An Overview of Sequential Monte Carlo Methods

By O. Cappe, S. Godsill, & E. Moulines
Presented by Bevan Jones
April 2010
Brown University

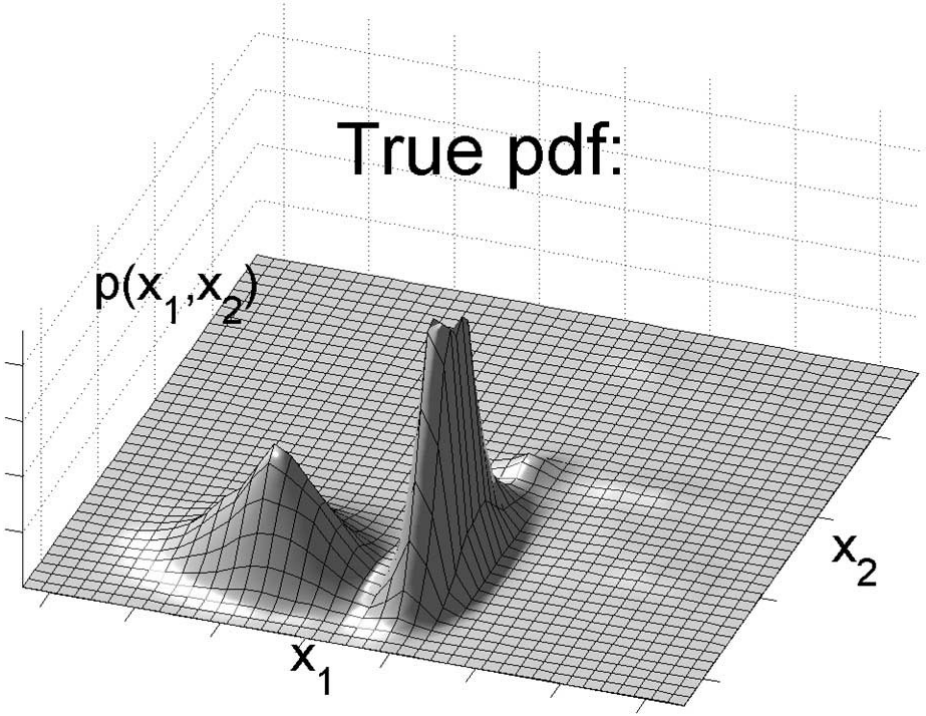
Motivation

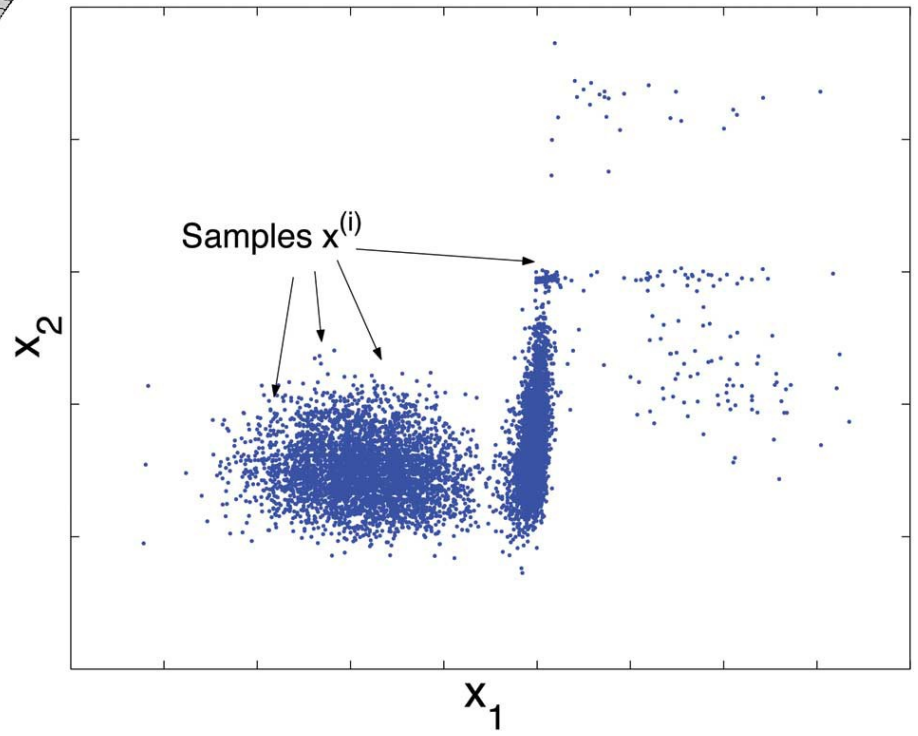
- Can use sampling to estimate where no analytical solution is available
- Can sample in an online fashion
- There are convergence guarantees, given certain conditions such as large enough number of samples

Outline

- Importance Sampling
- Sequential Importance Sampling
- Particle Filters
- Choosing a good proposal
  - Bootstrap filter
  - Optimal kernel
- Auxiliary Sampling
- Smoothing
- Parameter Estimation

Approximation by Sampling





Expectations by Monte Carlo

- The exact expectation of ***h(x)***

$$\bar{h} \stackrel{\text{def}}{=} \int h(x)p(x)dx$$

- Approximate by sampling ***N*** points $\boldsymbol{x^{(1)},\ldots,x^{(N)}}$

$$\bar{h} \approx \frac{1}{N}\sum_{i=1}^{N} h\left(x^{(i)}\right)$$

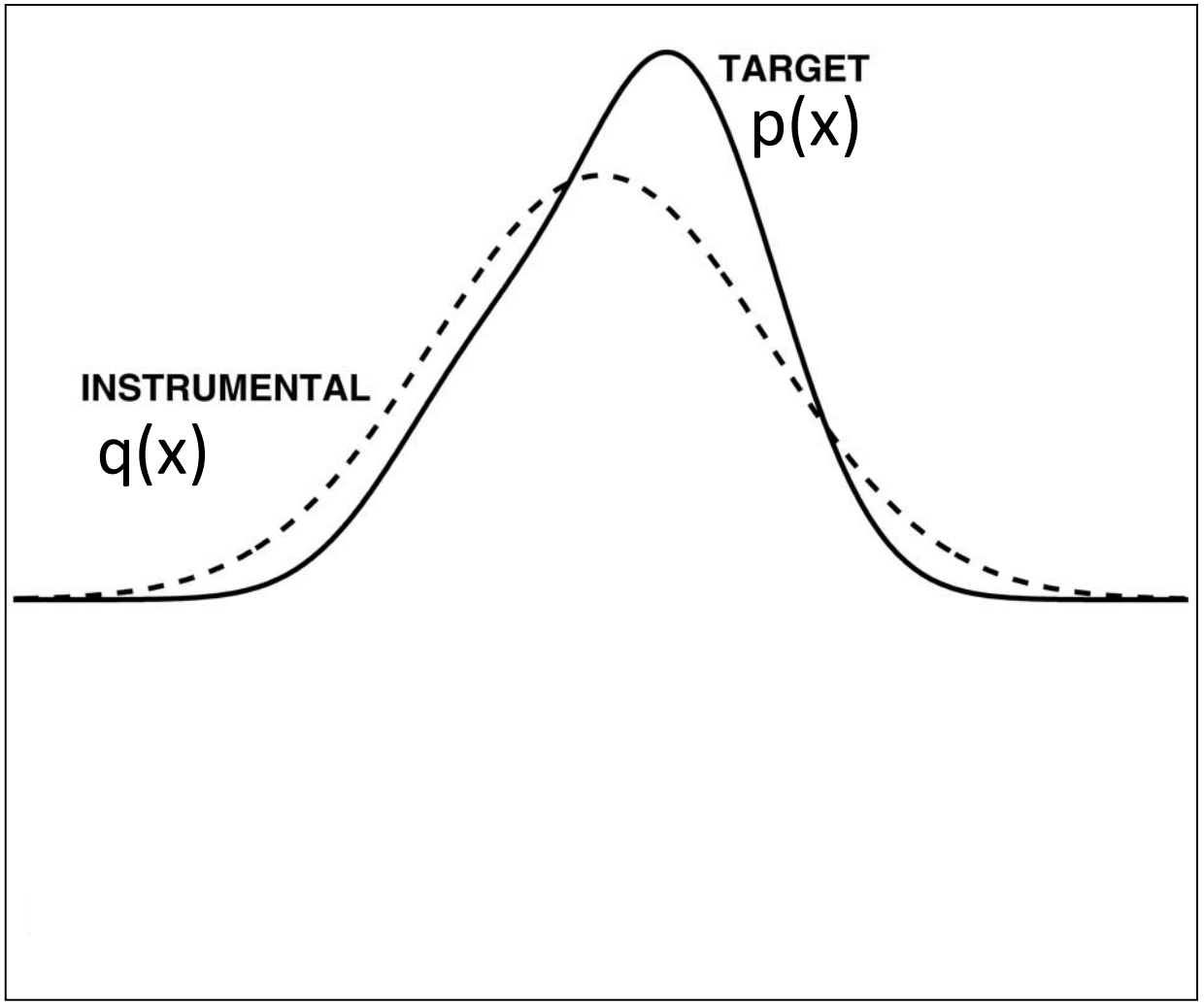
- But what if we don't know how to sample from ***p(x)***?

Importance Sampling

- If we can't sample from ***p(x)*** directly, sample from another proposal distribution ***q(x)*** instead
- Then correct by weighting each sampled point $\boldsymbol{x^{(i)}}$ according to $$\tilde{\omega}^{(i)} = \frac{p(x^{(i)})}{q(x^{(i)})}$$
- Then resample with replacement according to the normalized weights $$\omega^{(i)} = \frac{\tilde{\omega}^{(i)}}{\sum_{i=1}^{N} \tilde{\omega}^{(i)}}$$
- Multiply underrepresented points, discard overrepresented points

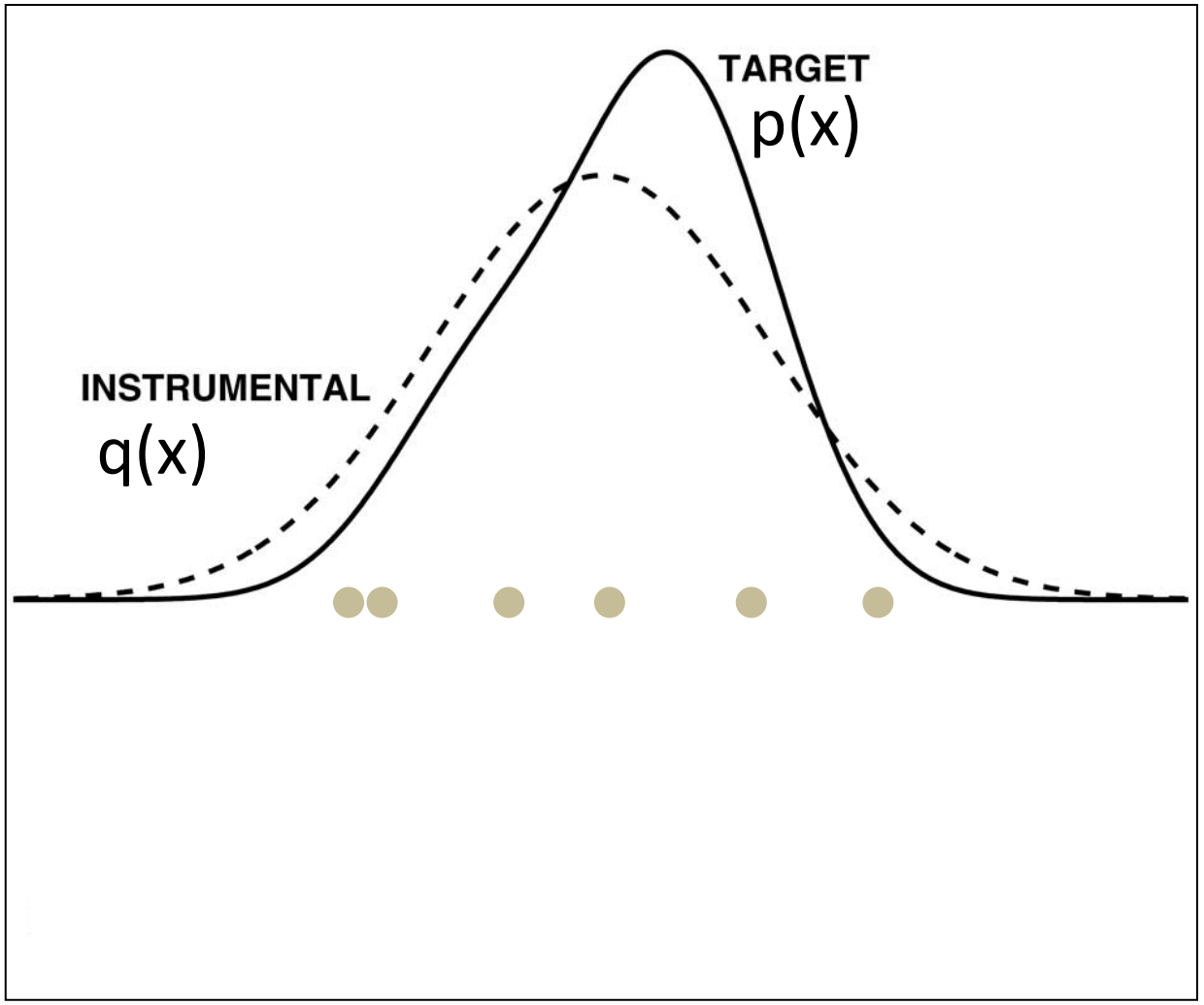
Resampling

- sample from q(x)



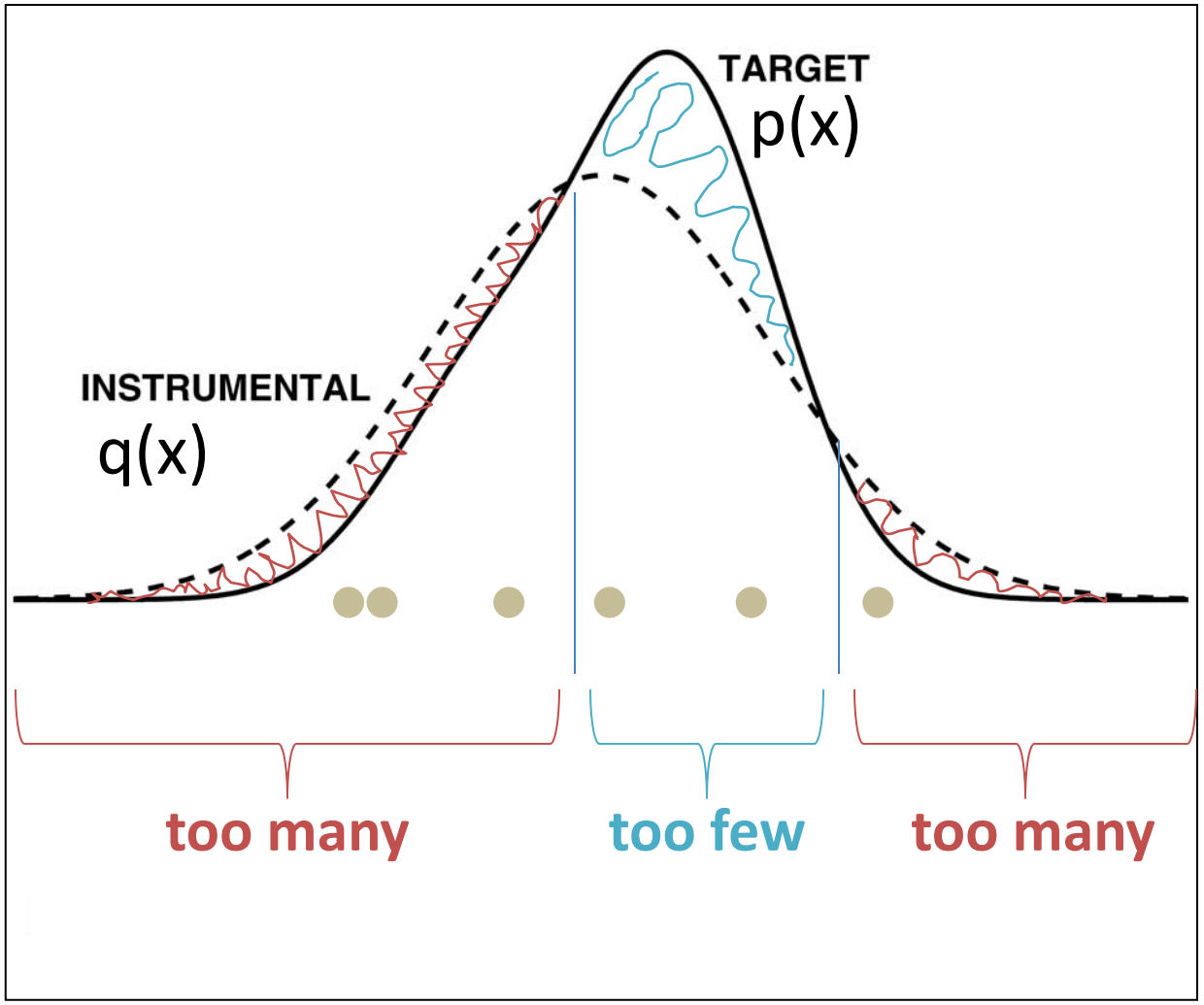
Resampling

- sample from q(x)



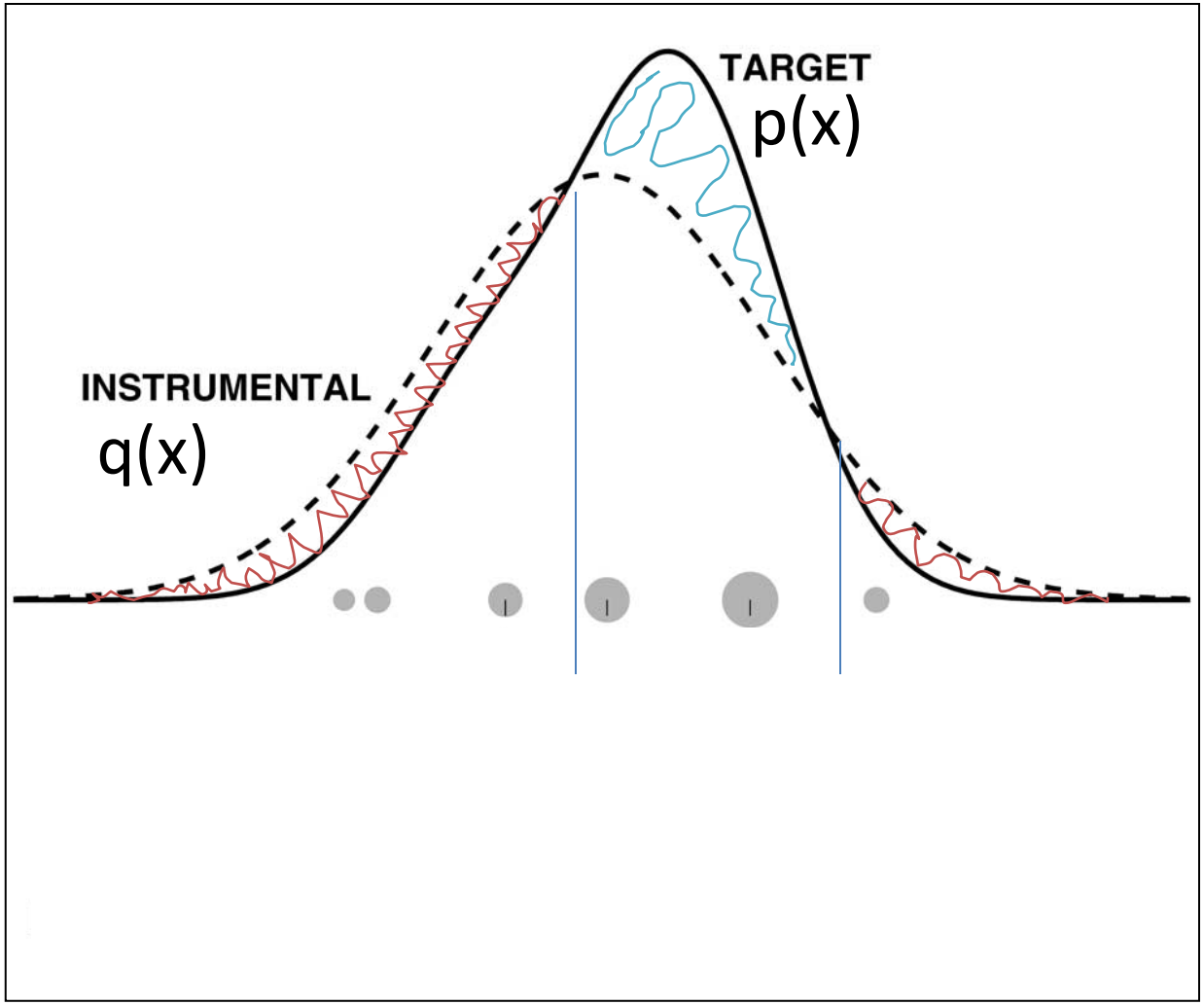
Resampling

- sample from q(x)



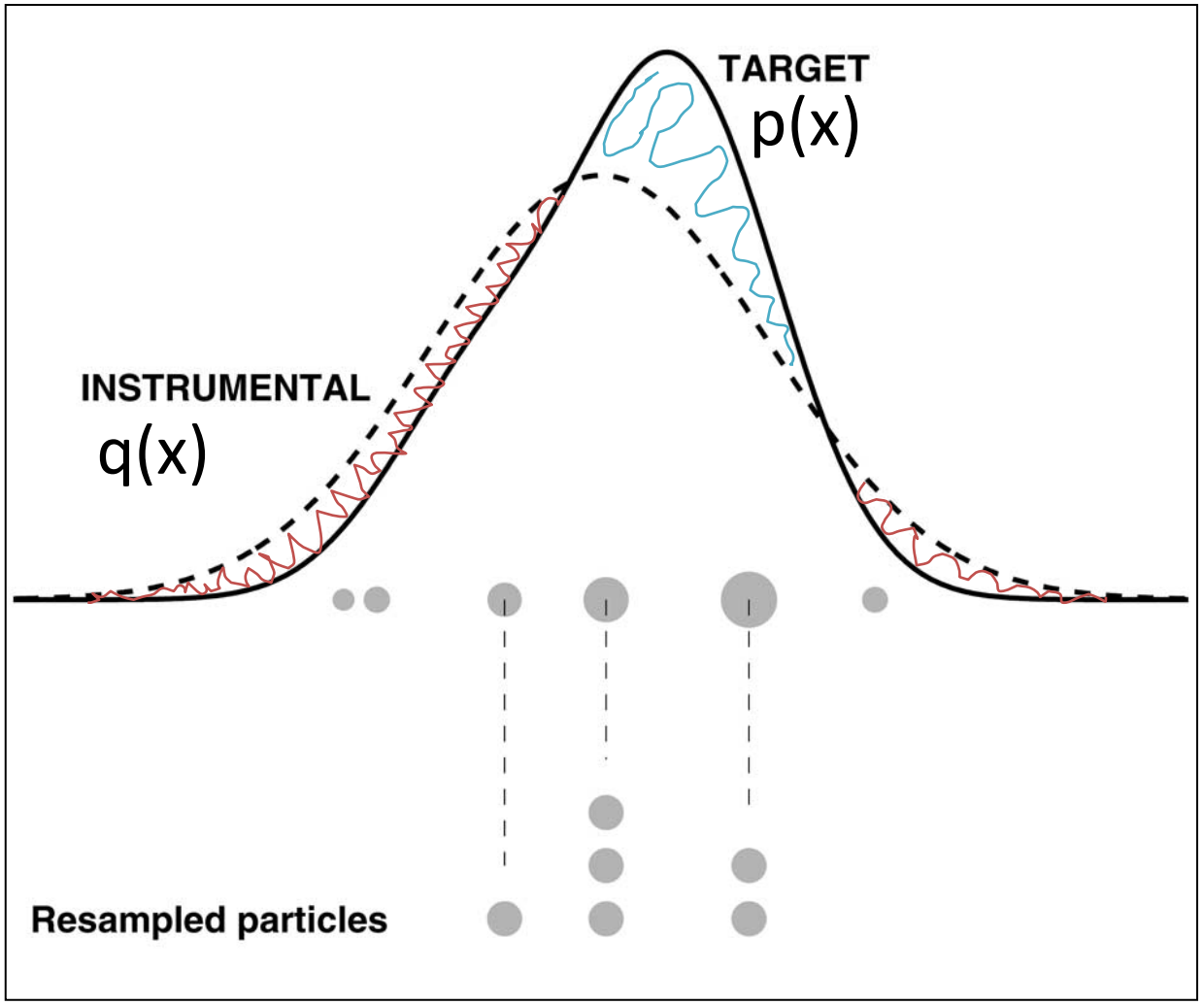
Resampling

- sample from q(x)
- Compute weights



Resampling

- sample from q(x)
- Compute weights
- Resample according to weights



Importance Sampling

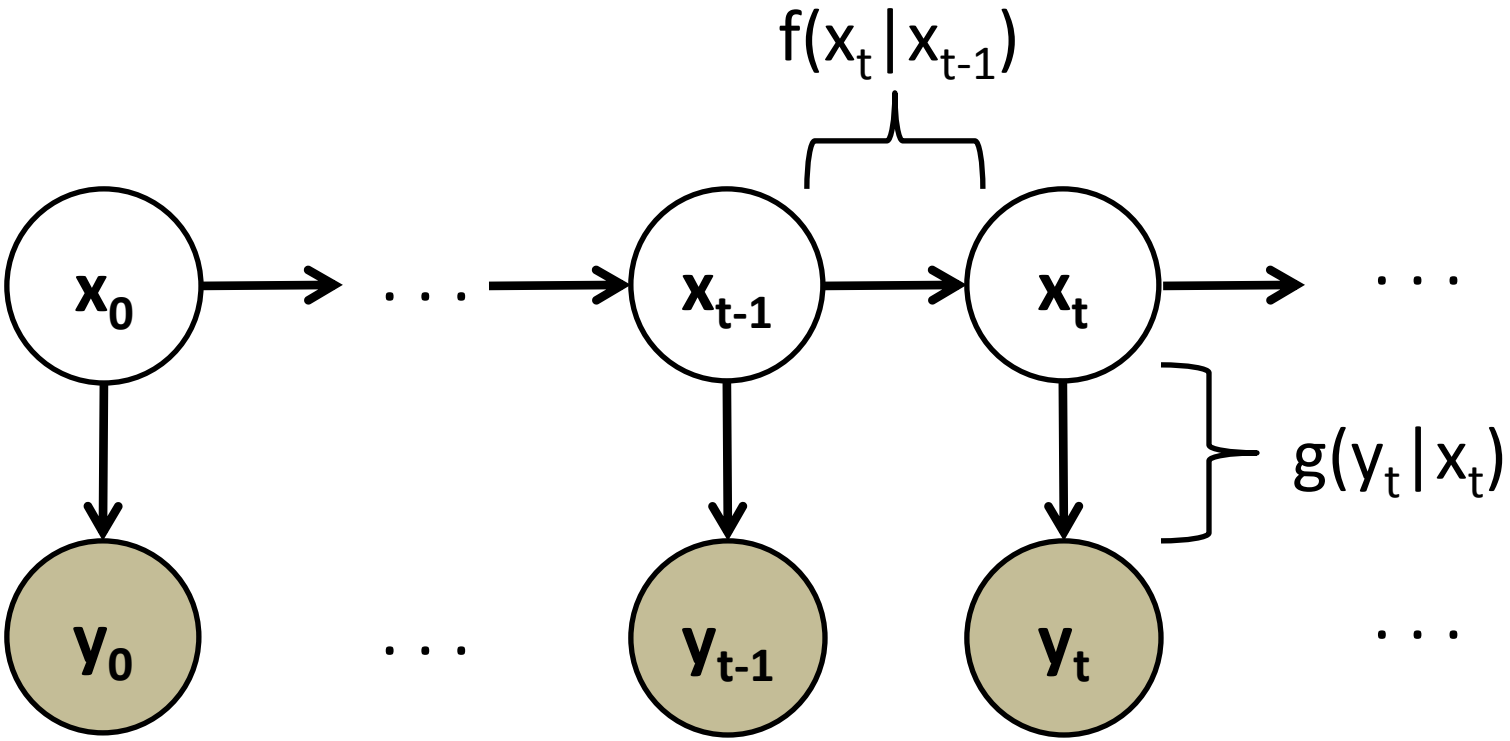
- Also works for multivariate distributions $p(x_{0:t})$ and $q(x_{0:t})$.
- But requires a method of sampling from $q(x_{0:t})$.
- Nice if $q(x_{0:t})$ factorizes recursively

$$q(x_{0:t}) = q(x_{0:t-1}) \cdot q(x_t \mid x_{0:t-1})$$

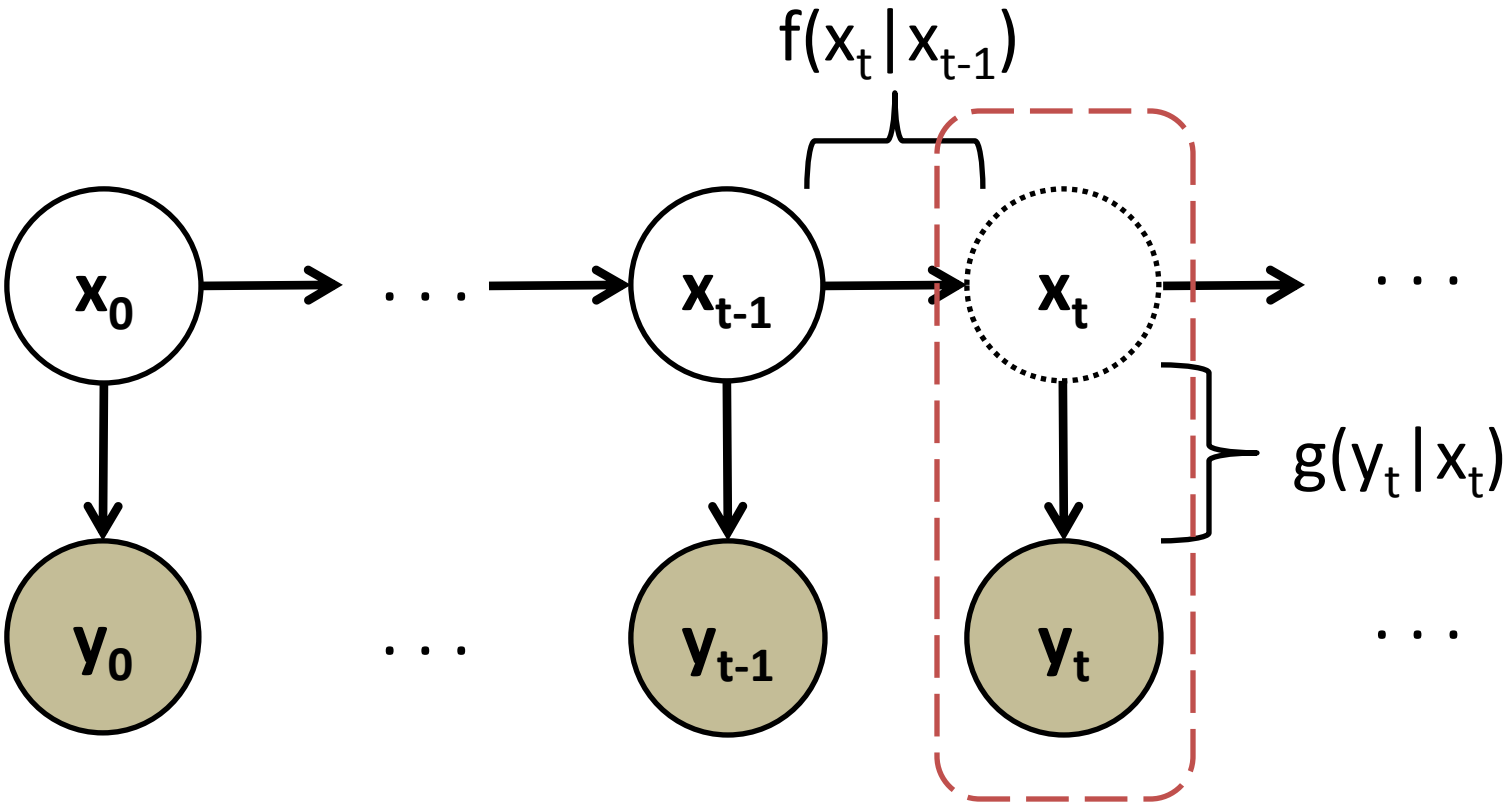
- Then we can do this sequentially

$$x_t \sim q(x_t \mid x_{0:t-1})$$

Nonlinear Time Series Model



Nonlinear Time Series Model



Sequential Importance Sampling

- Incremental updating

$$\pi_{0:t|0:t}(x_{0:t} \mid y_{0:t-1}) \propto \pi_{0:t-1}(x_{0:t-1} \mid y_{0:t-1}) \cdot f(x_t \mid x_{t-1}) g(y_t \mid x_t)$$

$$q_{0:t}(x_{0:t} \mid y_{0:t}) = \underbrace{q_{0:t-1}(x_{0:t-1} \mid y_{0:t-1})}_{\text{current path}} \cdot \underbrace{q_t(x_t \mid x_{t-1}, y_t)}_{\text{extension}}$$

$$\tilde{\omega}_t^{(i)} = \frac{\pi_{0:t|0:t}(x_{0:t}^{(i)} \mid y_{0:t-1})}{q_{0:t}(x_{0:t}^{(i)} \mid y_{0:t})}$$

Sequential Importance Sampling

- Incremental updating

$$\pi_{0:t|0:t}(x_{0:t} \mid y_{0:t-1}) \propto \pi_{0:t-1}(x_{0:t-1} \mid y_{0:t-1}) \cdot f(x_t \mid x_{t-1}) g(y_t \mid x_t)$$

$$q_{0:t}(x_{0:t} \mid y_{0:t}) = q_{0:t-1}(x_{0:t-1} \mid y_{0:t-1}) \cdot q_t(x_t \mid x_{t-1}, y_t)$$

$$\tilde{\omega}_t^{(i)} \propto \omega_{t-1}^{(i)} \cdot \frac{f(\tilde{x}_t^{(i)} \mid \tilde{x}_{t-1}) g(y_t \mid \tilde{x}_t^{(i)})}{q_t(\tilde{x}_t^{(i)} \mid \tilde{x}_{t-1}^{(i)}, y_t)}$$

current path

extension

$$\tilde{\omega}_t^{(i)} = \frac{\pi_{0:t|0:t}(x_{0:t}^{(i)} \mid y_{0:t-1})}{q_{0:t}(x_{0:t}^{(i)} \mid y_{0:t})}$$

Sequential Importance Sampling

- Incremental updating

$$\pi_{0:t|0:t}(x_{0:t} \mid y_{0:t-1}) \propto \pi_{0:t-1}(x_{0:t-1} \mid y_{0:t-1}) \cdot f(x_t \mid x_{t-1}) g(y_t \mid x_t)$$

$$q_{0:t}(x_{0:t} \mid y_{0:t}) = q_{0:t-1}(x_{0:t-1} \mid y_{0:t-1}) \cdot q_t(x_t \mid x_{t-1}, y_t)$$

$$\tilde{\omega}_t^{(i)} \propto \omega_{t-1}^{(i)} \cdot \frac{f(\tilde{x}_t^{(i)} \mid \tilde{x}_{t-1}) g(y_t \mid \tilde{x}_t^{(i)})}{q_t(\tilde{x}_t^{(i)} \mid \tilde{x}_{t-1}^{(i)}, y_t)}$$

current path

extension

- Everything can be defined recursively

Sequential Importance Sampling

initialization
for t = 1, …, T do

    for i = 1, …, N do
        draw sample extension $x_t \sim q(x_t | x_{t-1})$
        update weights
    end
    normalize weights

    compute estimate $\bar{h}_t = \sum_{i=1}^{N} \omega_t^{(i)} h_t\left(\tilde{x}_t^{(i)}\right)$

end

Particle Filter

initialization

for t = 1, …, T do

resample

for i = 1, …, N do

draw sample extension $x_t \sim q(x_t | x_{t-1})$

update weights

end

normalize weights

compute estimate $\bar{h}_t = \sum_{i=1}^{N} \omega_t^{(i)} h_t\left(\tilde{x}_t^{(i)}\right)$

end

Particle Filter

- Particles wander into improbable space
  - Large high-dimensional space getting larger each iteration
- Need to resample
  - High probability particles multiply (cloned)
  - Low probability particles die out
- No longer Importance Sampling
  - Particles are no longer independent since resampling results in shared histories

Choosing ***q***: Bootstrap Filter

- What to choose for ***q***?
- Math is easy if we let ***q*** be a simplification of π (ignore observations)

$$q(x_t \mid x_{t-1}) = f(x_t \mid x_{t-1})$$

- Sample from prior, resample using likelihood

$$\tilde{\omega}_t^{(i)} = \omega_{t-1}^{(i)} \cdot \frac{f(\tilde{x}_t^{(i)} \mid \tilde{x}_{t-1}) g(y_t \mid \tilde{x}_t^{(i)})}{f(\tilde{x}_t^{(i)} \mid \tilde{x}_{t-1}^{(i)})}$$

$$= \omega_{t-1}^{(i)} \cdot g(y_t \mid \tilde{x}_t^{(i)})$$

- Easy to implement, but proposal distribution can be very different from true posterior
  - poor performance

Optimal *q*

- Instead of ignoring observations, exploit all information (previous state & observation)

$$q(x_t \mid x_{t-1}) = \frac{f(x_t \mid x_{t-1}) g(y_t \mid x_t)}{\int f(x \mid x_{t-1}) g(y_t \mid x) dx} = p(x_t \mid y_t, x_{t-1})$$

- Weights also get a nice form

$$\omega_t^{(i)} \propto \omega_{t-1}^{(i)} \cdot \frac{p(x_t, y_t \mid x_{t-1})}{p(x_t \mid y_t, x_{t-1})} = \omega_{t-1}^{(i)} \cdot p(y_t \mid x_{t-1})$$

- Much closer match to the posterior, resulting in less conditional variance given current path
- But the integral above can be hard to compute

Better Resampling: Auxiliary Sampling

- Resample particles that will do better in future
- One way: sample paths from conditional marginal $\pi_{0:t-1|0:t}$

- Define discrete distribution on current paths, with $v^{(i)}$ as weights

$$q_t(dx_{0:t}) = q_{0:t-1}(dx_{0:t-1}|y_{0:t})q_t(dx_t|x_{t-1}, y_t)$$
$$= \left(\sum_{i=1}^{N} v_{t-1}^{(i)} \delta_{x_{0:t-1}^{(i)}}(dx_{0:t-1})\right)\left(q_t\left(dx_t|x_{t-1}^{(i)}, y_t\right)\right)$$

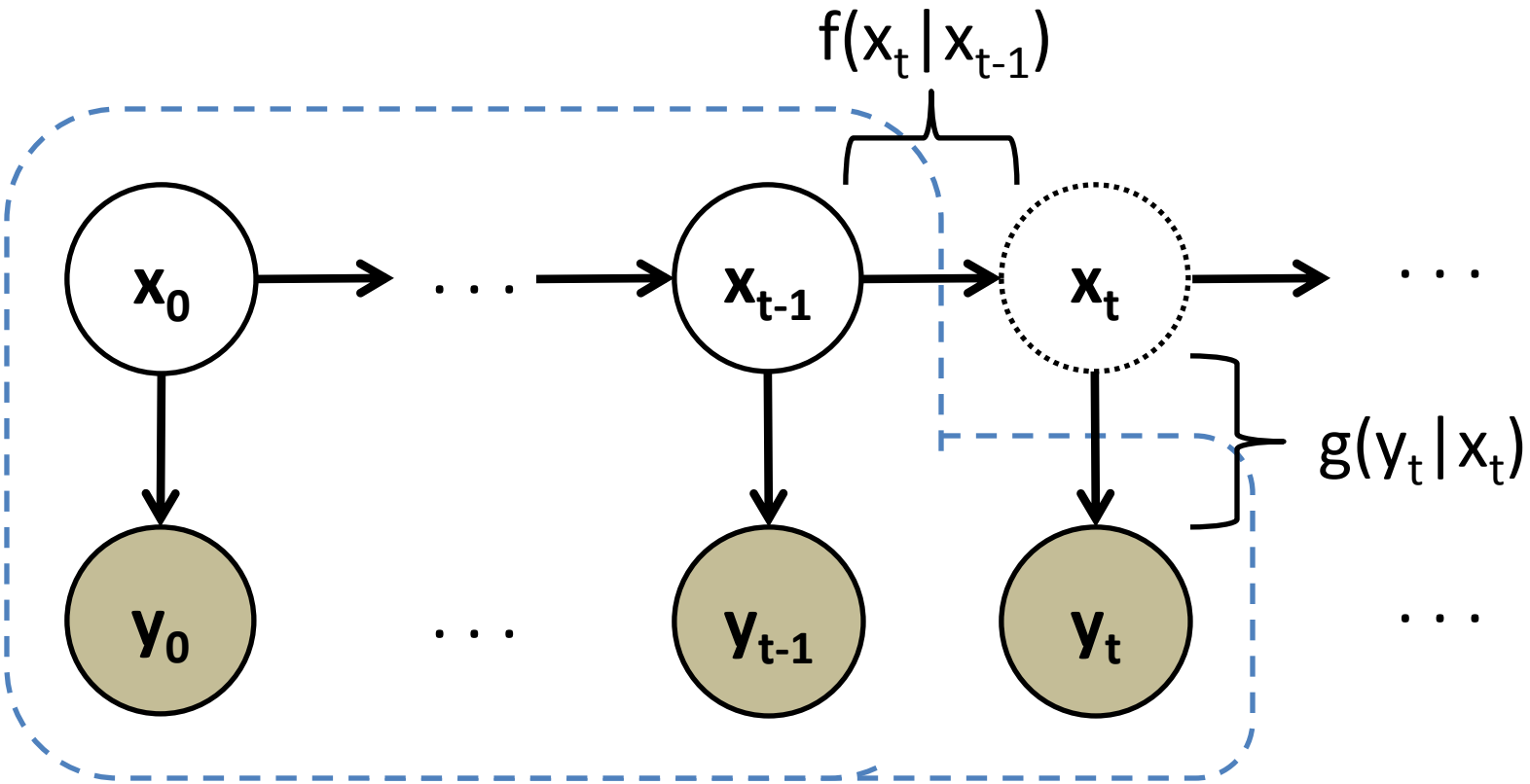
- Suggested $v^{(i)}$:

$$v_{t-1}^{(i)} = g(y_t \mid \mu_t^{(i)})\omega_{t-1}^{(i)} \qquad \text{where} \quad \mu_t^{(i)} = \mathrm{E}[f(x_t \mid x_{t-1}^{(i)})]$$

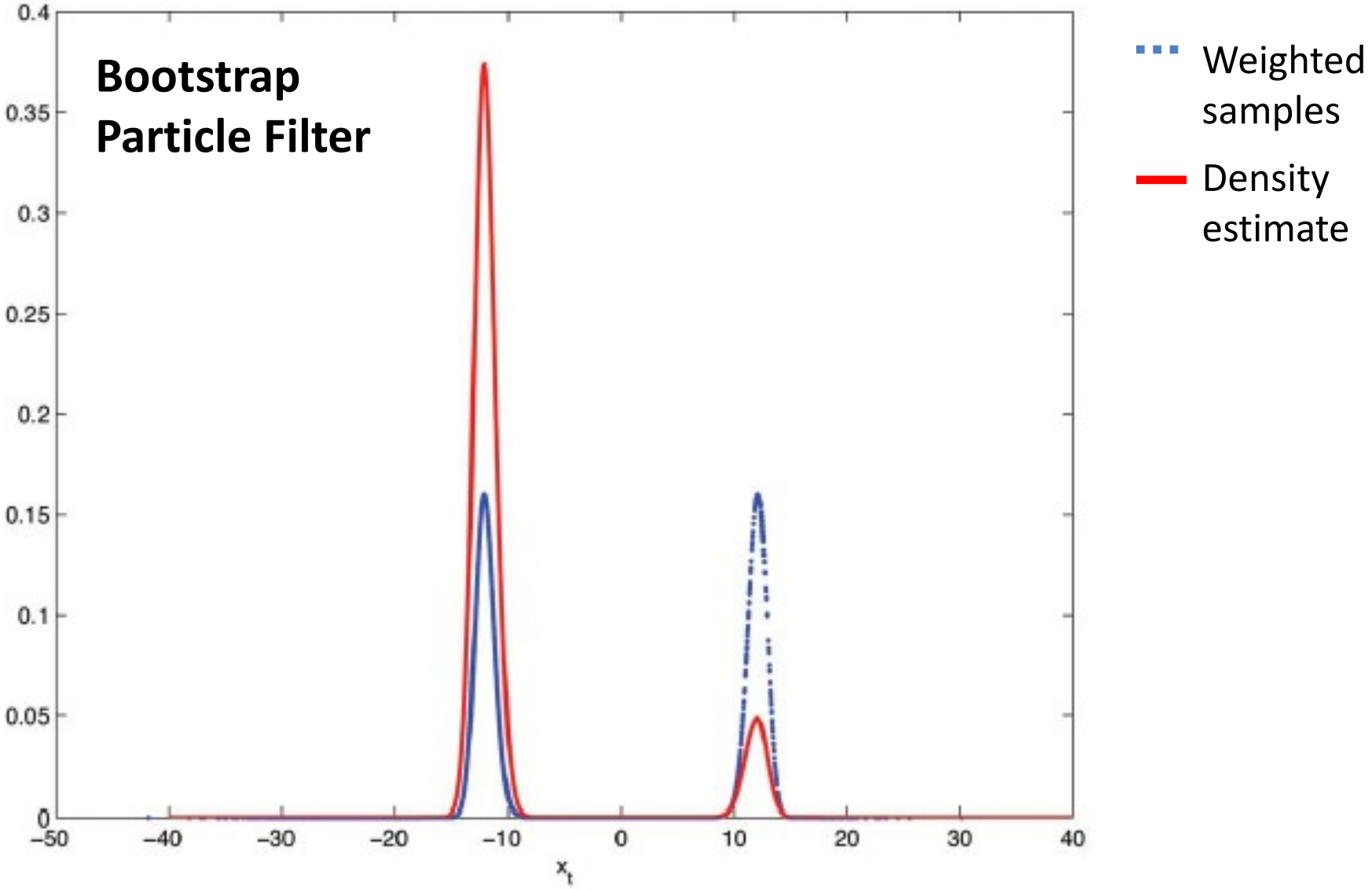
- New weight update rule:

$$\tilde{\omega}_t^{(i)} = \frac{\omega_{t-1}^{(i)}}{v_{t-1}^{(i)}} \cdot \frac{g(y_t \mid x_t^{(i)}) f(x_t^{(i)} \mid x_{t-1}^{(i)})}{q_t(x_t^{(i)} \mid x_{t-1}^{(i)}, y_t)}$$

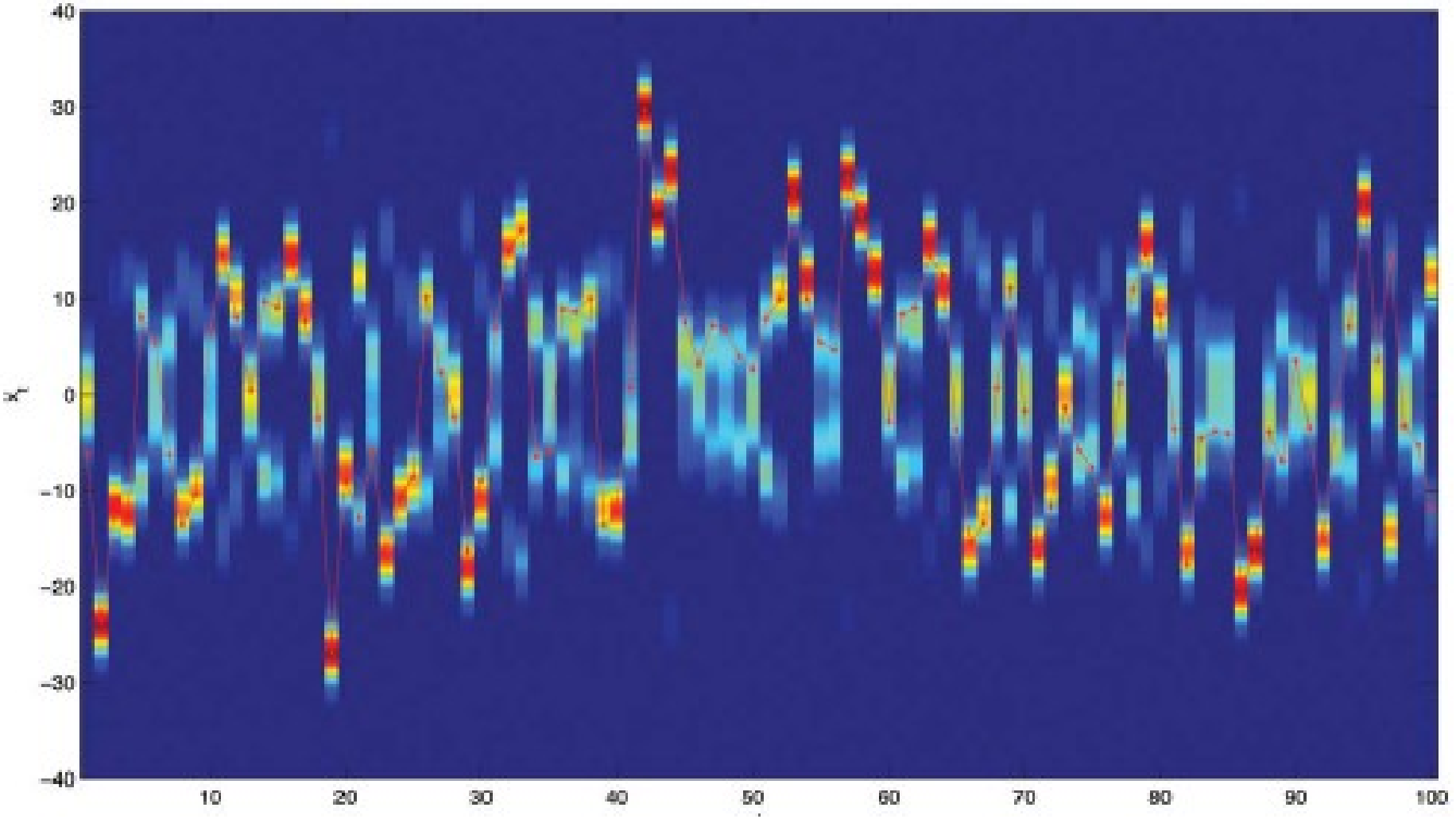
Nonlinear Time Series Model



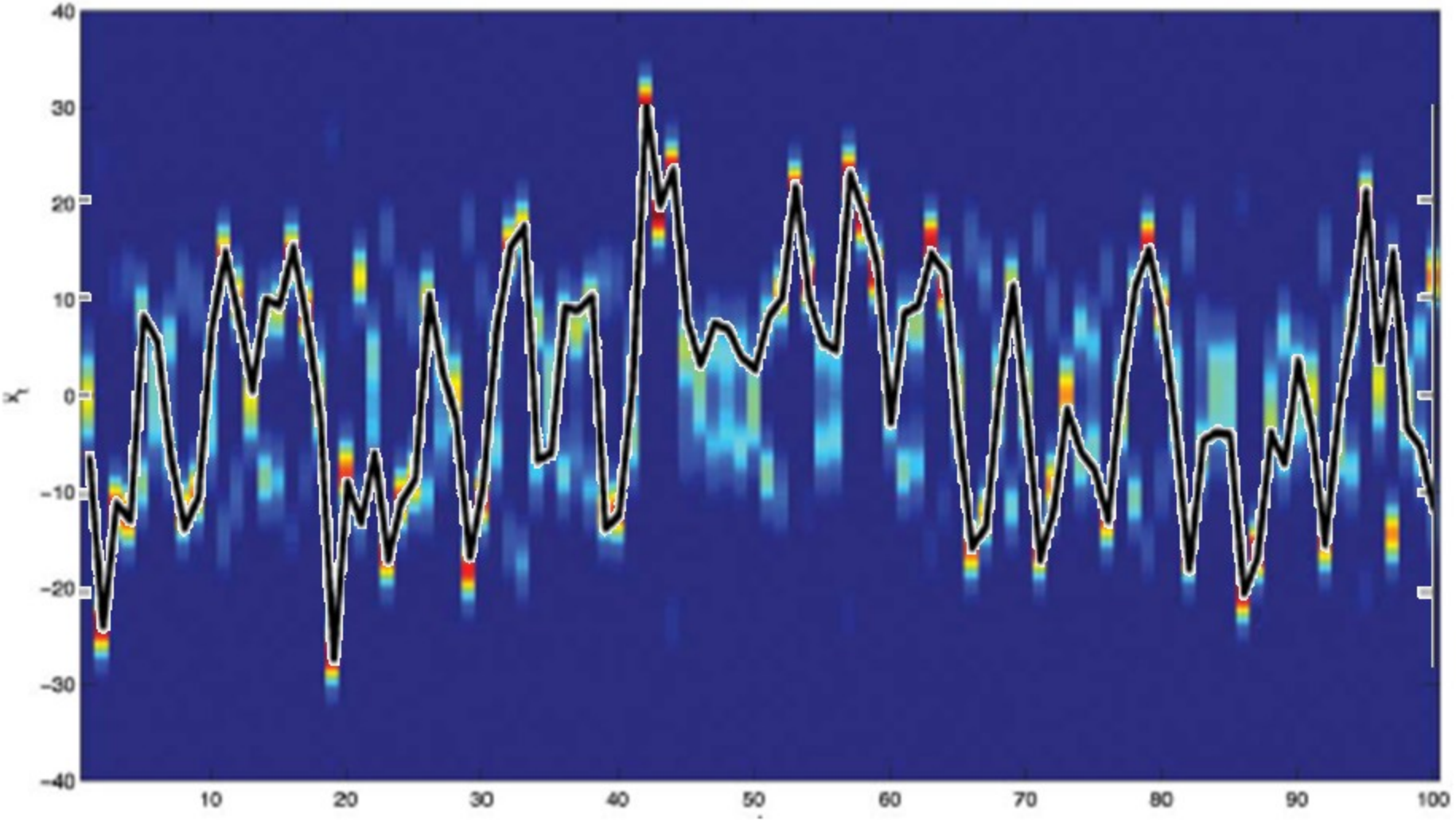
Particle Filter Simulation



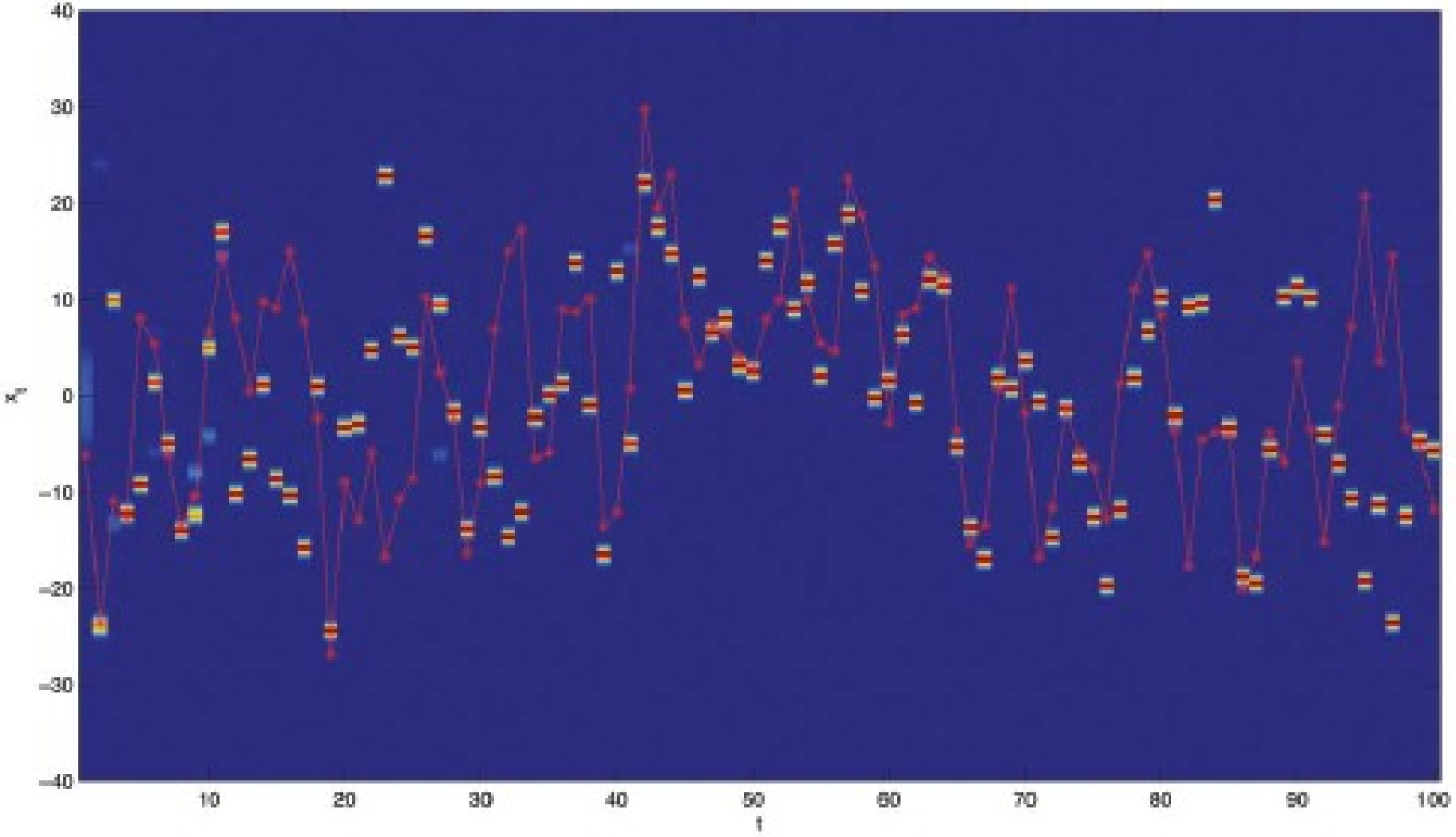
Particle Filter Simulation



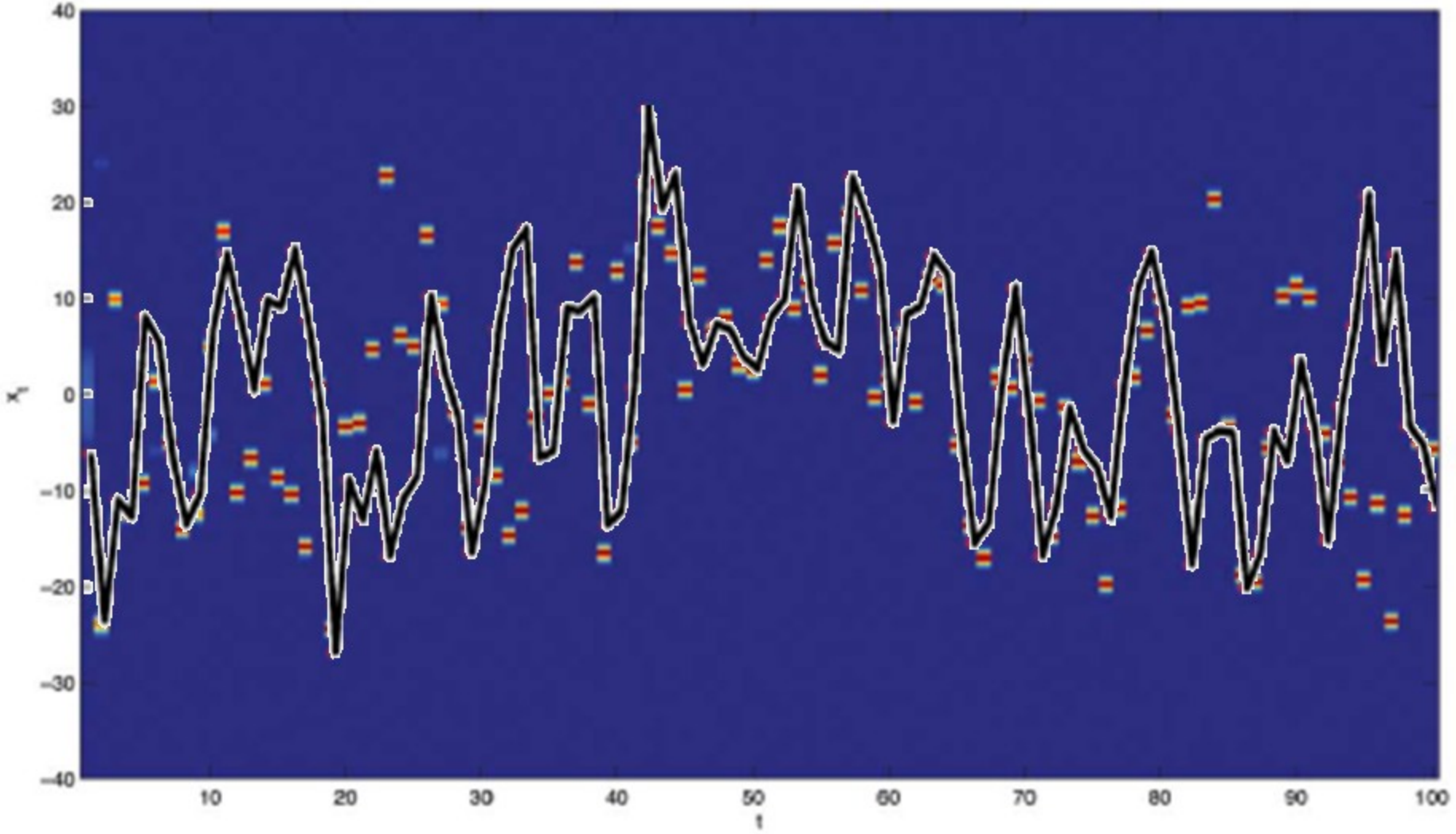
Particle Filter Simulation



Sequential Importance Sampling Simulation



Sequential Importance Sampling Simulation



Filtering & Smoothing

- Filtering

$$\pi_{t|0:t}(x_t \mid y_{0:t})$$ <= particle filter samples this for each ***t***

- Smoothing

$$\pi_{0:t|0:t}(x_{0:t} \mid y_{0:t})$$

Smoothing

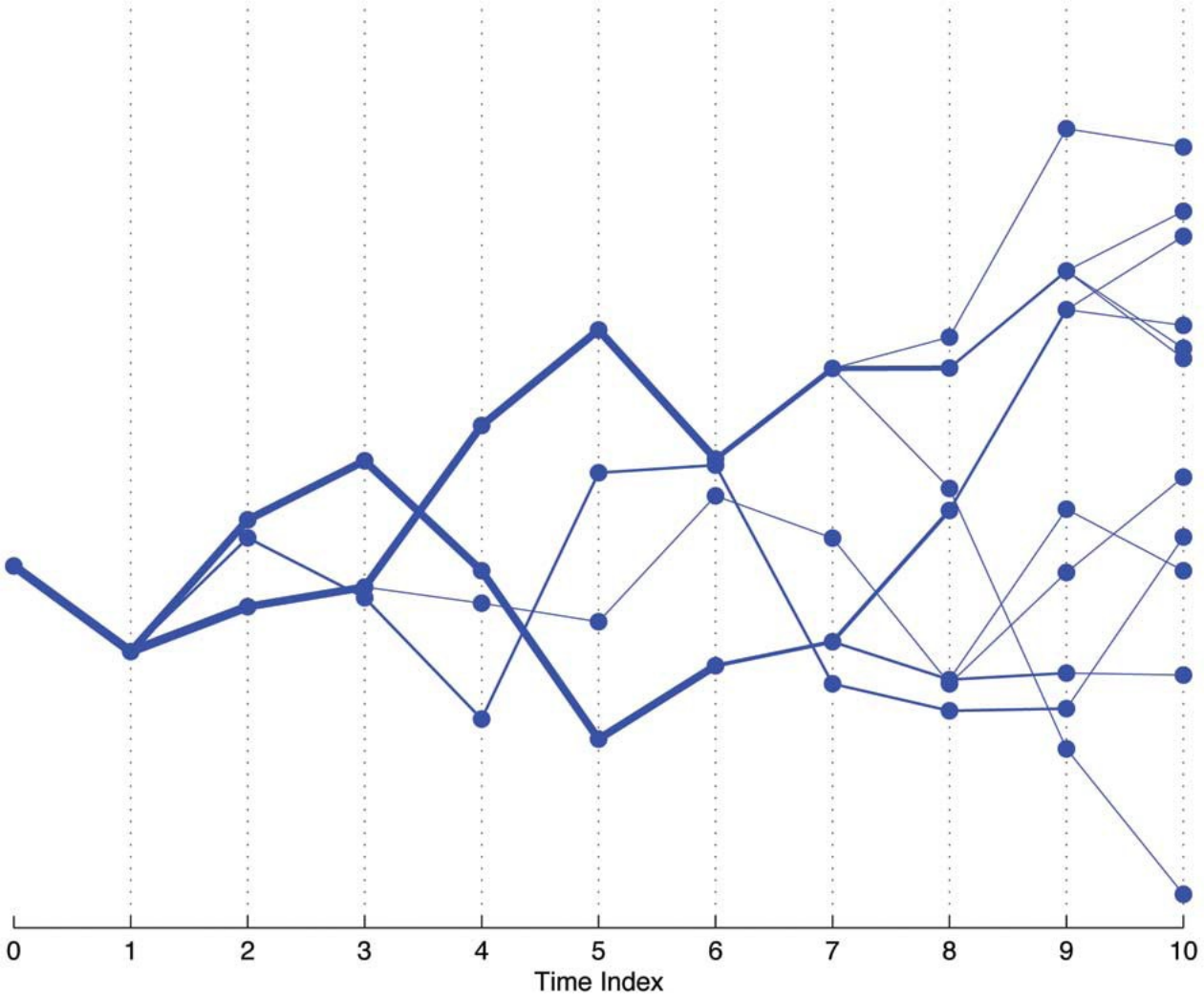
- Use particle filters to sequentially sample each state
- This sequence makes up a path
- Can treat this path as a sample from the joint smoothing distribution

$$\pi_{0:t|0:t}(x_{0:t} \mid y_{0:t})$$

- But because of resampling a lot of particles share paths

Particle Filter Trajectories

- Final particles share paths
- Few samples for earlier states



Forward Filtering-Backward Smoothing

- Use particle filters for entire dataset
- Sample backwards to get paths

$$\pi_{0:T|0:T}(x_{0:T} \mid y_{0:T}) = \pi_{T|0:T}(x_T \mid y_{0:T}) \prod_{t=0}^{T-1} p(x_t \mid x_{t+1}, y_{0:t})$$

where

$$p(x_t \mid x_{t+1}, y_{0:t}) \propto \pi_{t|0:t}(x_t \mid y_{0:t}) f(x_{t+1} \mid x_t)$$

- Approximate $p(x_t \mid x_{t+1}, y_{0:t})$ using particle histories

$$p(dx_t \mid x_{t+1}, y_{0:T}) \approx \sum_{i=1}^{N} \rho_t^{(i)}(x_{t+1}) \delta_{x_t^{(i)}}(dx_t)$$

where

$$\rho_t^{(i)}(x_{t+1}) \propto \omega_t^{(i)} f(x_{t+1} \mid x_t^{(i)})$$

Parameter Estimation

- Want to estimate parameter $\boldsymbol{\theta}$

  $\boldsymbol{f(x_{t+1}|x_t,\theta)}$ and $\boldsymbol{g(y_t|x_t,\theta)}$
- Perform grid search, computing likelihood for each $\boldsymbol{\theta}$ value
- Likelihood can also be computed sequentially

$$l_{0:T}(y_{0:T}\mid\theta)=p(y_0\mid\theta)\prod_{t=1}^{T-1}p(y_{t+1}\mid y_{0:t},\theta)$$

$$p(y_{t+1}\mid y_{0:t},\theta)=\int p(y_{t+1},x_{t+1}\mid y_{0:t},\theta)dx_{t+1}$$

$$=\int\int \pi_{t|0:t}(x_t\mid y_{0:t},\theta)f(x_{t+1}\mid x_t,\theta)g(y_{t+1}\mid x_{t+1},\theta)dx_t dx_{t+1}$$

$$\approx\sum_{i=1}^{N}\omega_t^{(i,\theta)}\int f(x_{t+1}\mid x_t^{(i,\theta)},\theta)g(y_{t+1}\mid x_{t+1},\theta)dx_{t+1}$$

Computing Likelihood Sequentially

- If using Bootstrap Filter

$$\tilde{x}_{t+1}^{(i)} \sim \sum_{i=1}^{N} \omega_t^{(i,\theta)} f(x \mid x_t^{(i,\theta)}, \theta) \quad \text{and} \quad \tilde{\omega}_{t+1}^{(i)} = \omega_t^{(i)} g(y_{t+1} \mid \tilde{x}_{t+1}^{(i)})$$

- Then

$$p(y_{t+1} \mid y_{0:t}, \theta) \approx \sum_{i=1}^{N} \omega_t^{(i,\theta)} \int f(x_{t+1} \mid x_t^{(i,\theta)}, \theta) g(y_{t+1} \mid x_{t+1}, \theta) dx_{t+1}$$

$$\approx E[g]$$

$$\approx \sum_{i=1}^{N} \omega_t^{(i,\theta)} \cdot \frac{\tilde{\omega}_{t+1}^{(i,\theta)}}{\omega_t^{(i,\theta)}}$$

$$= \sum_{i=1}^{N} \tilde{\omega}_{t+1}^{(i,\theta)}$$

Summary

- Sampling allows us to estimate solutions where analytic solutions are unavailable
- Particle filters allow us to sample sequentially, in an online fashion
- General purpose, with methods for solving various problems
  - Filtering
  - Smoothing
  - Parameter estimation
- Conceptually intuitive, theory more complicated