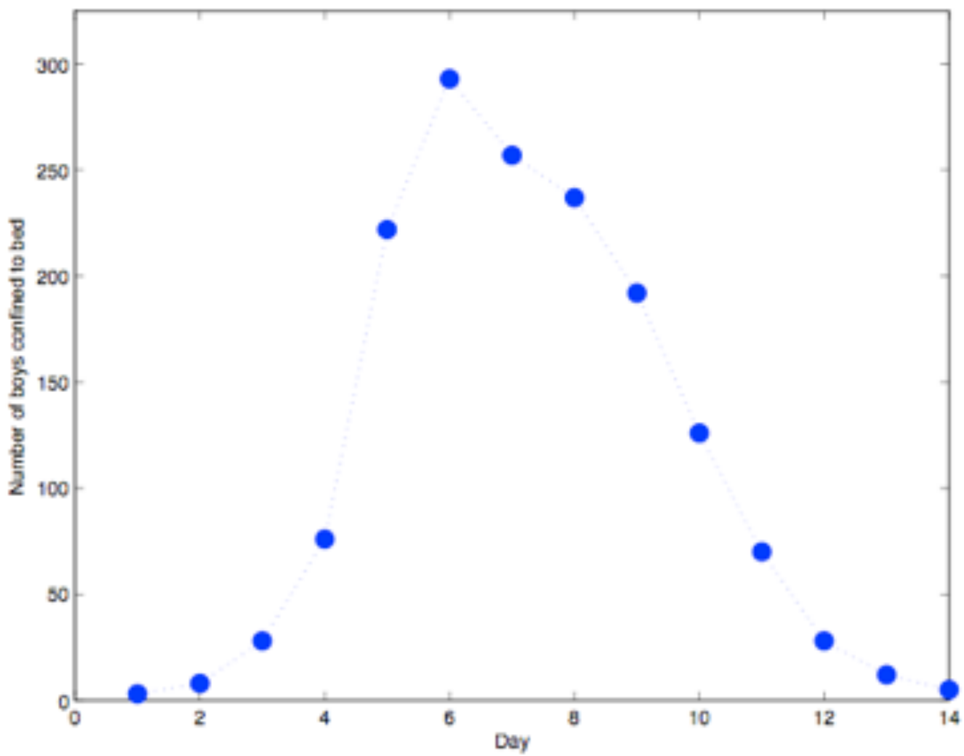
Inference and filtering

Model-based inquiry

Boarding School outbreak



Boarding School, England*
Jan 1978

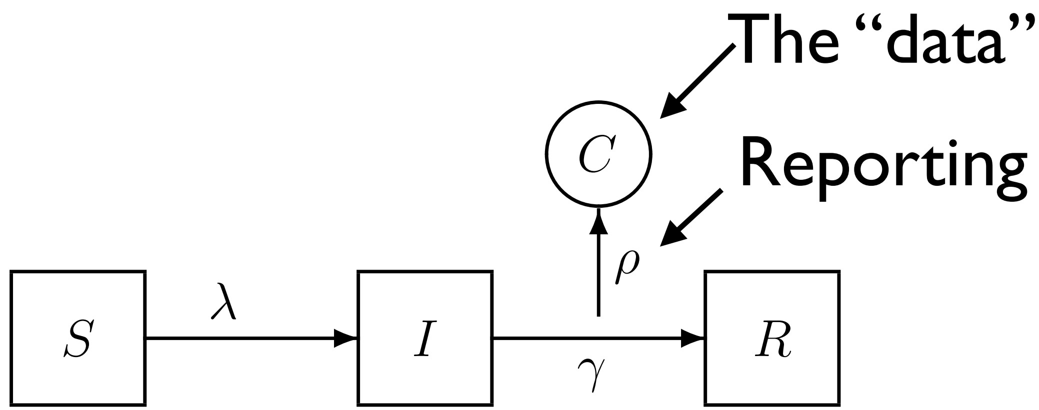
What is R_0 for this epidemic?

**legal disclaimer: not actual boarding school*

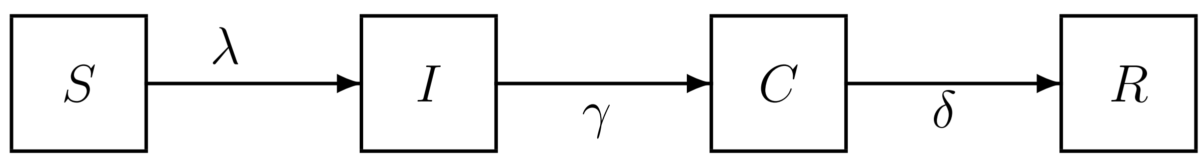
Model-based inquiry

$$\lambda = \frac{\beta I}{N}$$

Model 1



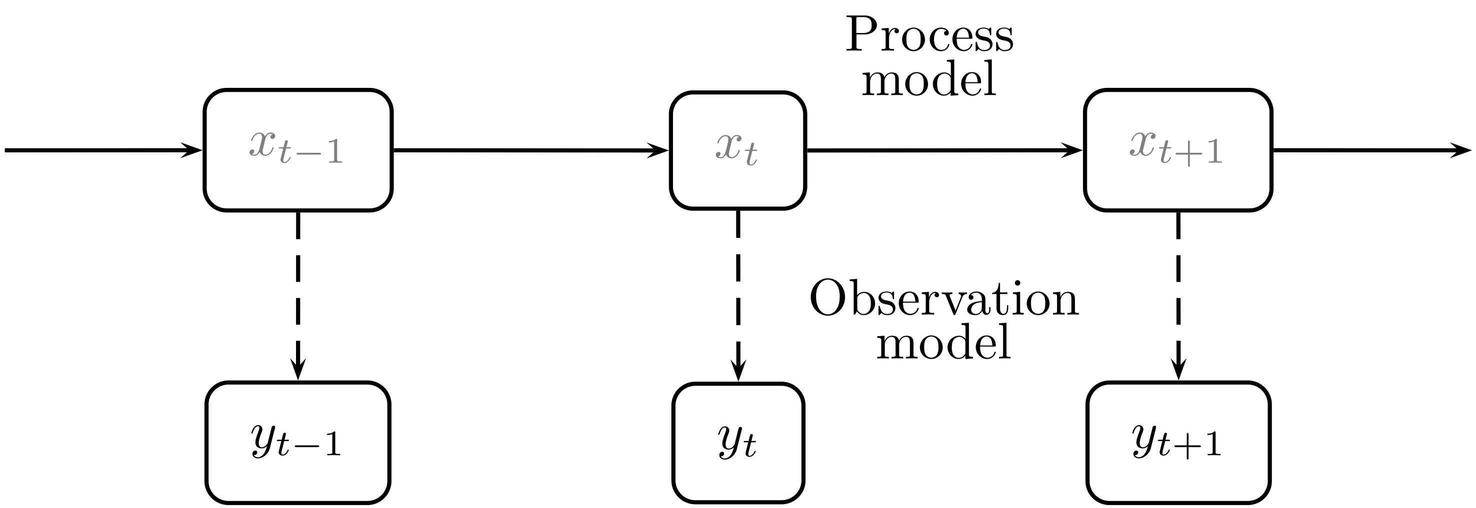
Model 2



which model?

Likelihood for Mechanistic Models

- "Hidden Markov" models, "State Space" models or "Partially Observed Markov Processes" (POMPs)
- Distinguish between "process" and "observation"



Known $P(x_t|x_{t-1}), P(y_t|x_t)$

Properties of Maximum Likelihood Estimators

- Consistency - biases disappear with increasing data
- Efficiency - with large amount of data, MLE has smallest uncertainty of any estimator
- Natural scale - 2 units of log-likelihood per degree of freedom corresponds (roughly) to “statistical significance”

Likelihood ratio testing

- Given two (nested) models, it has been shown (by Samuel Wilkes) that quantity -2 log (Θ_1/Θ) is $\chi 2$ distributed
- If comparing models that differ by 1 parameter, can test if log-likelihoods differ by $\chi 2(1)/2 \sim 2$

Properties of Maximum Likelihood Estimators

- Consistency - biases disappear with increasing data
- Efficiency - with large amount of data, MLE has smallest uncertainty of any estimator
- Natural scale - 2 units of log-likelihood per degree of freedom corresponds (roughly) to "statistical significance"

- Caveats
  - sometimes, don't have enough data
  - likelihood can be hard to compute

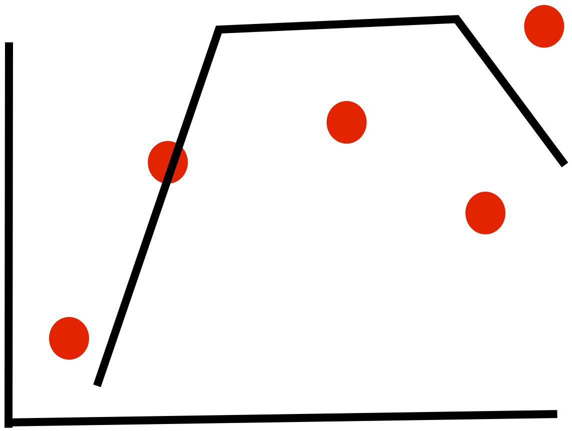
Likelihood & estimation

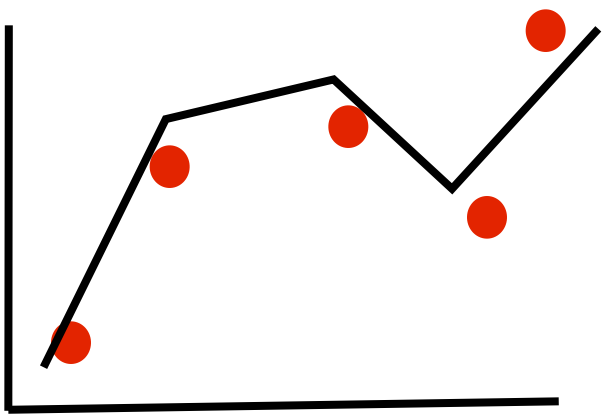
- Assume we have data, y, and model state, x (both are vectors containing state variables). Model predictions generated using set of parameters, θ

- Transmission dynamics subject to
  - "process noise": heterogeneity among individuals, random differences in timing of discrete events (environmental and demographic stochasticity)
  - "observation noise": random errors made in measurement process itself

Likelihood & estimation

- If we ignore process noise, then model is deterministic and all variability attributed to measurement error
- Observation errors assumed to be sequentially independent
- Maximizing likelihood in this context is called ‘trajectory matching’





Deterministic likelihood

- In absence of process noise, state of system at time *t*, x_t, is known function
- So, likelihood is just product of (independent) probability of "observing" each data point y_t

$$\mathcal{L}(\theta) = P(y_{1:T}|\theta) = \prod_{t=1}^{T} P(y_t|x_t, \theta)$$

$$ll(\theta) = log\mathcal{L}(\theta) = \sum_{t=1}^{T} log(P(y_t|x_t, \theta))$$

Deterministic likelihood

- If we assume measurement errors are normally distributed, with mean μ and variance σ^2 then

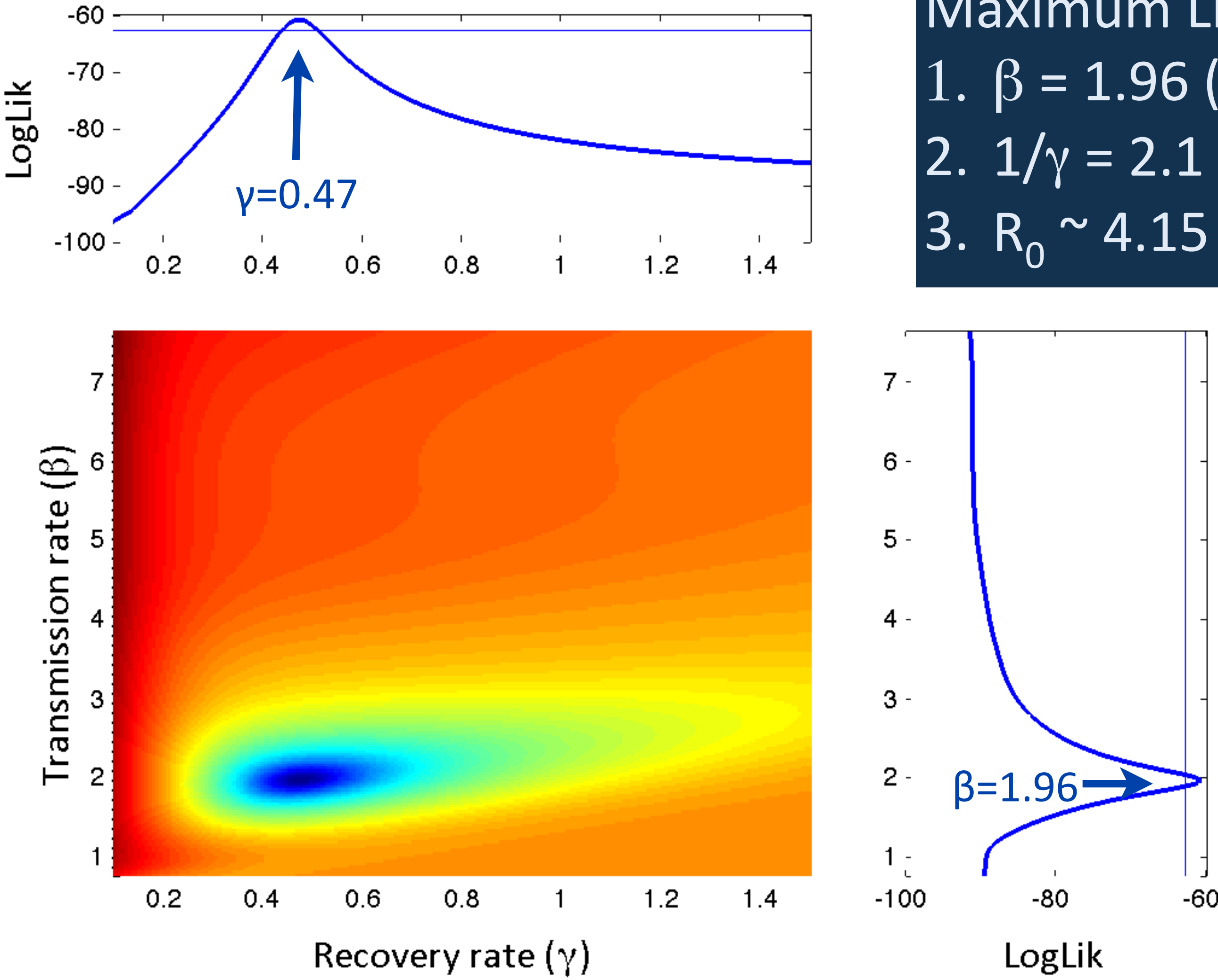
=length of data

$$ll(\theta) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{t=1}^{T}(y_t - x_t)^2$$

=SSE/n

=SSE

Model estimation: Influenza outbreak



Maximum Likelihood Estimates:

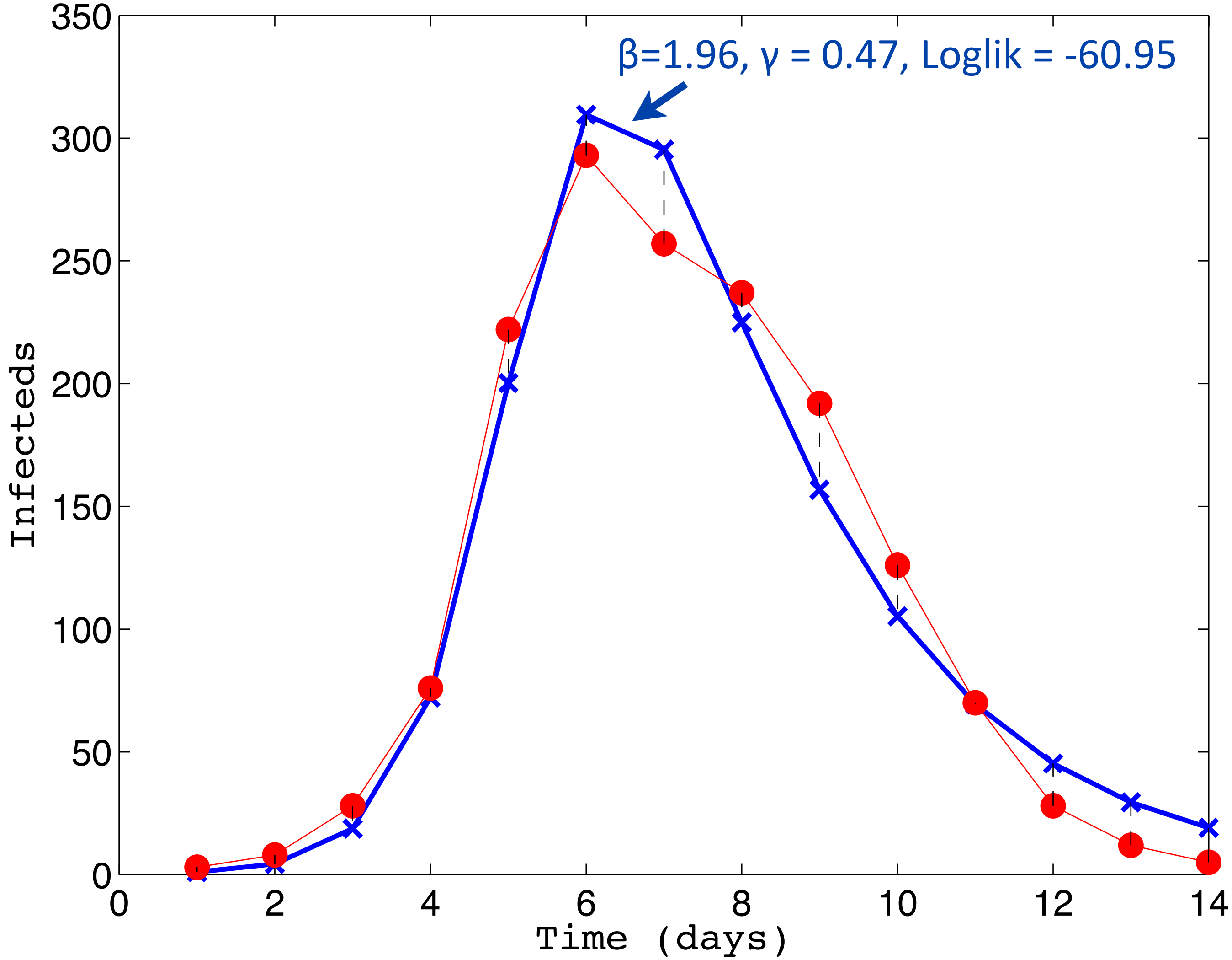
1. β = 1.96 (per day)
2. $1/\gamma$ = 2.1 days
3. $R_0 \sim 4.15$

Recall 2 log-likelihood units indicate significant difference

Can use likelihood profiles to put confidence intervals on estimates

β=1.96 (1.90,2.04)
γ=0.47 (0.43,0.50)

Likelihood estimation



Likelihood for Stochastic Models

- We know transmission is stochastic, so how to compute likelihood now?

$$\mathcal{L}(\theta) = P(y_{1:T}|\theta) = \sum_{x_1}\sum_{x_2}\cdots\sum_{x_T}\prod_{t=1}^{T} P(y_t|x_t,\theta)P(x_t|x_{t-1},\theta)$$

- Can use Monte Carlo methods to approximate likelihood

Monte Carlo theorem

- If $a_1, a_2, \ldots, a_n \sim p$ (where p is some pdf) and iid, and b_i (i=1,…,N) be independent and distributed according to p then

$$\hat{\mu}_N = \frac{1}{N}\sum_{i=1}^{N} f(b_i)$$

is a Monte Carlo estimator of $E(f(a))$

$$\hat{\mu}_N = \frac{1}{N}\sum_{i=1} f(b) \approx E[f(a)] = \int f(a)p(a)da$$

Monte Carlo approximation

- Use stochastic model to produce *N* trajectories each of length *T*. Call these $x_{t,k}$ (k=1,…,N)
- Compute likelihood for each trajectory

$$\mathcal{L}_k(\theta) = \prod_{t=1}^{T} P(y_t | x_{t,k}, \theta) P(x_{t,k} | x_{t-1,k}, \theta)$$

- By Monte Carlo theorem,

$$\mathcal{L}(\theta) \approx \frac{1}{N} \sum_k \frac{\mathcal{L}_k(\theta)}{w_k}$$

Monte Carlo approximation

$$\mathcal{L}(\theta) \approx \frac{1}{N} \sum_k \frac{\mathcal{L}_k(\theta)}{w_k}$$

- Here, w_k represents probability of proposing each trajectory k
- Simple choice for w_k : $$w_k = \prod_{t=1}^{T} P(x_{t,k} | x_{t-1,k}, \theta)$$

Recall: $$\mathcal{L}_k(\theta) = \prod_{t=1}^{T} P(y_t | x_{t,k}, \theta) P(x_{t,k} | x_{t-1,k}, \theta)$$

$$= \frac{1}{N} \sum_{k=1}^{N} \frac{\prod_{t=1}^{T} P(y_t | x_{t,k}, \theta) P(x_{t,k} | x_{t-1,k}, \theta)}{\prod_{t=1}^{T} P(x_{t,k} | x_{t-1,k}, \theta)}$$

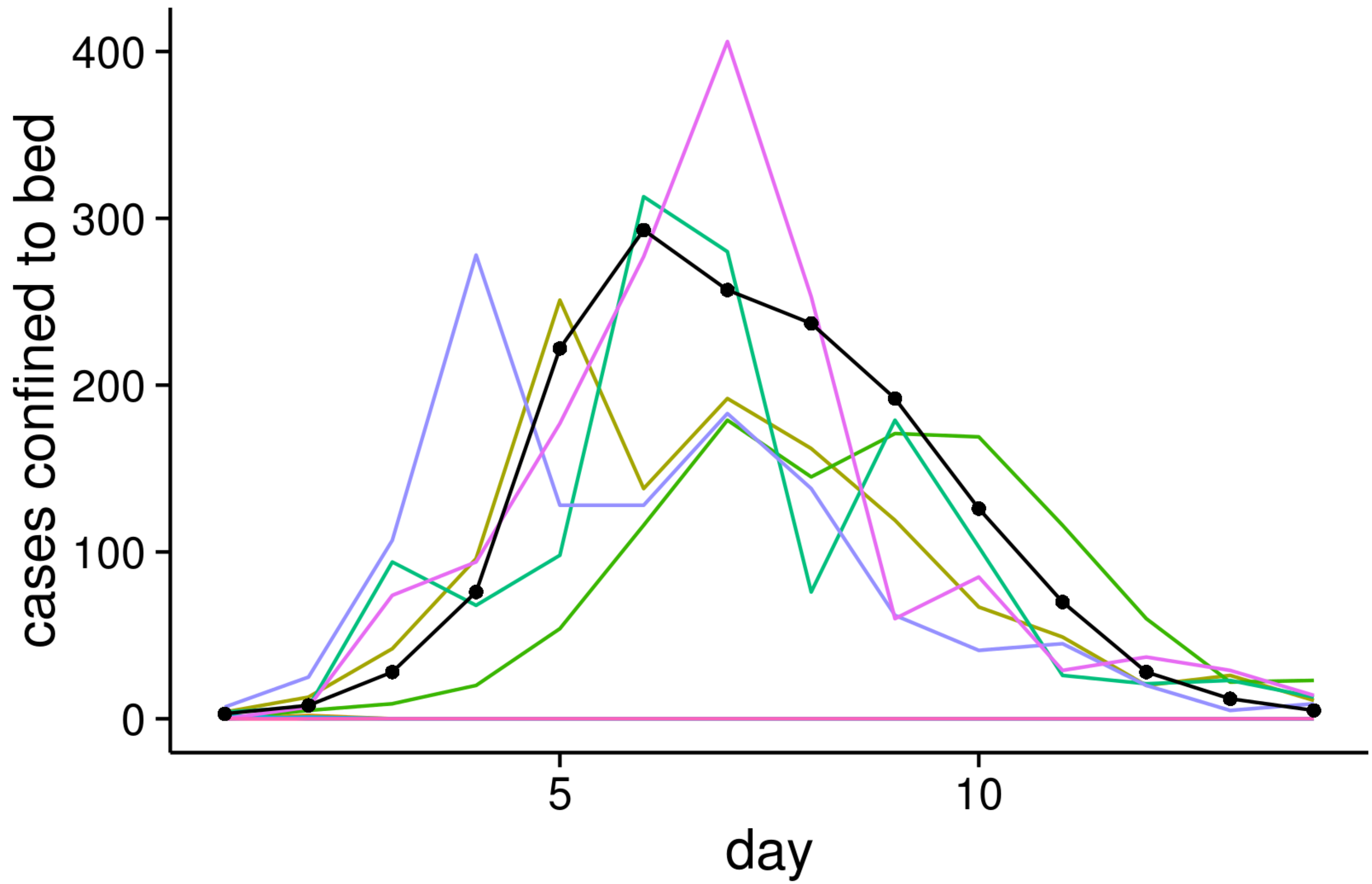
Monte Carlo approximation

$$= \frac{1}{N}\sum_{k=1}^{N} \frac{\prod_{t=1}^{T} P(y_t|x_{t,k},\theta)P(x_{t,k}|x_{t-1,k},\theta)}{\prod_{t=1}^{T} P(x_{t,k}|x_{t-1,k},\theta)}$$

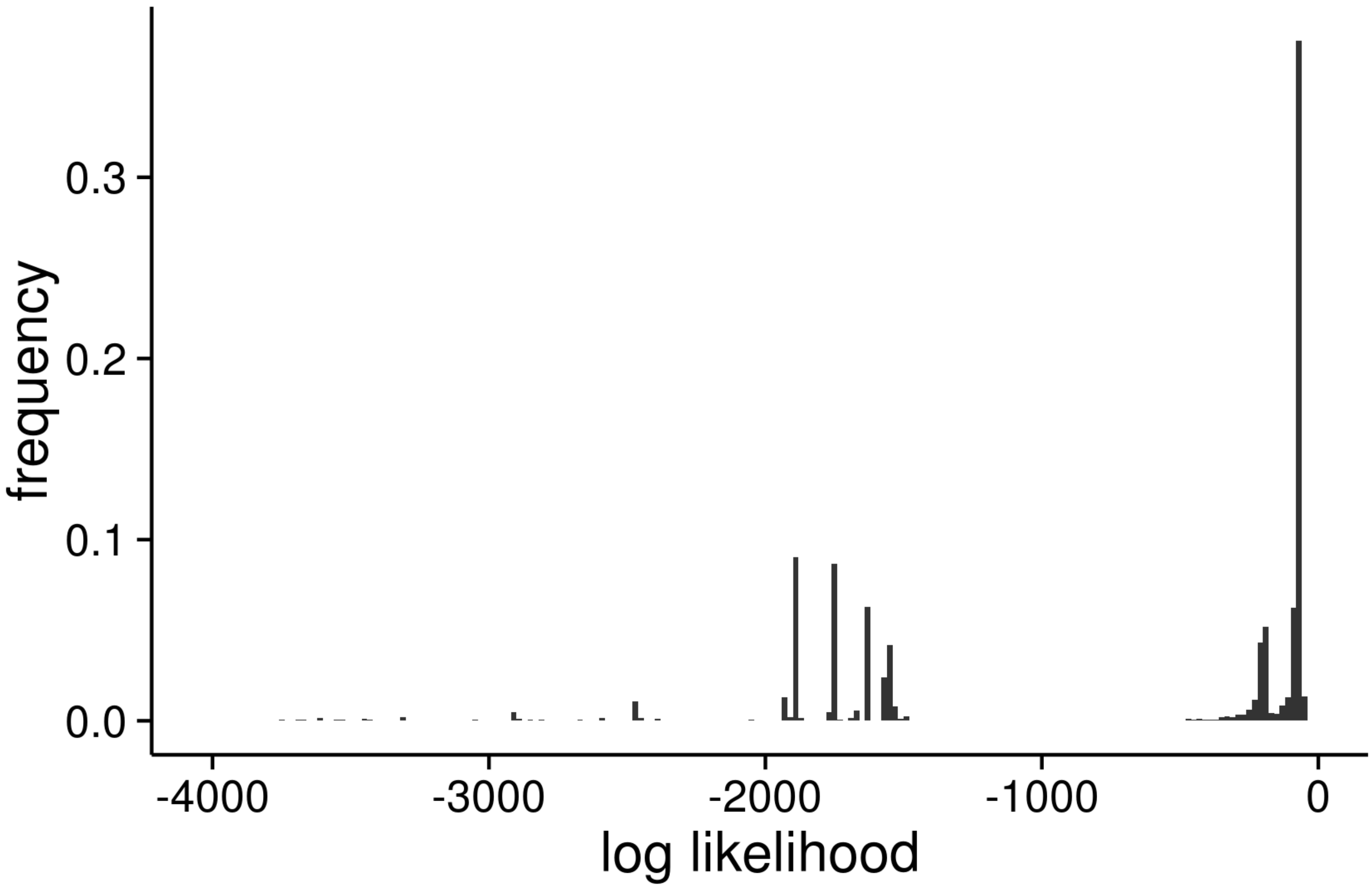
$$= \frac{1}{N}\sum_{k=1}^{N}\prod_{t=1}^{T} P(y_t|x_{t,k},\theta)$$

- This means, we generate trajectories by simulation, then simply compute likelihood of data for each trajectory and average

Stochastic epidemics



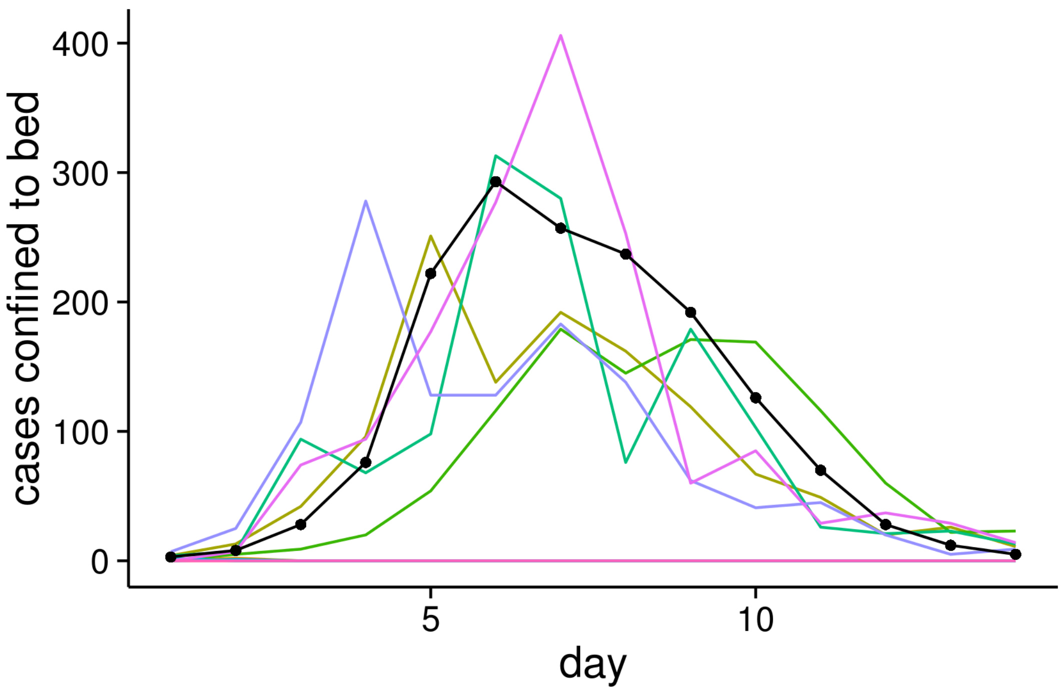
Stochastic epidemics



Estimated log-likelihood based on 5000 simulations: -63 ± 16

Error in estimate is very high ⇒ estimate is imprecise

What's the problem?



- Too many trajectories don't pass near data
- Also, when trajectory diverges from data, it almost never comes back
- Due to fact that we are *proposing* trajectories unconditional on the data
- Problem will get worse with longer data sets

Importance Sampling

- If $a_1, a_2, \ldots, a_n \sim p$ (where p is some pdf) and iid, and b_i (i=1,…,n) be independent and distributed according to p then

$$\hat{\mu}_n = \frac{1}{n} \sum_i f(b_i)$$

is a Monte Carlo estimator of $E(f(a))$

- What if it's difficult to sample directly from distribution of a?

Importance Sampling

- What if we generate random samples from another (easier to simulate) distribution and make suitable correction?
- Wish to compute E(*f*(*a*)), where *a*~*p* but difficult/ impossible to sample from *p*
- Suppose *q* is a probability distribution that is easy to simulate (called *proposal distribution*)
- Now, notice that

$$E(f(b)) = \int f(a)p(a)da = \int f(a)\frac{p(a)}{q(a)}q(a)da$$

Importance Sampling

- From Monte Carlo theorem

$$E(f(b)) \approx \frac{1}{N}\sum_i f(c_i)\frac{p(c_i)}{q(c_i)}$$

where c_i are iid random variables drawn from q

- Let $w_i = p(c_i)/q(c_i)$ be *importance weights*, then

$$E(f(b)) \approx \frac{\sum_i w_i f(c_i)}{\sum_i w_i}$$

The particle filter

- Let's factorize likelihood differently (=more efficiently)

$$\mathcal{L}(\theta) = P(y_{1:T}|\theta) \;= \prod_{t=1}^{T} P(y_t|y_{1:t-1},\theta)$$

$$= \prod_{t=1}^{T} \sum_{x_t} P(y_t|x_t,\theta) {\color{blue}P(x_t|y_{1:t-1})}$$

Markov property

$${\color{blue}P(x_t|y_{1:t-1},\theta)} = \sum_{x_{t-1}} P(x_t|x_{t-1},\theta) {\color{red}P(x_{t-1}|y_{1:t-1},\theta)}$$

The particle filter

$$P(x_t|y_{1:t-1},\theta) = \sum_{x_{t-1}} P(x_t|x_{t-1},\theta)P(x_{t-1}|y_{1:t-1},\theta)$$

Prediction distribution

Filtering distribution

Baye's Theorem

$$P(x_{t-1}|y_{1:t-1},\theta) = P(x_t|y_t, y_{1:t-1},\theta)$$

$$= \frac{P(y_t|x_t,\theta)P(x_t|y_{1:t-1},\theta)}{\sum_{x_t} P(y_t|x_t,\theta)P(x_t|y_{1:t-1},\theta)}$$

Particle filter to compute likelihood

Generate $x^F_{t,k}$ from $P(x_t|y_{1:t},\theta)$ and $x^P_{t,k}$ from $P(x_t|y_{1:t-1},\theta)$

- Draw $x^P_{t,k}$ by one-step simulation from $x^F_{t-1,k}$
- Generate $x^F_{t,k}$ by resampling according to $P(y_t|x_t,\theta)$
- Monte Carlo theorem tell us conditional likelihood is

$$\mathcal{L}_t(\theta) = P(y_t|y_{t-1},\theta) = \sum_{x_t} P(y_t|x_t,\theta) P(x_t|y_{1:t-1},\theta)$$

$$\approx \frac{1}{N}\sum_k P(y_t|x^P_{t,k},\theta)$$

Particle filter to compute likelihood

Can iterate procedure through data, alternatively simulating and sampling until *t*=*T*

The full log-Likelihood is then approximately

$$\log(\mathcal{L}(\theta)) = \sum_t \log(\mathcal{L}_t(\theta))$$

Particle filter: Matlab code

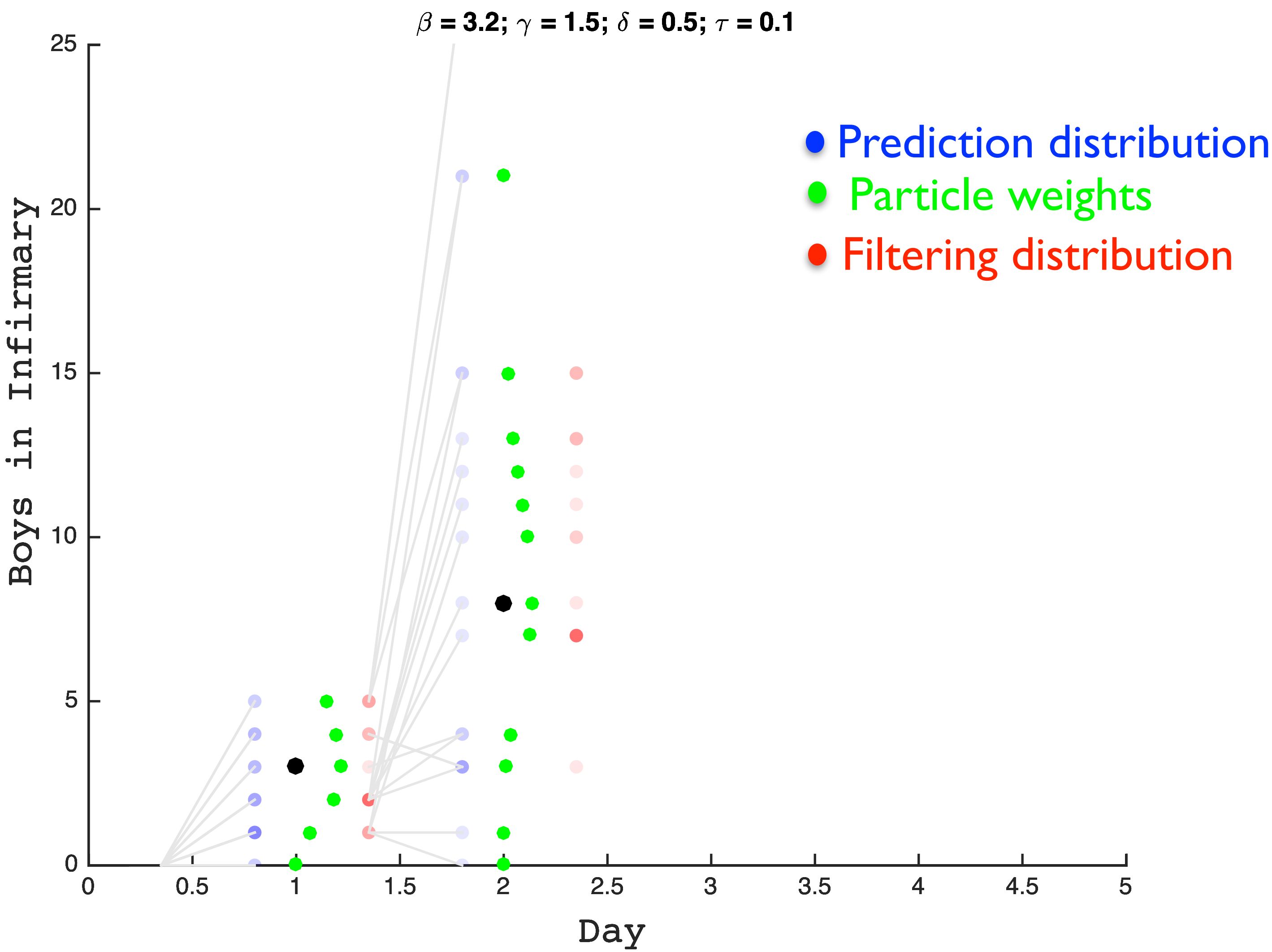
```
X0 = [S0 I0 C0 R0]';              % initial conditions
Np = 5000;                        % number of particles
particles = repmat(X0,1,Np);      % initialize particles
```

```
>> particles = repmat(X0,1,10)

   763   763   763   763   763   763   763   763   763   763    <---- Susceptibles
     1     1     1     1     1     1     1     1     1     1    <---- Infectious
     0     0     0     0     0     0     0     0     0     0    <---- Convalescent
     0     0     0     0     0     0     0     0     0     0    <---- Recovered
```

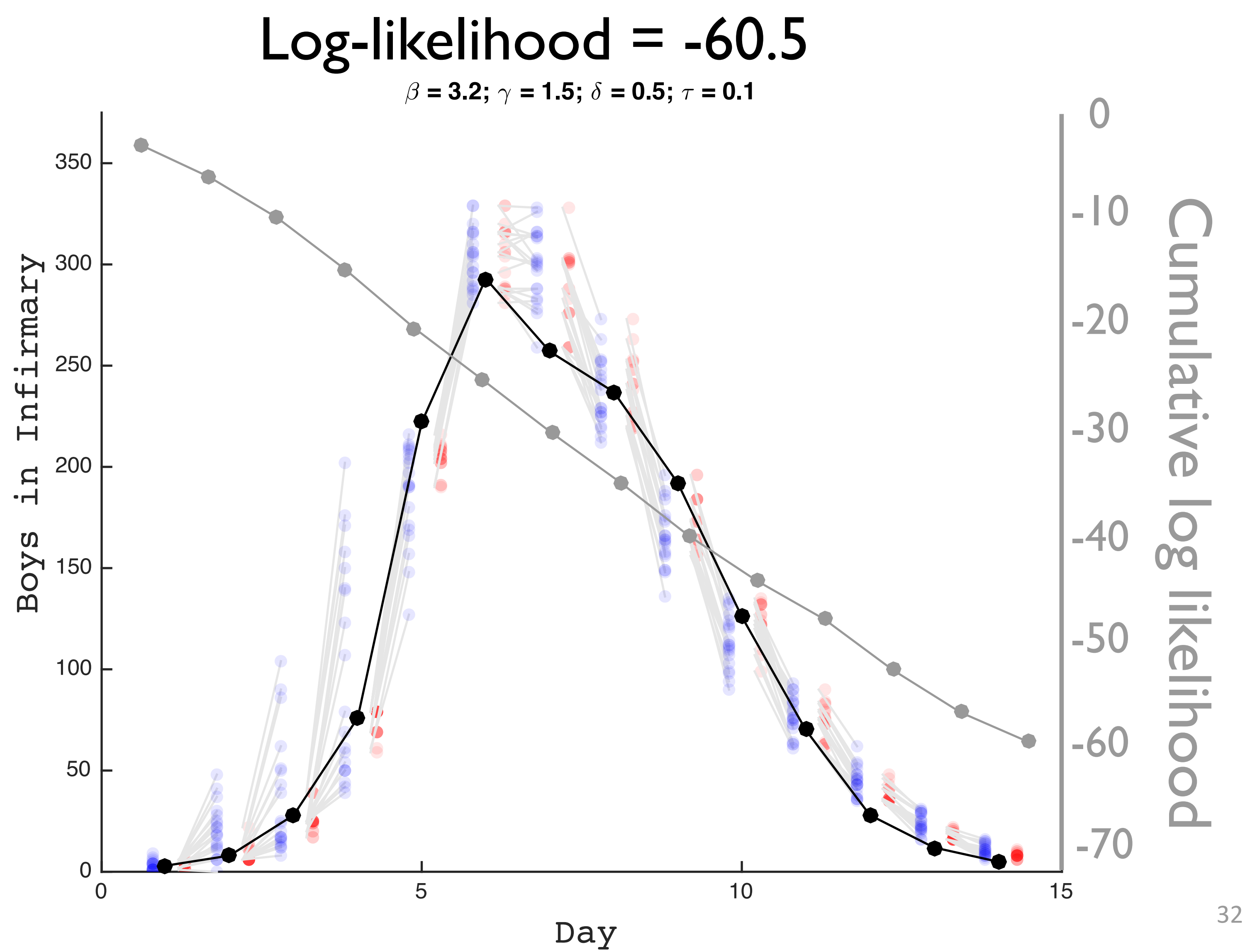
Particle filter: Matlab code

```
X0 = [S0 I0 C0 R0]';                % initial conditions
Np = 5000;                          % number of particles
particles = repmat(X0,1,Np);        % initialize particles
Y = data;                           % set data vector

for k = 1:N                         % step through data
    % advance particles according to state process model
    particles = SICR_simulateState(t(k),t(k+1),particles,beta/popsize,gamma,delta);

    % compute weights according to measurement model
    weights = nbinpdf(Y(k),nbr,1./(1+particles(3,:)/nbr));

    % compute conditional log likelihood
    ell(k) = log(mean(weights));

    % resample particles to update filtering distribution
    particles = particles(:,randsample(Np,Np,true,weights));
end
logL = sum(ell);
```

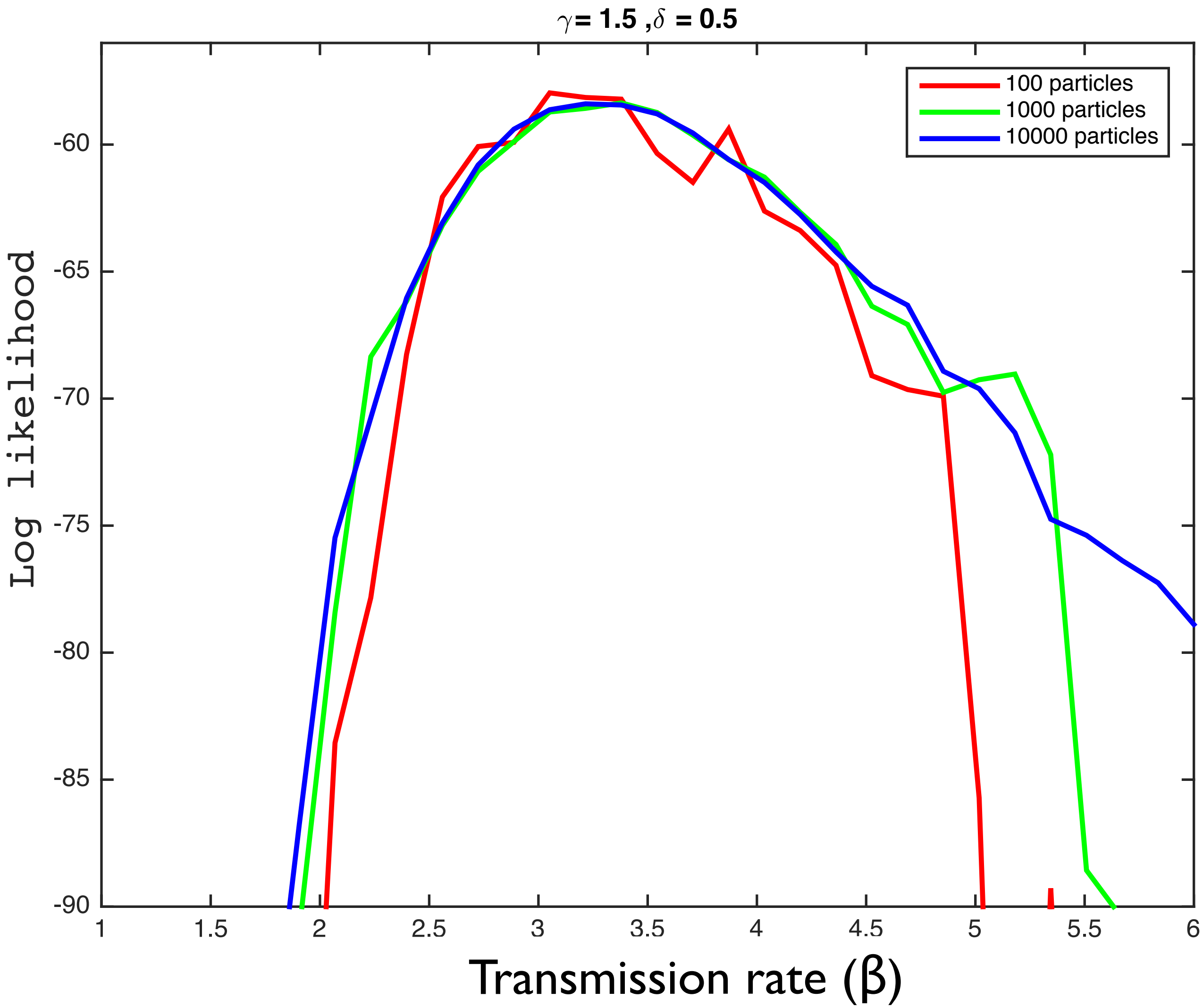
The particle filter in action



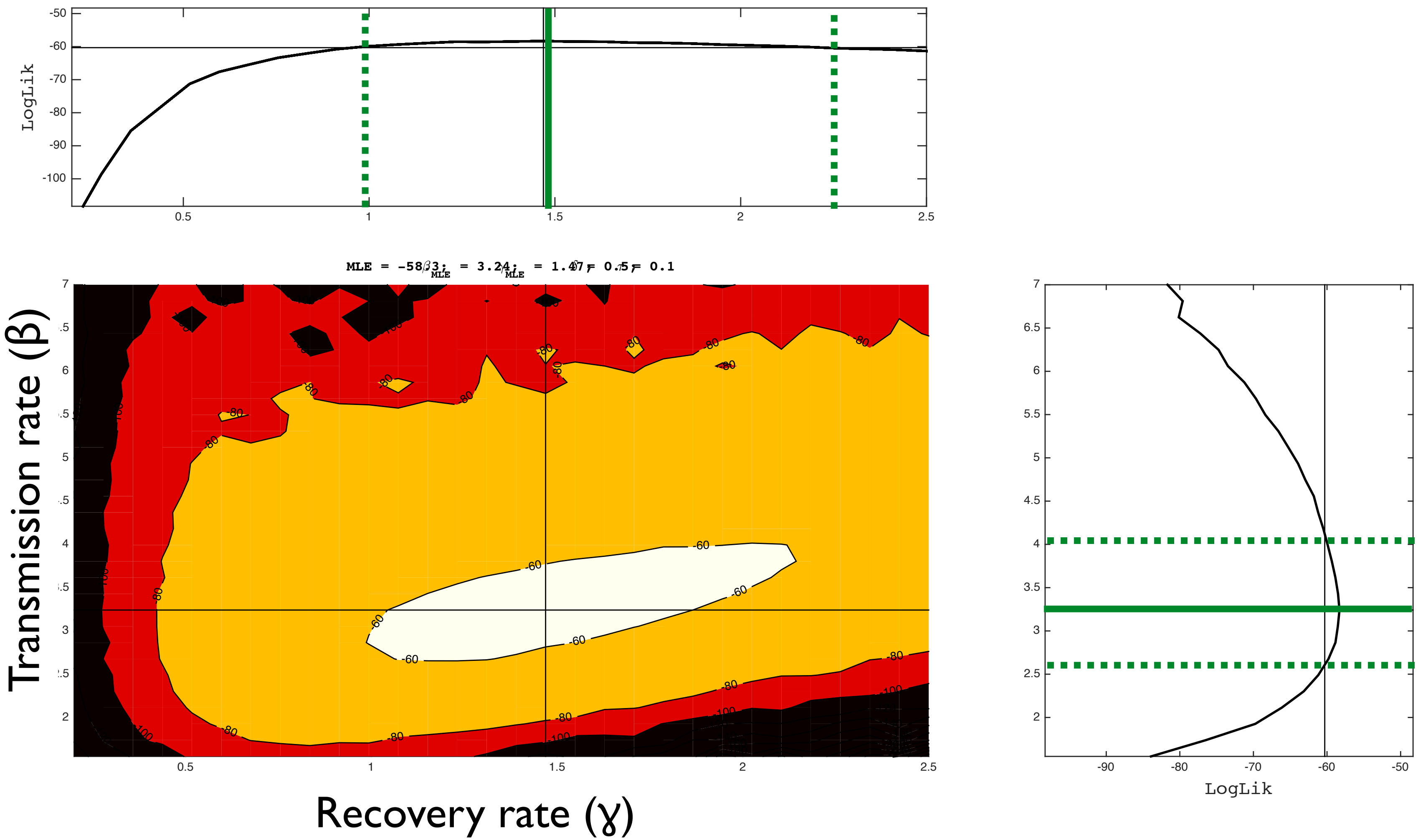
The particle filter in action



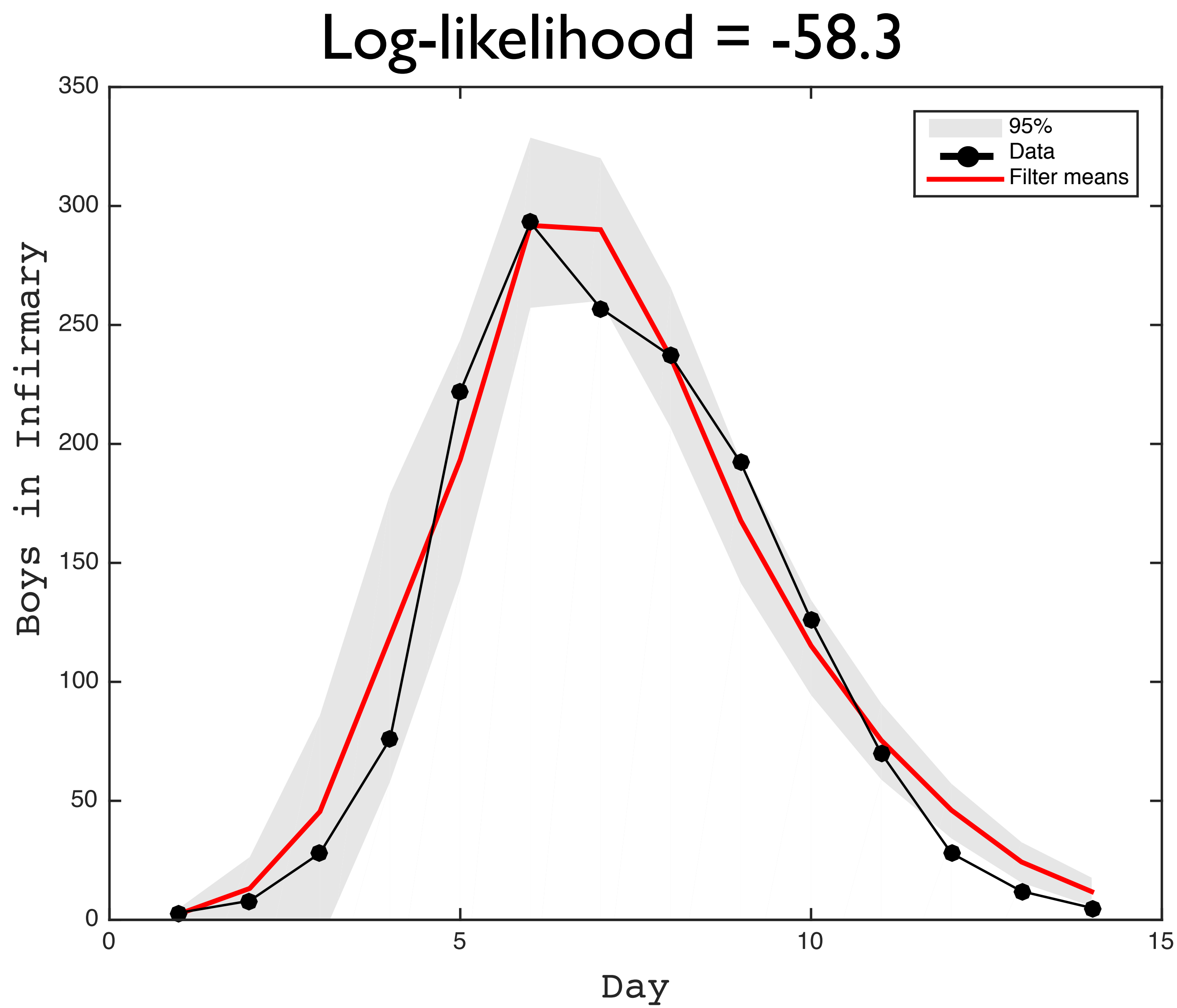
Effect of particle number



Likelihood surface



Prediction means



The particle filter

Model comparison.

	detSIR	detSICR	stochSICR
R_0	3.71		
IP (da)	2.00		
CP (da)	—		
logLik	-72.80		

Further, ...

1. Include uncertainty in initial conditions
   - We took I(0) = 1. Instead could estimate I(0) together with β and γ (now have 1 fewer data points)
2. Explicit observation model
   - Assumed here measurement errors negative binomially distributed
3. What is appropriate model?
   - SEIR model? (latent period before becoming infectious)
   - SEICR model? ("confinement to bed")
   - Time varying parameters? (e.g. action taken to control spread)

Further, ...

4. Optimization
   - The 'particle filter' allow us to calculate likelihood - how we optimize it?
5. Can we simultaneously estimate numerous parameters?
   - More complex models have more parameters... estimate all from 14 data points? ⇒ identifiability
6. More complex models are more flexible, so tend to fit better
   - How do we determine if increased fit justifies increased complexity? ⇒ information criteria