prod_{i=0}^{10} \binom{I_{10i}}{y_{10i}} p^{y_{10i}} (1-p)^{I_{10i}-y_{10i}}$$

$$p(\vec{\theta}) = \begin{cases} \frac{1}{50}, & 0 \leq c \leq 50, 0 \leq p \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

Why Do We Want to Use a Markov Chain to Generate the Samples θ ?

- Simple to generate samples from distributions where cumulative distribution function CDF is known in advance
 - Draw random number x from uniform distribution
 - Return parameters in CDF that gives a cumulative probability at value x
- Can't typically use above to sample from posterior
 - We don't know the CDF up front (it depends on simul.!)
 - Often dealing with such a high dimensional θ space that trying to derive the CDF would entail huge effort
- Markov Chain approaches lets us generate samples of the parameters (θ) from posterior without knowing it or its CDF in advance

Rough Markov Chain Operation

- The Markov Chain generation process involves
 - Drawing a candidate value θ^* from θ (e.g. just generated from random perturbation around last generated sample
 - Below formulation assumes symmetric perturbation
 - Assessing (relative) posterior of this candidate value of θ
 - Run the simulation model
 - Required b/c formulated likelihood statements for $P(y | \theta^*)$ generally depend on simulation model output (for that value of parameters θ^*)
 - Compute relative posterior of candidate θ^* using $P(y | \theta^*)P(\theta^*)$
 - $P(\theta)$ is just the value of the prior at this point
 - $P(y | \theta)$ is in general calculated using not just θ and y , but also outputs and intermediate values of simulation model produced for value of θ
 - Accept or reject this candidate θ based on ratio between this relative posterior at θ vs. last generated sample
 - If accepted, emit this as our new sample!
 - If rejected, emit existing value and go on to the next candidate!
- Run through “burn in” time before using results

Takes
significant
work!



Schematic of Random Walk Metropolis

Hastings Algorithm (1 walker)

$i=0$

Pick a value for θ_0 // can draw from crude dist.

Until convergence criterion is satisfied

Pick a random value for perturbation $\Delta \theta$

Create a candidate value $\theta = \theta_i + \Delta \theta$

if $\text{uniform}(0,1) \leq \min \left\{ 1, \frac{\text{Posterior}(\theta)}{\text{Posterior}(\theta_i)} \right\} /*$ NB: can calc even if only know posterior up to a constant */

$\theta_{i+1} = \theta$ //Accept

else

$\theta_{i+1} = \theta_i$ // Reject (keep same sample for emission)

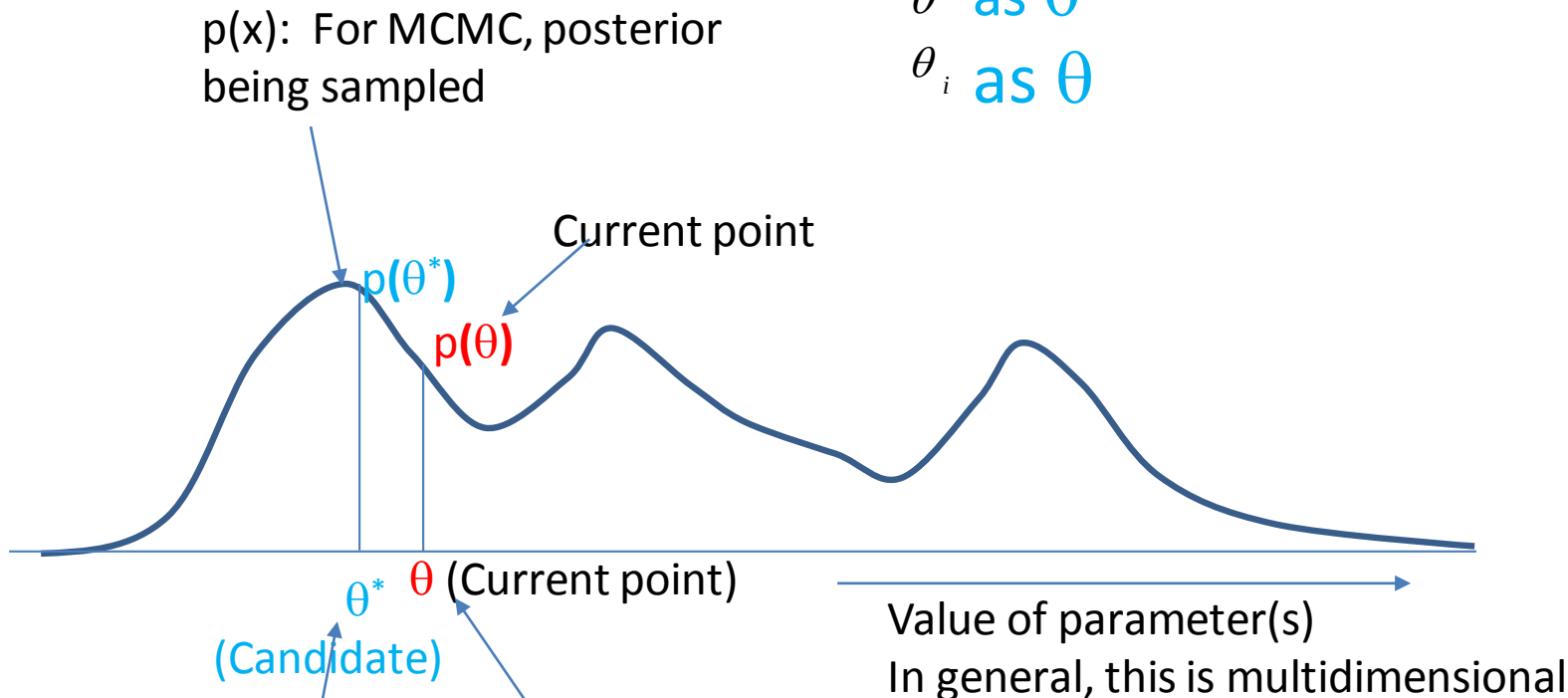
Emit θ_{i+1} as next sample // emit regardless if was accepted

$i=i+1$

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

For simplicity, we denote

θ as θ^*
 θ_i as θ



We move from θ to θ^* with probability $\min \left(1, \frac{Posterior(\theta^*)}{Posterior(\theta)} \right)$

NB: Even if $p(\theta^*) < p(\theta)$,
 May still transition current pos. from θ to (candidate) $\theta^* \Rightarrow$ **aids repeated explore entire distribution**

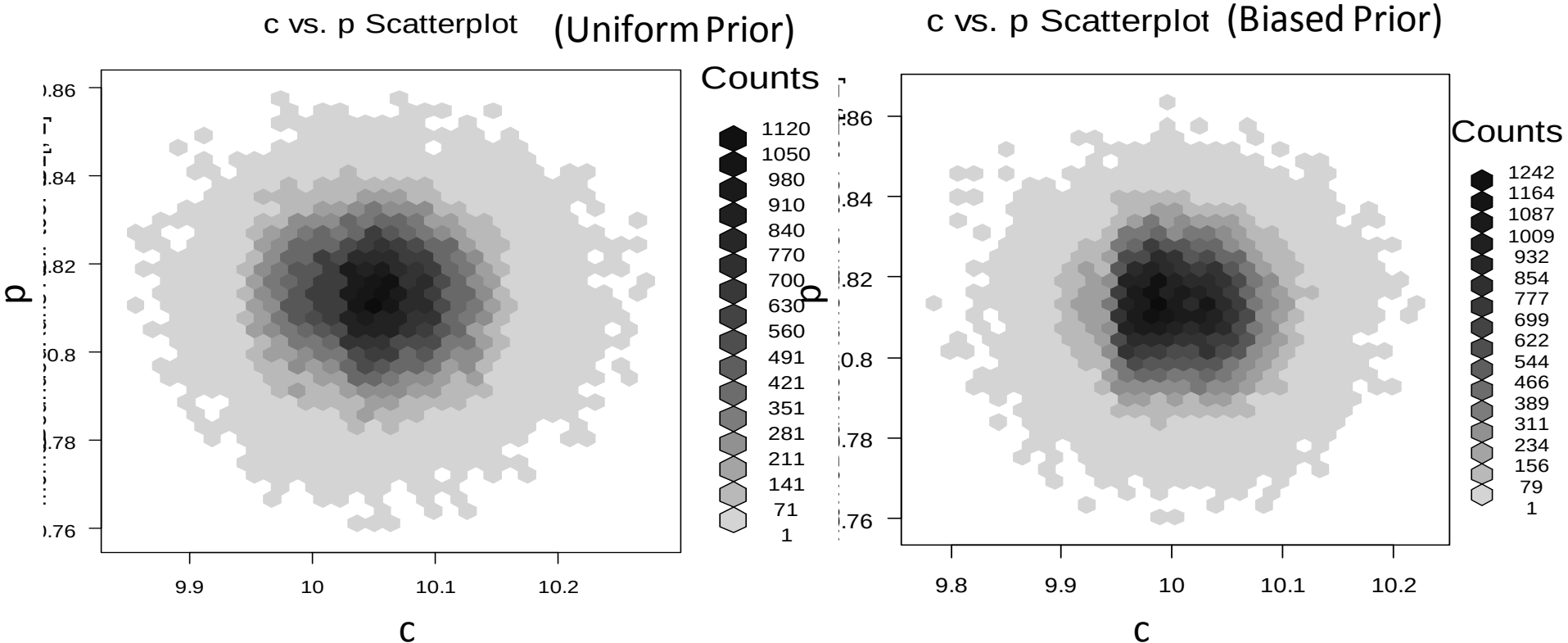
Computational Effort

- Running the simulation model often takes a large computational effort (often 100,000s of runs)
 - Typically far more than generating random values or other elements of the MCMC “core” algorithm!
- We have to run the model many times beyond the values of θ that are actually yielded as samples
 - Through burn-in period (could be 100,000s of runs)
 - In rejected samples
 - Model still has to be run to get output to assess likelihood that observed data could results from this set of parameter value
 - Not actually among generated points
- Much of this work (e.g. different walkers) can be done in parallel (concurrently on diff. machines)

Our Case

- We implemented an R-Vensim interface to permit R & R package MCMCpack to call Vensim
- We drew 100,000 MCMC iterations, following 20,000 burn-in iterations
- The MCMC samples were used in 2 SEIR scenarios (baseline, hygienic intervention w/lower β)

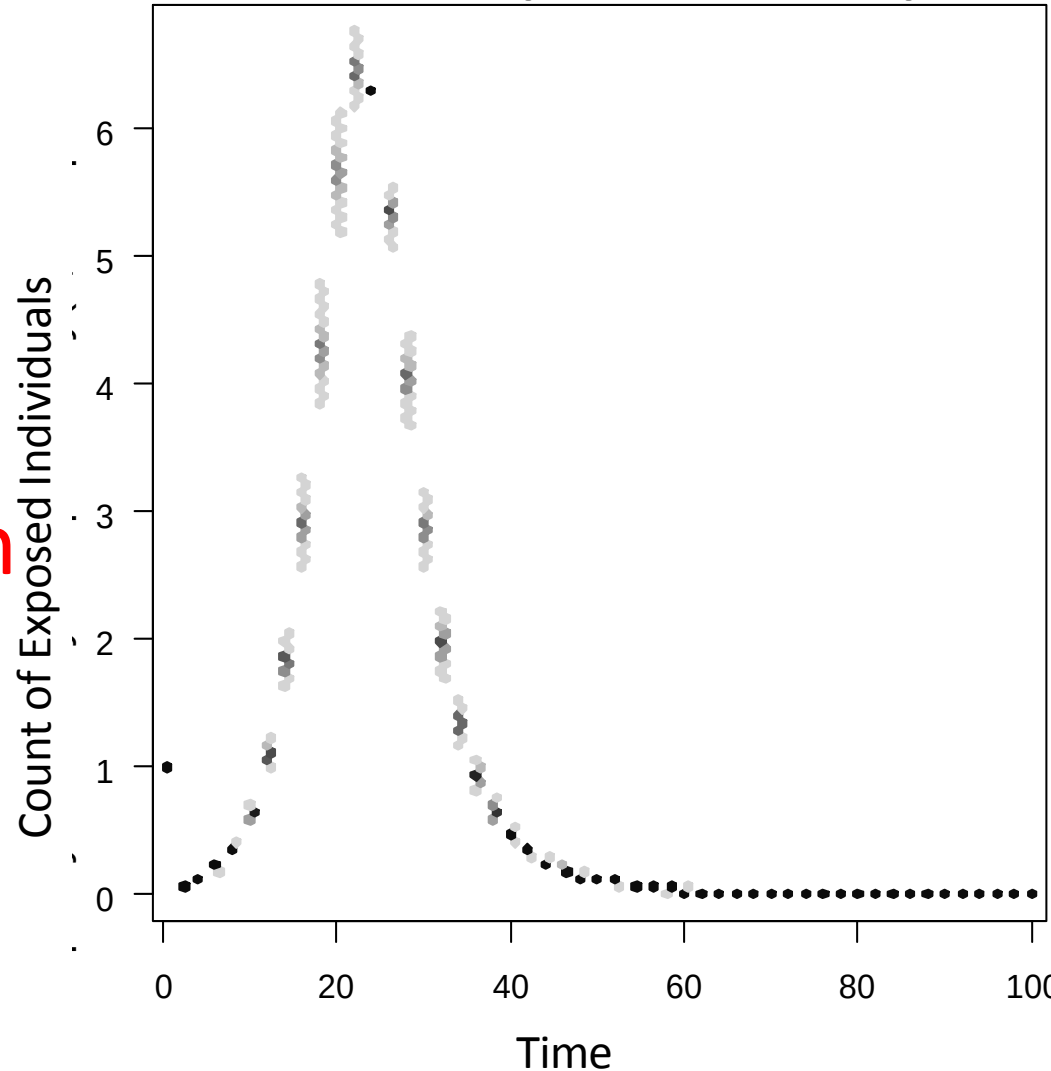
Results: Posterior Distribution Samples for Parameters for Distinct Priors



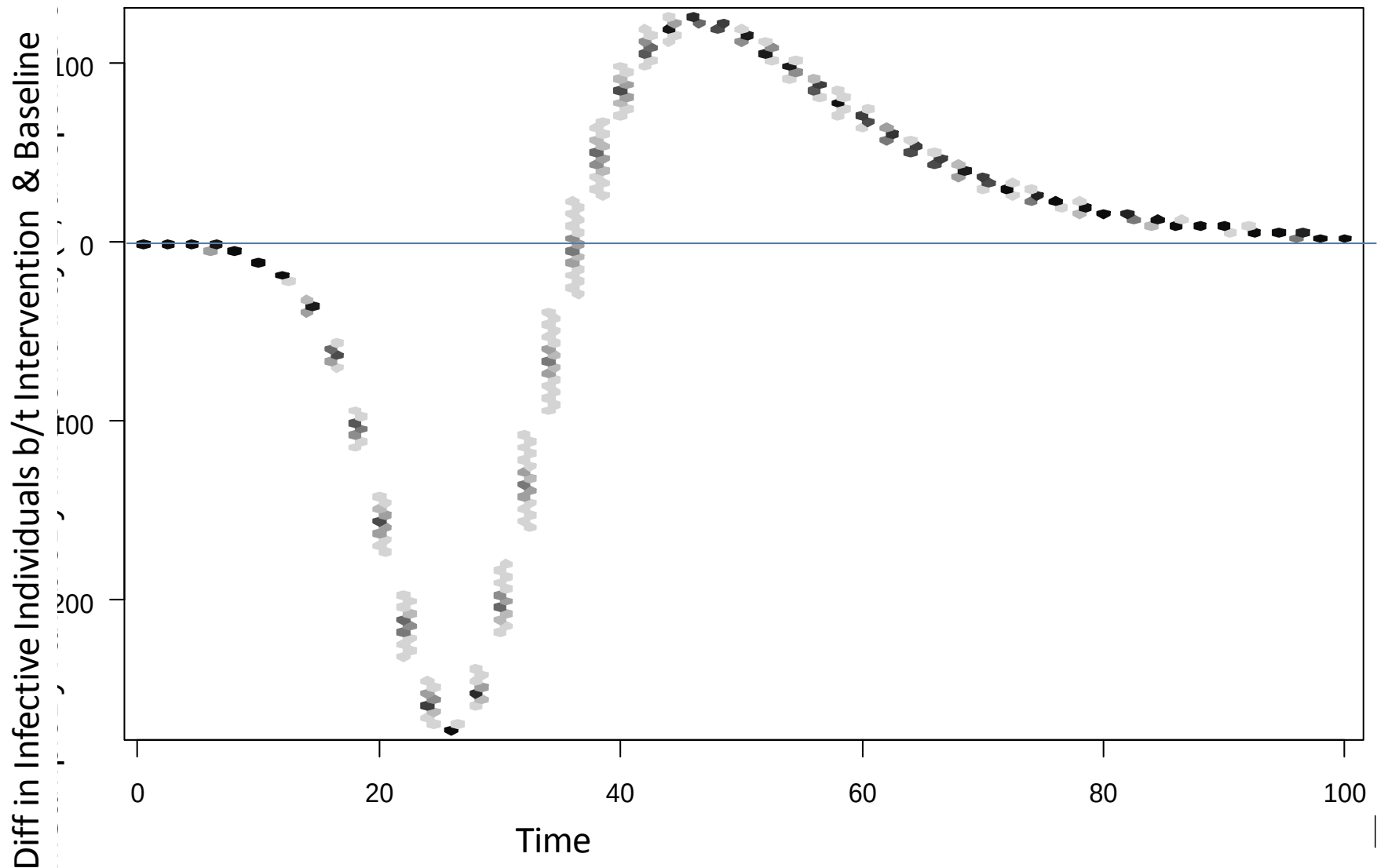
Deriving Posteriors over Variables [here, Exposed] Samples from Posterior Distribution over Parameters (Baseline)

- Here, we just run the D model with each value sampled from the posterior

Deriving confidence intervals around such output is generally not possible for conventional calibration results



Posteriors for differences in infective case count b/t baseline & intervention scenarios



Interweaving Calibration and Sensitivity Analysis

- Sampling the posterior distribution permits a natural form of sensitivity analysis
 - We can explore robustness with respect to calibration-induced uncertainty
 - Parameters
 - Outputs
 - Cross-scenario comparisons
- Further forms of sensitivity analysis provide additional key insights

Role of Simulation Model(s)

- In simulation-model based parameter estimation, the simulation model(s) serves as a critical mediator between parameters and observed data
 - The model helps us project behavioral consequences of parameter assumptions (in light of dynamic hypotheses captured by those model(s))
 - The estimated likelihood that the output data will be observed is typically conditional on such simulation model outputs rather than directly on the parameters θ
 - This process can also be used with stochastic simulation models (but would typically require many more Monte Carlo runs to deal with variability in sim. output given θ)
- Id of sim model can just be another parameter in θ , but situation is more complex if θ differs b/t models

Practical Concerns 1

- Burn-in time
 - Need to run for long enough that starts generating values approximately distributed as posterior
 - There are ways of judging this
- Tuning algorithm parameters (e.g. random walk perturbation size) to yield medium acceptance rate
 - Recommendations vary (50%; 1D: 0.44, n-D: 0.23)
 - High enough to explore (don't just remain in same place)
 - Not so high that just going around in local area of optimum, without significantly improving estimate
 - If old and new ratios are close to 1, likelihood of accepting new estimate is close to 1

Practical Concerns 2

- High covariance b/t parameters lowers efficiency
 - Do iteration over residuals using regression
- Easiest if operating in unbounded space: Transform
- Sample thinning rate: To reduce correlation between successive samples, accept only 1 out of every n samples
 - This will help draws from posterior be closer to independent draws - which is what is being sought
- Assessing Convergence
 - Here, combine statistics between walkers with those from the same walker to get an estimate of how much would be gained by further sampling
 - If within-walker statistics are similar to those between walkers, this is a sign that may be doing good job in exploring space

Many MCMC Algorithms

- There are many variants of MCMC algorithms
- These differ in
 - Range of applicability (e.g. Gibbs sampling is associated with successive conditional formulation of model)
 - Simplicity (e.g. Random walk Metropolis-Hastings) vs. Sophistication e.g.
 - Choosing between models of different dimension – e.g. Reverse Jump MCMC
 - Allowing asymmetric random perturbations, asymmetric candidate draws from random distribution (Metropolis-Hastings vs. Metropolis)

Broader Context of Monte Carlo Methods

- MCMC is a batch (“offline”) Monte Carlo algorithm
 - Processes all data at once to yield posterior
- There is a popular class of Sequential Monte Carlo (“online”) methods (most notably, Particle Filtering)
 - Goal conceptually similar to that of (Extended) Kalman filtering, but impose far less restrictive assumptions
 - Here, at discrete points over time, take new data and formulate new estimates incorporating that data
- Rejection sampling: Simpler technique (no Markov Chain) when can bound the posterior distribution

Conclusions

- MCMC approaches provide a more general alternative to traditional calibration for **deterministic** models
 - Allows computing posterior distributions of any model variable (e.g. exposed, infected, incident cases) for one scenario or between scenarios
 - Capacity to formally define probabilistic model in MCMC fosters clarity in distributional assumptions
- MCMC permits greater robustness in model conclusions in light of the associated uncertainties
 - Sampling from the posterior distribution allows for a natural form of sensitivity analysis with respect to calibration-induced uncertainty in parameters, outputs, &c.
- With stochastic models & to estimate underlying state rather than parameter estimation, use SMC/PF
- For estimating both parameters & state, use PMCMC