Computer Science

CSC 665-1: Advanced Topics in Probabilistic Graphical Models

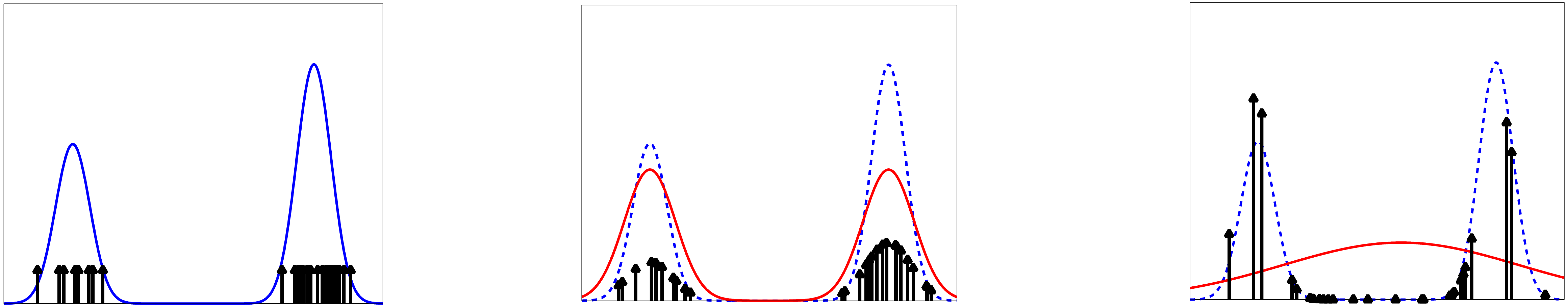
Particle Filtering

Instructor: Prof. Jason Pacheco

(Slides adapted from Prof. Erik Sudderth)

CS242: Lecture 6B Outline

- Importance sampling and likelihood weighting
- Sequential Monte Carlo: Particle Filters



Monte Carlo Estimators

$$\mu \triangleq \mathbb{E}[f] = \int f(x)p(x)\,dx \approx \frac{1}{L}\sum_{\ell=1}^{L} f(x^{(\ell)}) \triangleq \hat{f}_L$$

- Expectation estimated from *empirical distribution* of L samples:

$$\hat{p}_L(x) = \frac{1}{L}\sum_{\ell=1}^{L} \delta_{x^{(\ell)}}(x) \qquad x^{(\ell)} \sim p(x)$$

Practical challenge:
Must draw samples!

- For any *L* this estimator, a random variable, is *unbiased*:

$$\mathbb{E}[\hat{f}_L] = \frac{1}{L}\sum_{\ell=1}^{L} \mathbb{E}[f(x^{(\ell)})] = \mathbb{E}[f]$$

- Guarantees about estimator quality as number of samples *L* grows:

$$\mathrm{Var}[\hat{f}_L] = \frac{1}{L}\mathrm{Var}[f] = \frac{1}{L}\mathbb{E}[(f(x)-\mu)^2] \qquad \Pr\left(\lim_{L\to\infty} \hat{f}_L = \mu\right) = 1$$

Importance Sampling

Target Distribution: *Proposal Distribution:*

$$p(x) = \frac{1}{Z} p^*(x) \qquad q(x) = \frac{1}{Z'} q^*(x) \qquad q(x) > 0 \text{ where } p(x) > 0$$

$$\mathbb{E}[f] = \int f(x) p(x)\, dx = \int f(x) w(x) q(x)\, dx \qquad w(x) = \frac{p(x)}{q(x)}$$

- Estimate target moments via *importance weighted* samples:

$$\hat{f}_L = \sum_{\ell=1}^{L} w_\ell f(x^{(\ell)}) \qquad w_\ell = \frac{w^*(x^{(\ell)})}{\sum_{m=1}^{L} w^*(x^{(m)})} \qquad w^*(x) = \frac{p^*(x)}{q^*(x)}$$

Assumes we can *evaluate un-normalized densities*, and *sample* $x^{(\ell)} \sim q(x)$

- Estimator is *asymptotically unbiased*, and *minimum-variance proposal distribution* is

$$\hat{q}(x) \propto |f(x)| p(x)$$

For evaluation of f(x), this is more efficient than sampling from target p(x)!

Importance Sampling

Target Distribution:

$$p(x) = \frac{1}{Z} p^*(x)$$

Proposal Distribution:

$$q(x) = \frac{1}{Z'} q^*(x) \qquad q(x) > 0 \text{ where } p(x) > 0$$

$$\mathbb{E}[f] = \int f(x)p(x)\,dx = \int f(x)w(x)q(x)\,dx \qquad w(x) = \frac{p(x)}{q(x)}$$

- Optimal proposal can be derived via *Jensen's inequality*:

$$\mathrm{Var}_q[f(x)w(x)] = \mathbb{E}_q[f^2(x)w^2(x)] - \mu^2$$

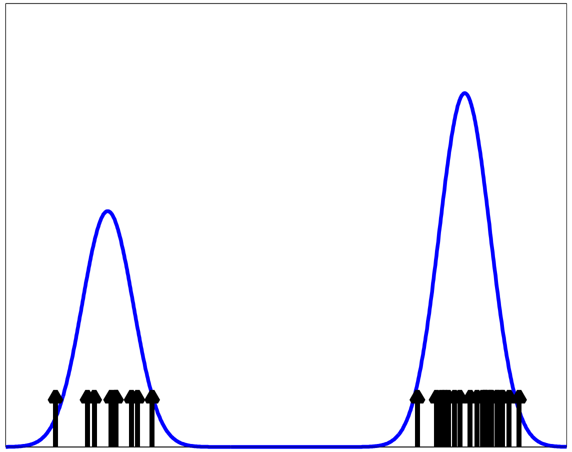
$$\mathbb{E}_q[f^2(x)w^2(x)] \geq \left(\mathbb{E}_q[|f(x)|w(x)]\right)^2 = \left(\int |f(x)|p(x)\,dx\right)^2$$

- Estimator is *asymptotically unbiased*, and *minimum-variance proposal distribution* is

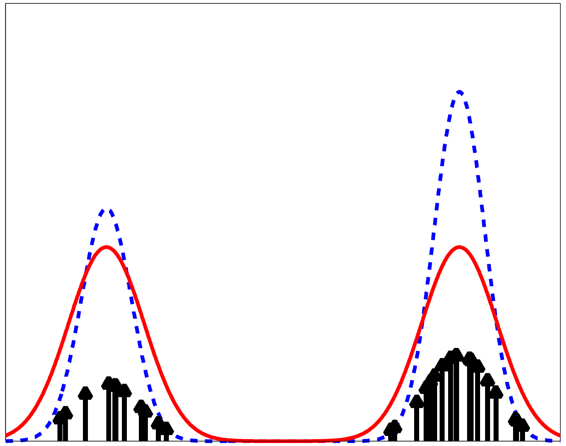
$$\hat{q}(x) \propto |f(x)|p(x)$$

For evaluation of f(x), this is more efficient than sampling from target p(x)!

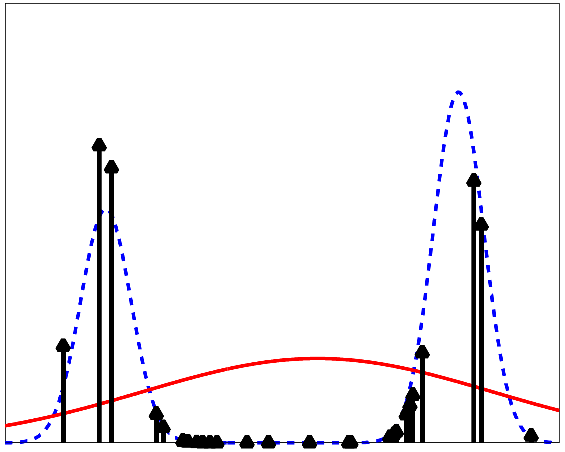
Selecting Proposal Distributions



Target Distribution



Good Proposal

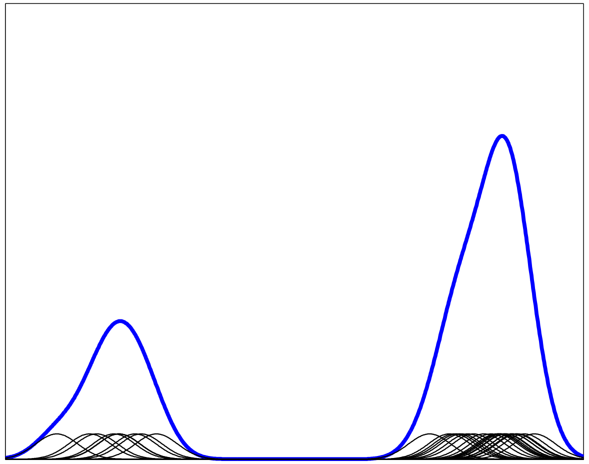


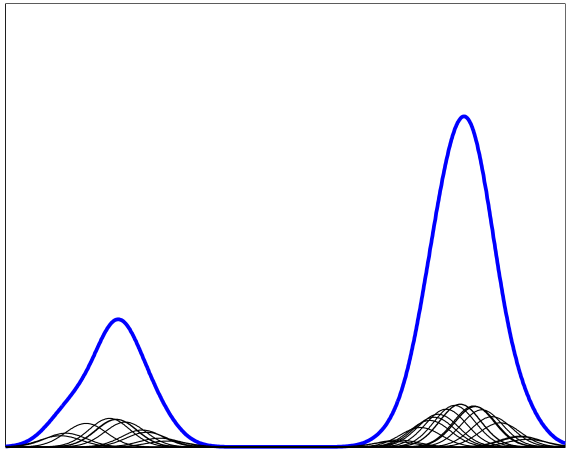
Poor Proposal

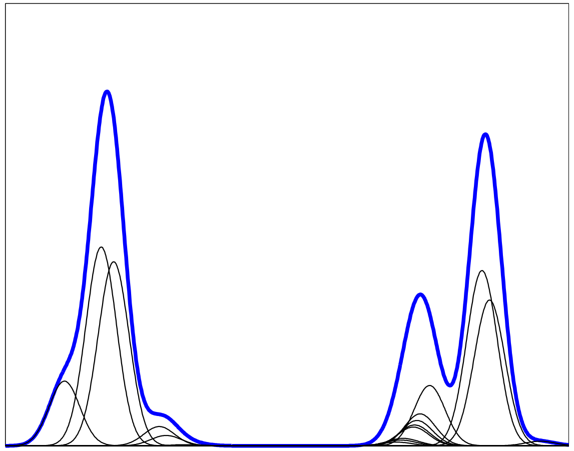
Kernel or Parzen window estimators interpolate to predict density:

$$\hat{p}(x) = \sum_{\ell=1}^{L} w^{(\ell)} \mathcal{N}(x; x^{(\ell)}, \Lambda)$$

$$w^{(\ell)} \propto \frac{p(x^{(\ell)})}{q(x^{(\ell)})}$$

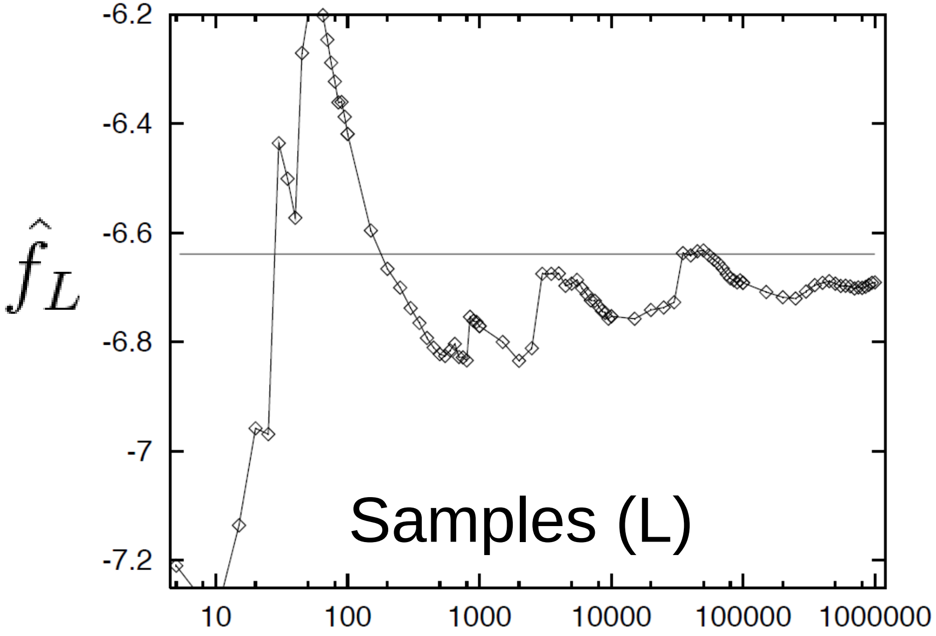




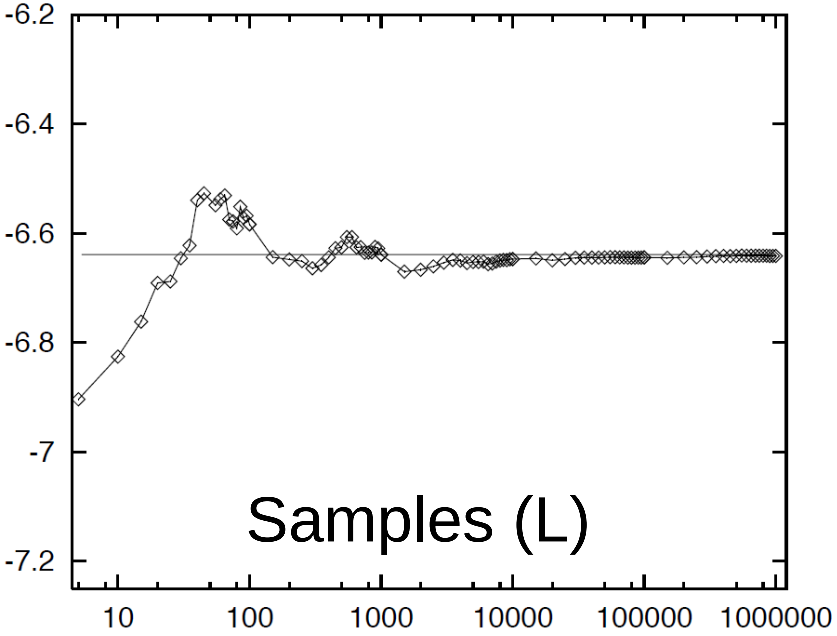


Selecting Proposal Distributions

- For a toy one-dimensional, heavy-tailed target distribution:



Gaussian Proposal



Cauchy (Student's-t) Proposal

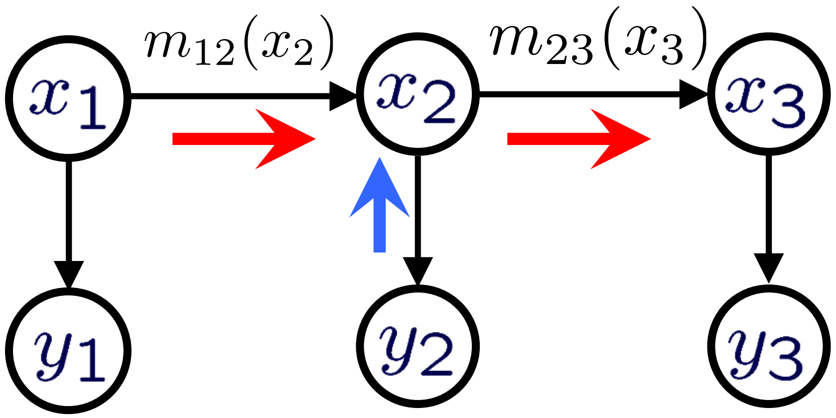
Empirical variance of weights may not predict estimator variance!

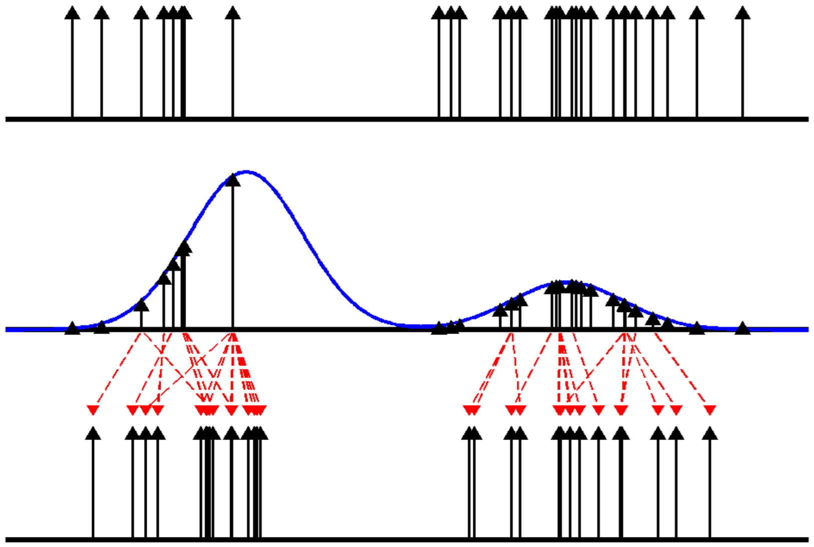
- Always (asymptotically) unbiased, but variance of estimator can be enormous unless weight function bounded above:

$$\mathbb{E}_q[\hat{f}_L] = \mathbb{E}_p[f] \qquad \mathrm{Var}_q[\hat{f}_L] = \frac{1}{L}\mathrm{Var}_q[f(x)w(x)] \qquad w(x) = \frac{p(x)}{q(x)}$$

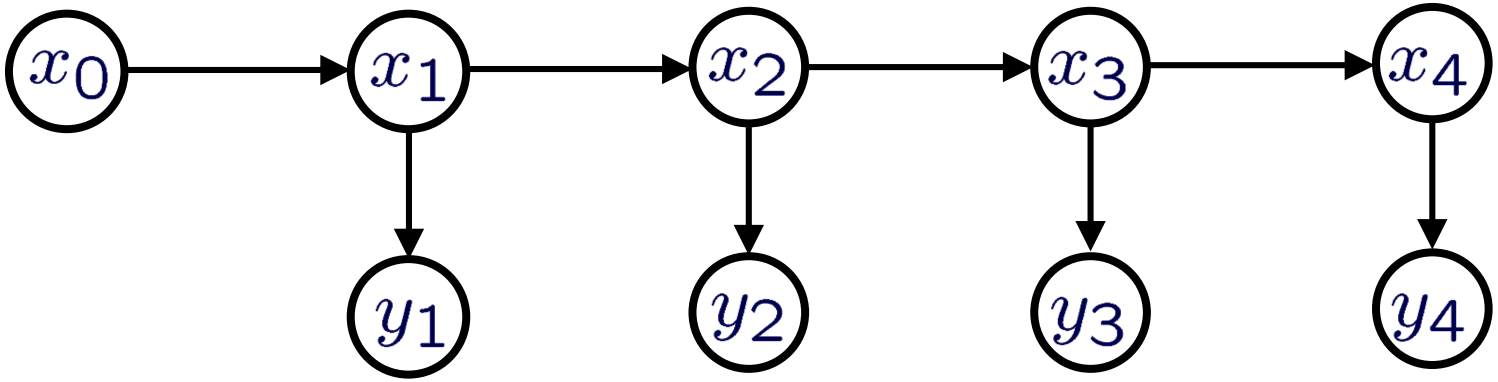
CS242: Lecture 6B Outline

- Importance sampling and likelihood weighting
- Sequential Monte Carlo: Particle Filters





Non-linear State Space Models



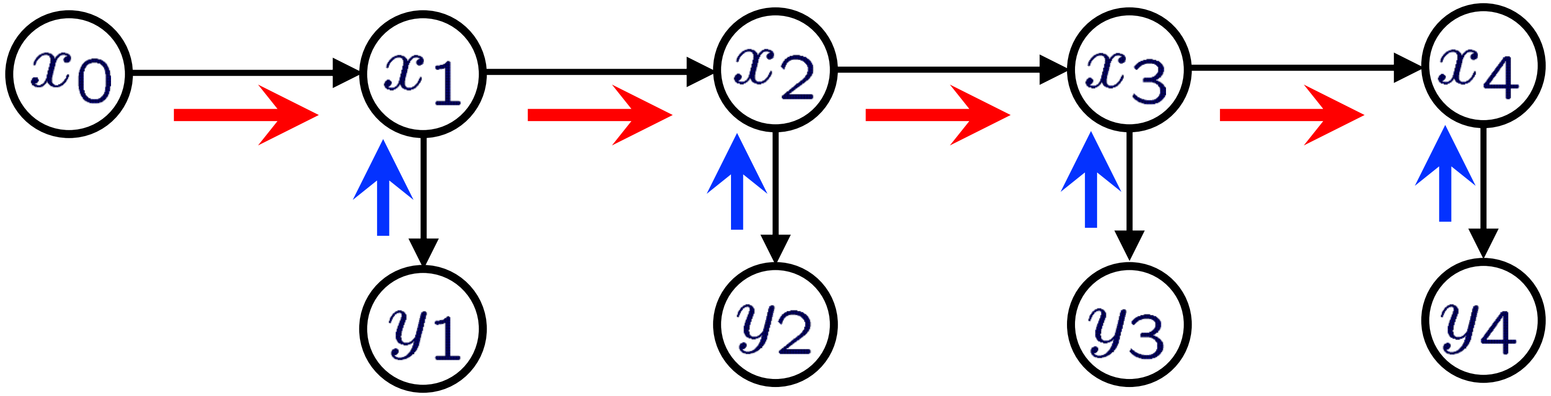
$x_t \in \mathbb{R}^d$

$y_t \in \mathbb{R}^k$

$$x_{t+1} = f(x_t, w_t) \qquad w_t \sim \mathcal{F}$$
$$y_t = g(x_t, v_t) \qquad v_t \sim \mathcal{G}$$

- State dynamics and measurements given by potentially complex *nonlinear functions*
- Noise sampled from *non-Gaussian* distributions
- Usually no closed form for messages or marginals

Sequential Importance Sampling



- Suppose interested in some complex, global function of state:

$$\mathbb{E}[f] = \int f(x)p(x \mid y)\, dx \approx \sum_{\ell=1}^{L} w_\ell f(x^{(\ell)}) \qquad w_\ell \propto \frac{p(x^{(\ell)} \mid y)}{q(x^{(\ell)} \mid y)} \qquad x^{(\ell)} \sim q(x \mid y)$$

- Could use Markov structure to construct efficient proposal:

$$q(x \mid y) = q(x_0) \prod_{t=1}^{T} q(x_t \mid x_{t-1}, y_t)$$

$$q(x_t \mid x_{t-1}, y_t) \approx p(x_t \mid x_{t-1}, y)$$

$$w_\ell^t \propto w_\ell^{t-1} \frac{p(x_t^{(\ell)} \mid x_{t-1}^{(\ell)}) p(y_t \mid x_t^{(\ell)})}{q(x_t^{(\ell)} \mid x_{t-1}^{(\ell)}, y_t)}$$

Weights will become degenerate, with most approaching zero

Particle Resampling

$$p(x_t \mid y_{\bar{t}}) \approx \sum_{\ell=1}^{L} \omega_t^{(\ell)} \delta_{x_t^{(\ell)}}(x_t)$$

where $\quad y_{\bar{t}} = \{y_1, \ldots, y_t\}$

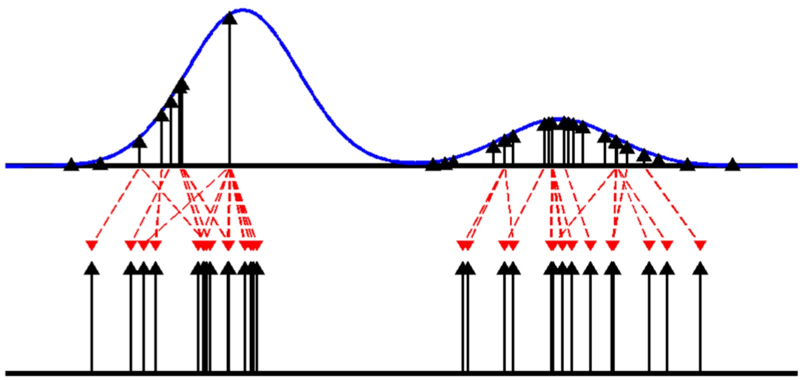
$$p(x_t \mid y_{\bar{t}}) \approx \sum_{\ell=1}^{L} \frac{1}{L} \delta_{\bar{x}_t^{(\ell)}}(x_t)$$

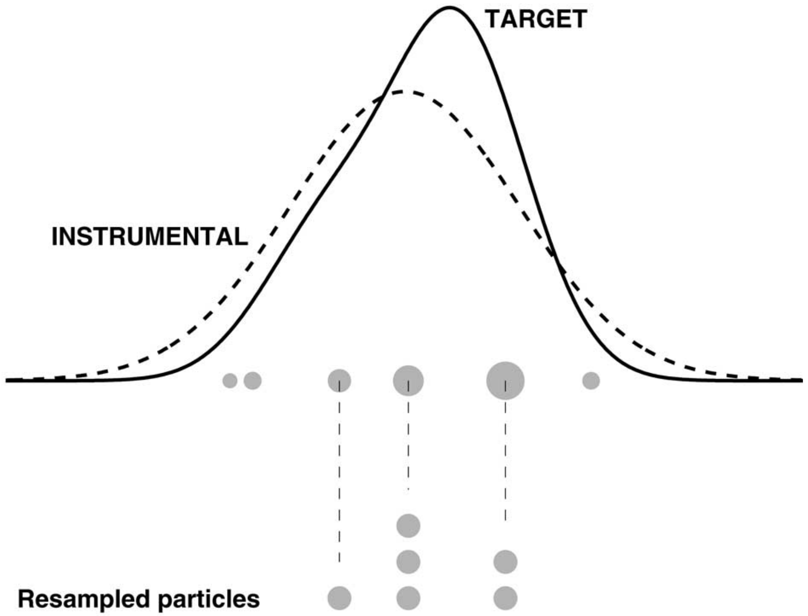
$$\bar{x}_t^{(\ell)} = x_t^{(j_\ell)}$$

$$j_\ell \sim \mathrm{Cat}(\omega_t)$$

Resample with replacement produces random discrete distribution with same mean as original distribution

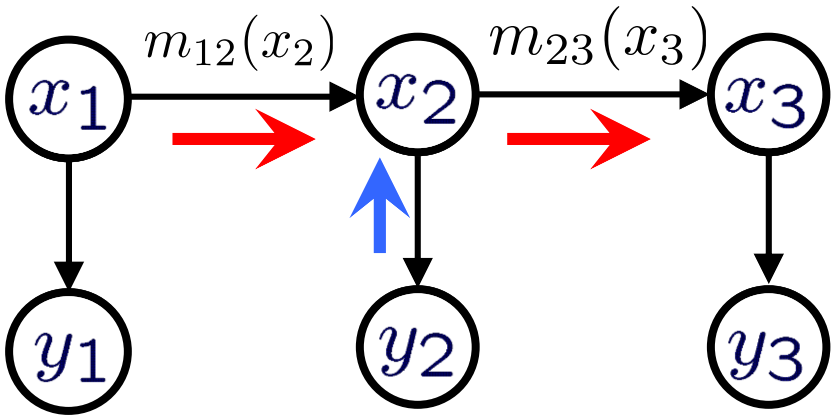
While remaining unbiased, resampling avoids degeneracies in which most weights go to zero





Particle Filtering Algorithms

- Represent state estimates using a set of samples
- Propagate over time using *sequential* *importance sampling* *with* *resampling*



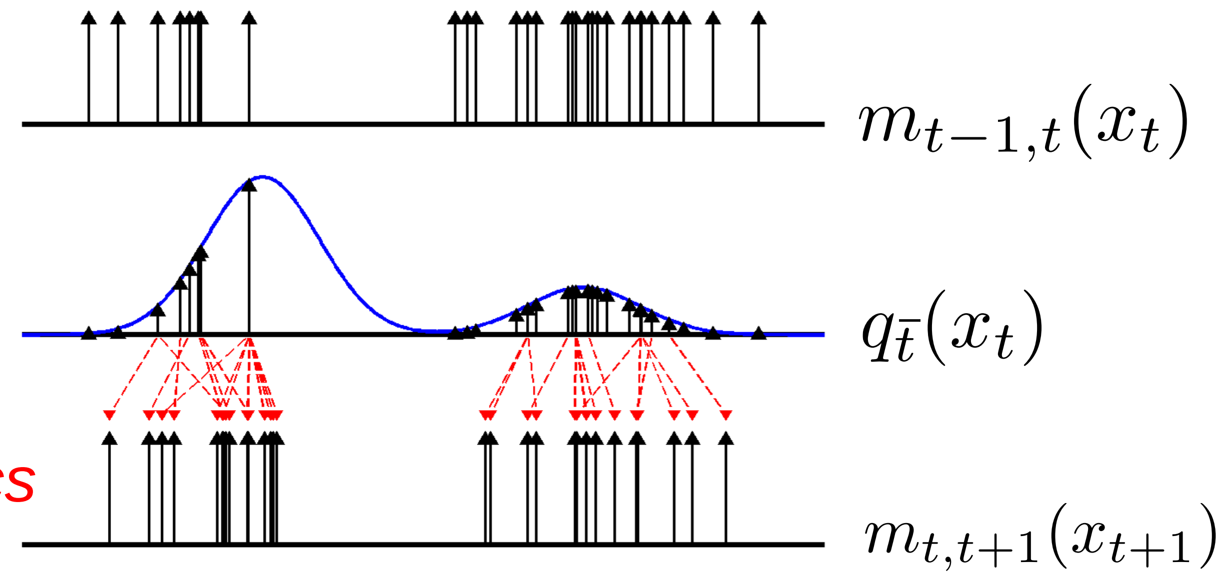
Sample-based density estimate

↓

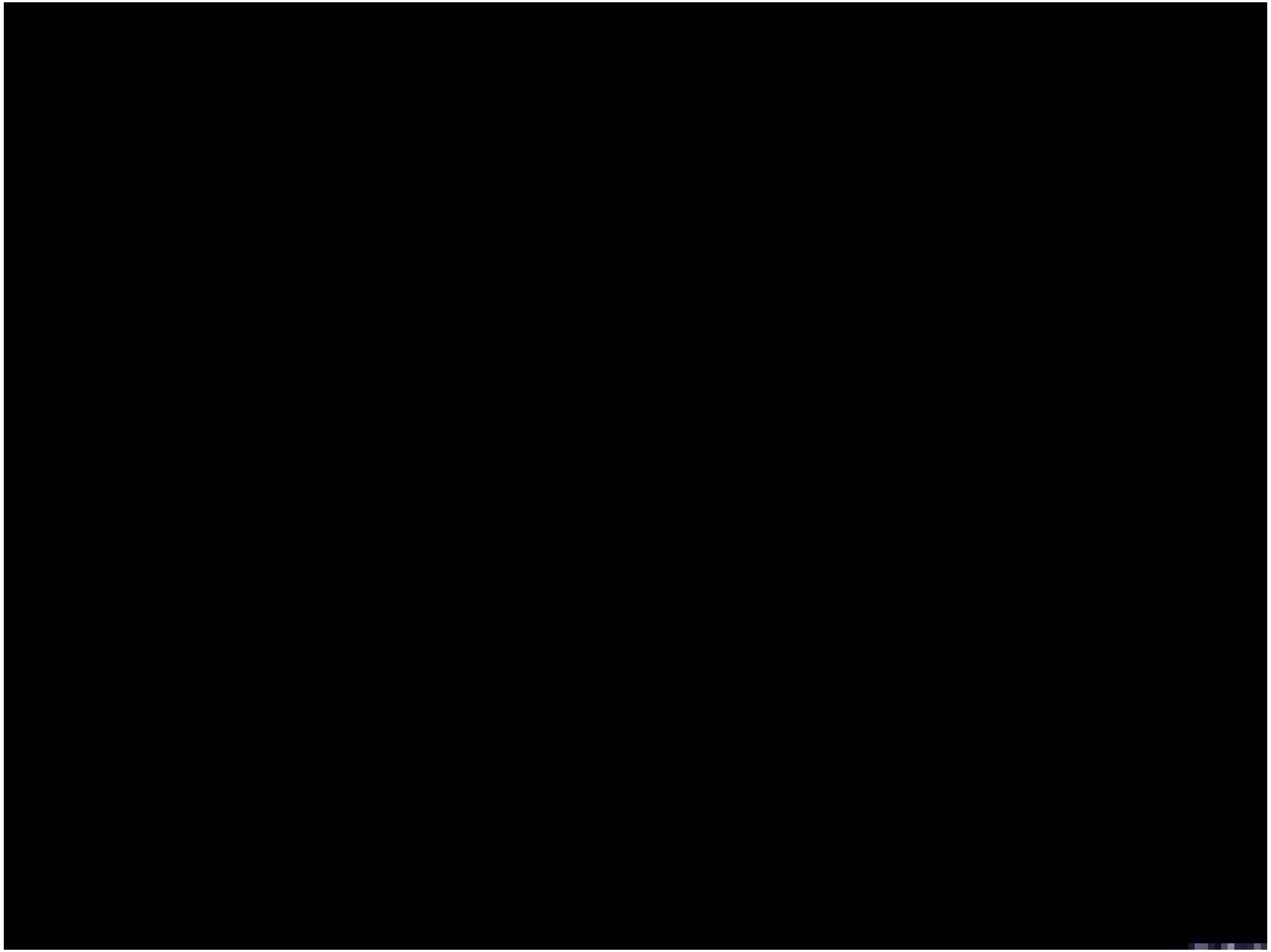
Weight by observation likelihood

↓

Resample & propagate by dynamics

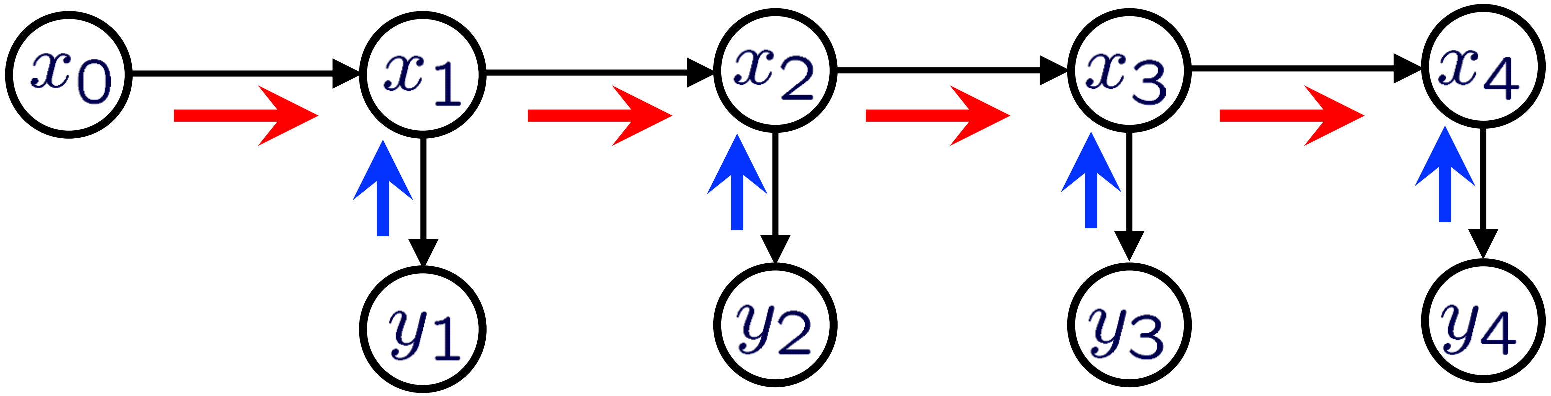


Particle Filters: The Movie



(M. Isard, 1996)

BP for State-Space Models



$$m_{t-1,t}(x_t) \propto p\left(x_t \mid y_{\overline{t-1}}\right) \quad \text{where} \quad y_{\bar{t}} = \{y_1, \ldots, y_t\}$$

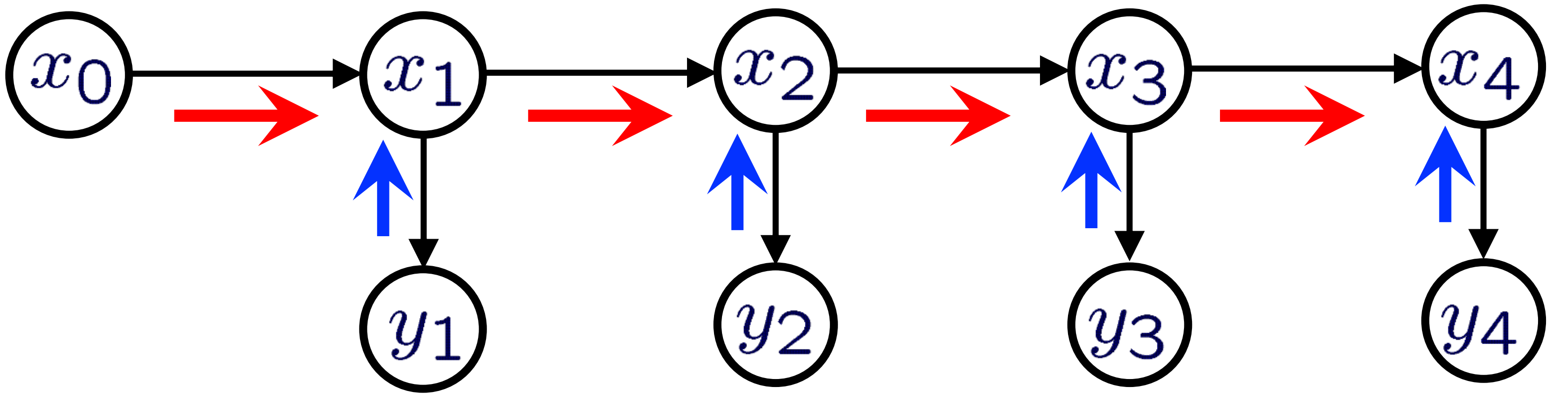
$$m_{t-1,t}(x_t)\, p(y_t \mid x_t) \propto p(x_t \mid y_{\bar{t}}) = q_{\bar{t}}(x_t)$$

Prediction (Integral/Sum step of BP):

$$m_{t-1,t}(x_t) \propto \int p(x_t \mid x_{t-1}) q_{\overline{t-1}}(x_{t-1})\, dx_{t-1}$$

Inference (Product step of BP): $q_{\bar{t}}(x_t) = \dfrac{1}{Z_t} m_{t-1,t}(x_t) p(y_t \mid x_t)$

Particle Filter: Measurement Update



- **Incoming message:** A set of *L* weighted particles

$$m_{t-1,t}(x_t) \approx \sum_{\ell=1}^{L} w_{t-1,t}^{(\ell)} \delta(x_t, x_t^{(\ell)}) \qquad \sum_{\ell=1}^{L} w_{t-1,t}^{(\ell)} = 1$$

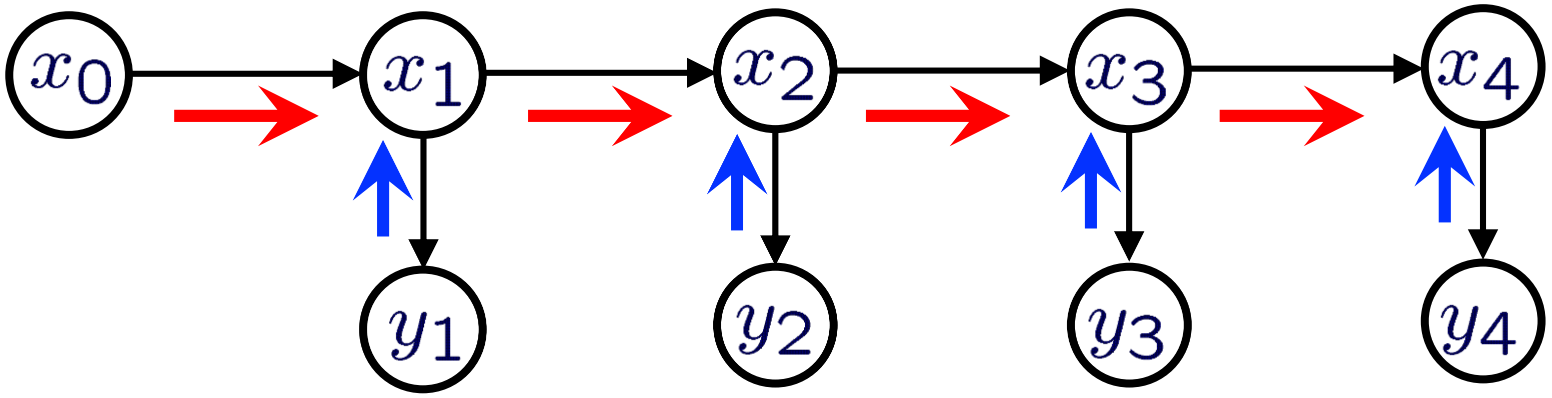
- **Bayes' Rule:** Posterior at particles proportional to prior times likelihood

$$q_{\bar{t}}(x_t) \propto m_{t-1,t}(x_t)\, p(y_t \mid x_t) \propto \sum_{\ell=1}^{L} w_{t-1,t}^{(\ell)}\, p(y_t \mid x_t^{(\ell)})\, \delta(x_t, x_t^{(\ell)})$$

$$q_{\bar{t}}(x_t) = \sum_{\ell=1}^{L} w_t^{(\ell)} \delta(x_t, x_t^{(\ell)}) \qquad w_t^{(\ell)} \triangleq \frac{w_{t-1,t}^{(\ell)}\, p(y_t \mid x_t^{(\ell)})}{\sum_{m=1}^{L} w_{t-1,t}^{(m)}\, p(y_t \mid x_t^{(m)})}$$

*Variance of importance weights **increases** with each update*

Particle Filter: Sample Propagation



- **State Posterior Estimate:** A set of *L* weighted particles

$$q_{\bar{t}}(x_t) = \sum_{\ell=1}^{L} w_t^{(\ell)} \delta(x_t, x_t^{(\ell)}) \qquad \sum_{\ell=1}^{L} w_t^{(\ell)} = 1$$

- **Prediction:** Sample next state conditioned on current particles

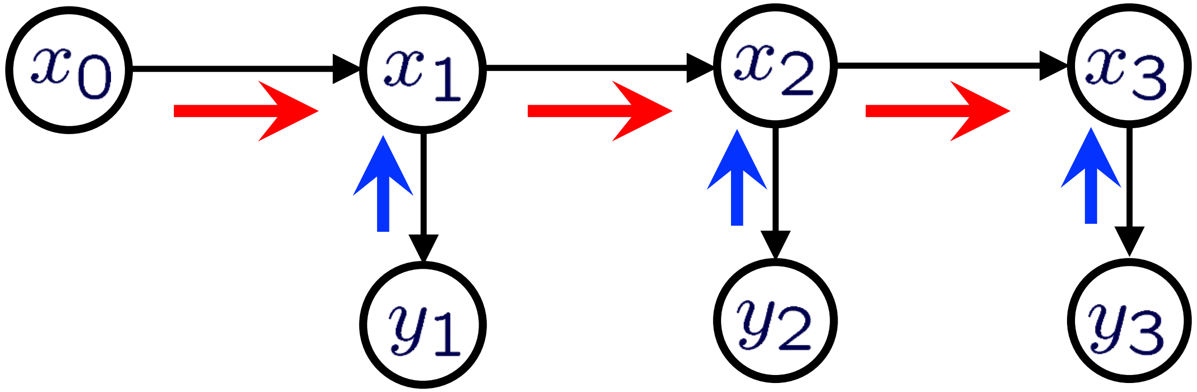
$$m_{t,t+1}(x_{t+1}) = \sum_{\ell=1}^{L} w_{t,t+1}^{(\ell)} \delta(x_{t+1}, x_{t+1}^{(\ell)})$$

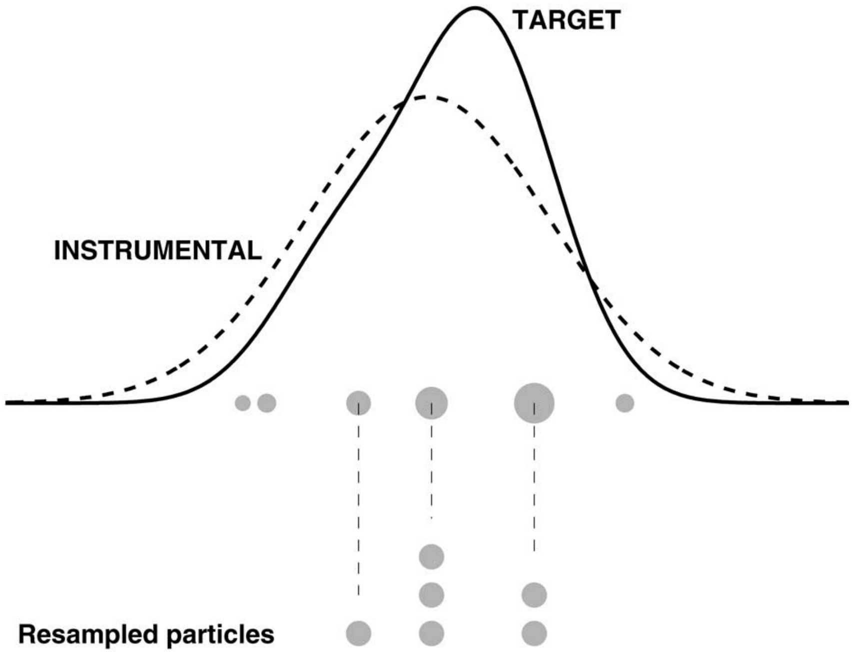
$$x_{t+1}^{(\ell)} \sim p(x_{t+1} \mid x_t^{(\ell)})$$

$$w_{t,t+1}^{(\ell)} = w_t^{(\ell)}$$

Assumption for now: Can exactly simulate temporal dynamics

Particle Filter: Resampling





- **State Posterior Estimate:**

$$q_{\bar{t}}(x_t) = \sum_{\ell=1}^{L} w_t^{(\ell)} \delta(x_t, x_t^{(\ell)})$$

- **Prediction:** Sample next state conditioned on randomly chosen particles

$$m_{t,t+1}(x_{t+1}) = \sum_{\ell=1}^{L} w_{t,t+1}^{(\ell)} \delta(x_{t+1}, x_{t+1}^{(\ell)})$$

$$\tilde{x}_t^{(\ell)} \sim q_{\bar{t}}(x_t)$$

$$x_{t+1}^{(\ell)} \sim p(x_{t+1} \mid \tilde{x}_t^{(\ell)})$$

$$w_{t,t+1}^{(\ell)} = 1/L$$

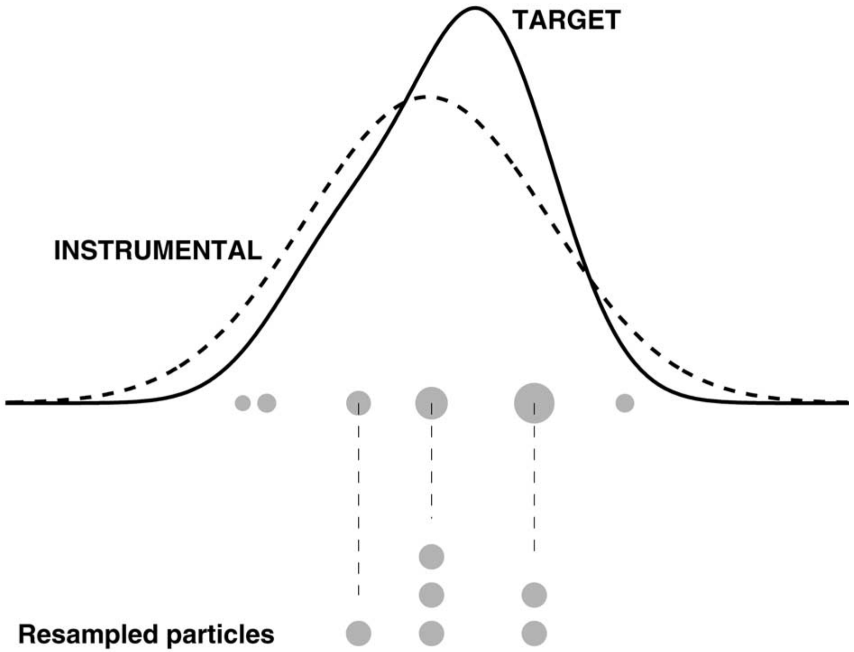
Resampling with replacement preserves expectations, but increases the variance of subsequent estimators

Particle Filter: Resampling

- **Effective Sample Size:**

$$L_{\text{eff}} = \left(\sum_{\ell=1}^{L} \left(w^{(\ell)} \right)^2 \right)^{-1}$$

$$1 \leq L_{\text{eff}} \leq L$$



- **State Posterior Estimate:**

$$q_{\bar{t}}(x_t) = \sum_{\ell=1}^{L} w_t^{(\ell)} \delta(x_t, x_t^{(\ell)})$$

- **Prediction:** Sample next state conditioned on randomly chosen particles

$$m_{t,t+1}(x_{t+1}) = \sum_{\ell=1}^{L} w_{t,t+1}^{(\ell)} \delta(x_{t+1}, x_{t+1}^{(\ell)})$$

$$\tilde{x}_t^{(\ell)} \sim q_{\bar{t}}(x_t)$$

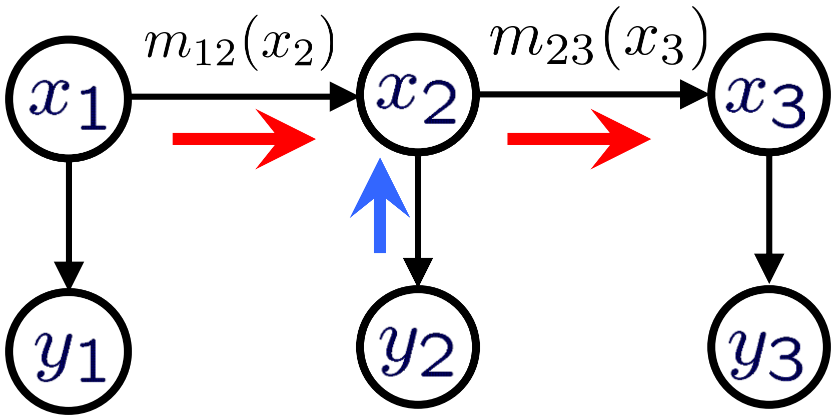
$$x_{t+1}^{(\ell)} \sim p(x_{t+1} \mid \tilde{x}_t^{(\ell)})$$

$$w_{t,t+1}^{(\ell)} = 1/L$$

Resampling with replacement preserves expectations, but increases the variance of subsequent estimators

Particle Filtering Algorithms

- Represent state estimates using a set of samples
- Propagate over time using *sequential importance sampling with resampling*



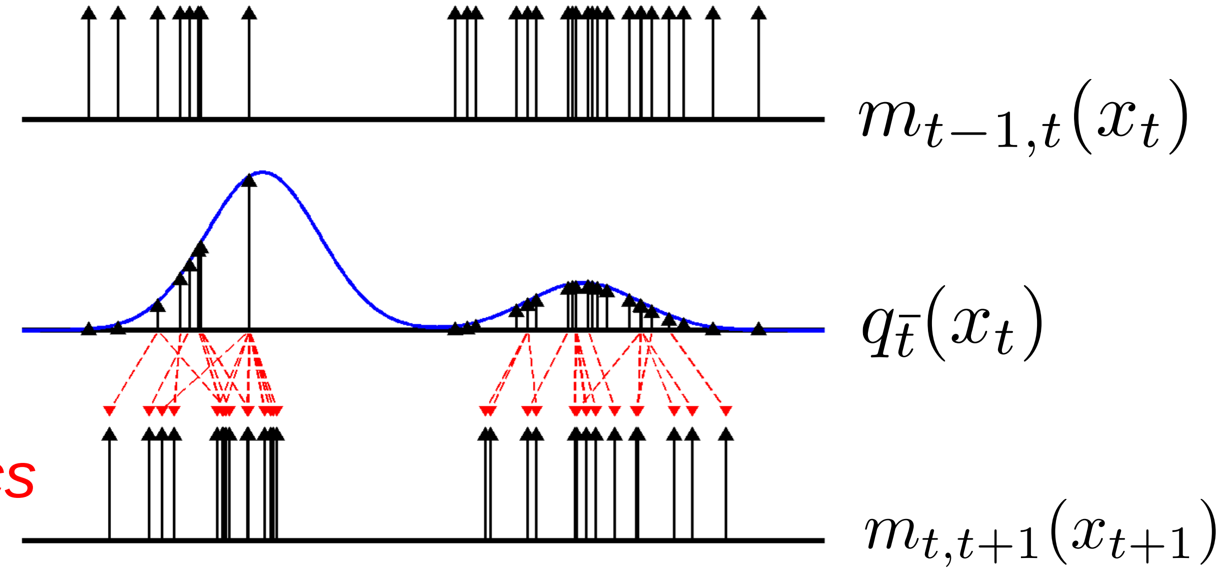
Sample-based density estimate

↓

Weight by observation likelihood

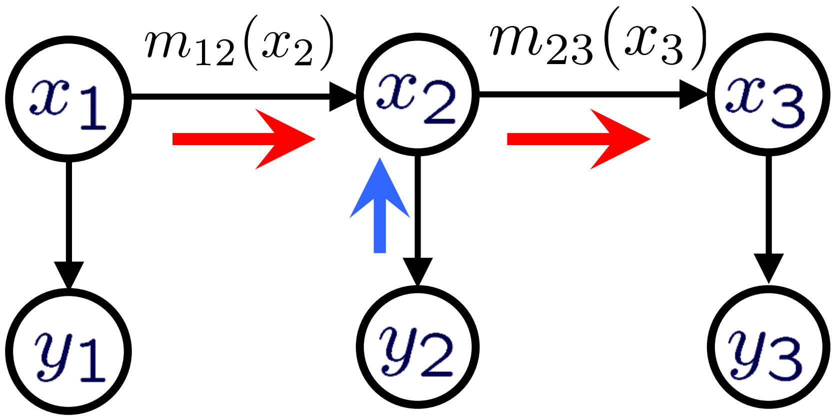
↓

Resample & propagate by dynamics



Bootstrap Particle Filter Summary

- Represent state estimates using a set of samples
- Propagate over time using *sequential importance sampling* with *resampling*



- Assume sample-based approximation of incoming message:

$$m_{t-1,t}(x_t) = p(x_t \mid y_{t-1}, \ldots, y_1) \approx \sum_{\ell=1}^{L} \frac{1}{L} \delta_{x_t^{(\ell)}}(x_t)$$

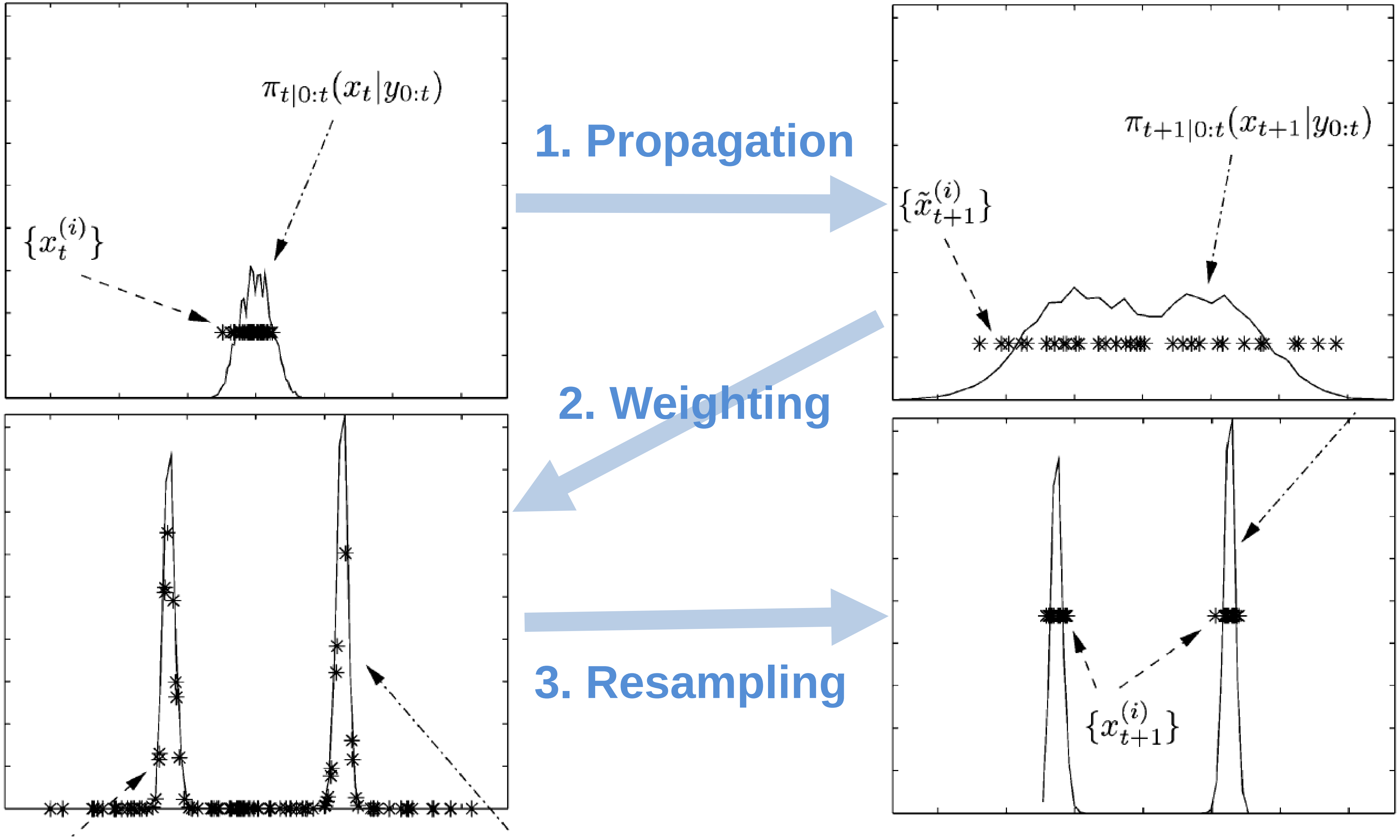
- Account for observation via importance weights:

$$p(x_t \mid y_t, y_{t-1}, \ldots, y_1) \approx \sum_{\ell=1}^{L} w_t^{(\ell)} \delta_{x_t^{(\ell)}}(x_t) \qquad w_t^{(\ell)} \propto p(y_t \mid x_t^{(\ell)})$$

- Sample from forward dynamics distribution of next state:

$$m_{t,t+1}(x_{t+1}) \approx \sum_{m=1}^{L} \frac{1}{L} \delta_{x_{t+1}^{(m)}}(x_{t+1}) \qquad x_{t+1}^{(m)} \sim \sum_{\ell=1}^{L} w_t^{(\ell)} p(x_{t+1} \mid x_t^{(\ell)})$$

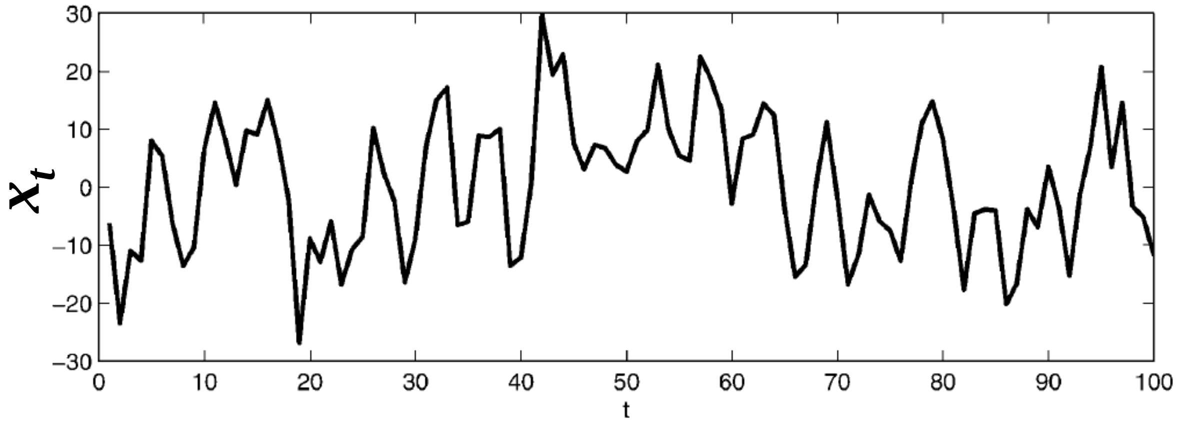
Bootstrap Particle Filter Summary

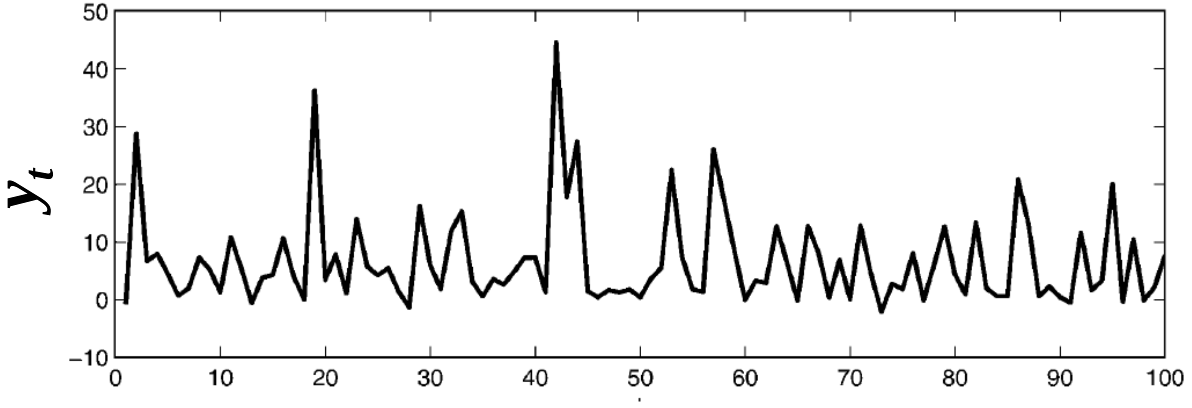


[Source: Cappe]

Toy Nonlinear Model

Nonlinear dynamics and observation model…





$$x_t = \frac{x_{t-1}}{2} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t) + u_t$$

Dynamics

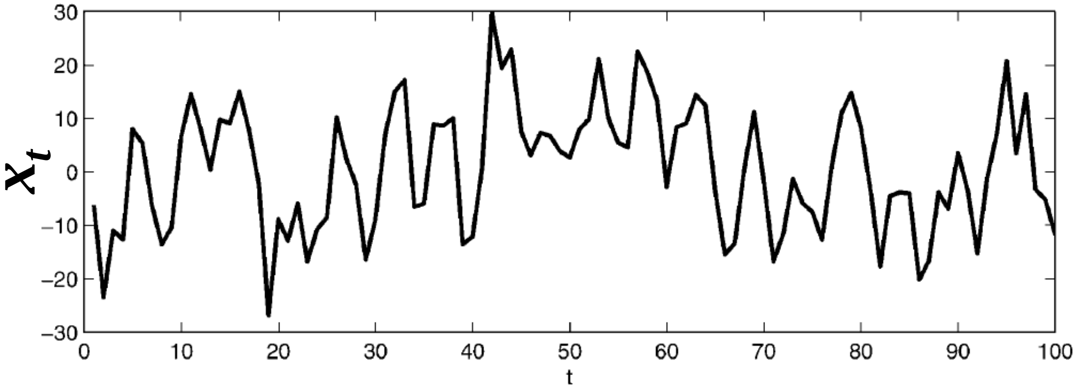
$$y_t = \frac{x_t^2}{20} + v_t$$

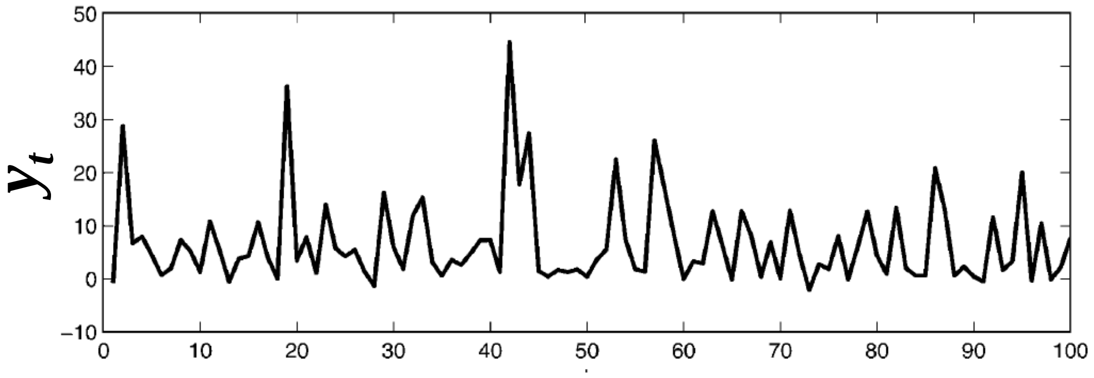
Measurement

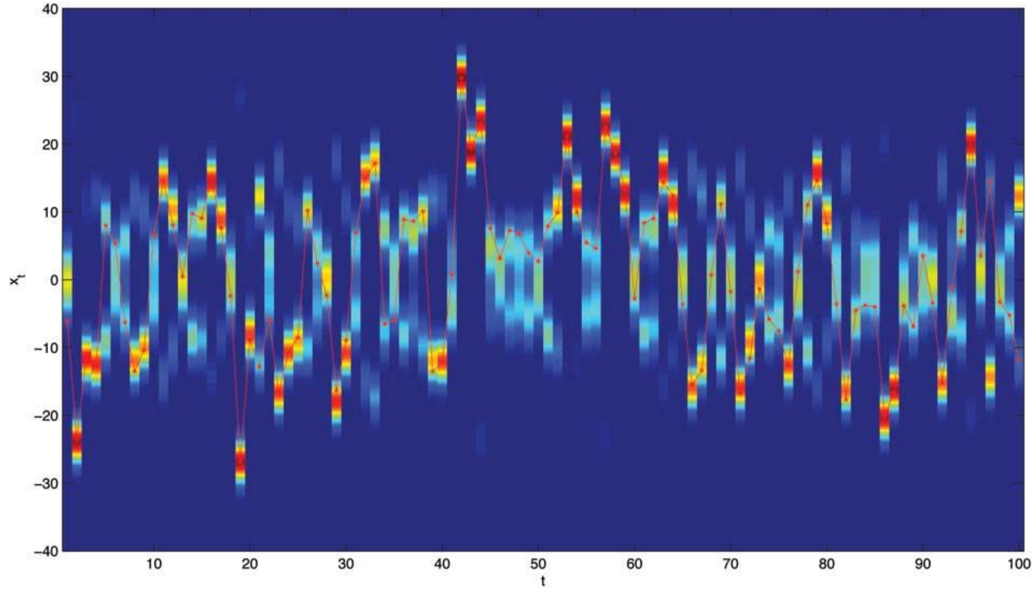
Gaussian Noise $N(0,\sigma^2)$

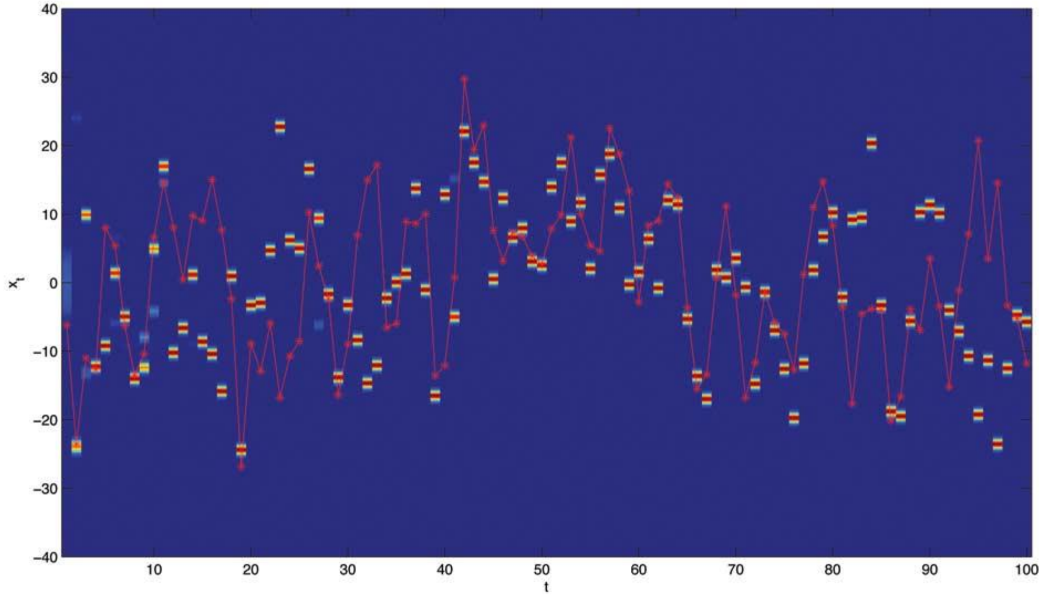
…filter equations lack closed form.

Toy Nonlinear Model







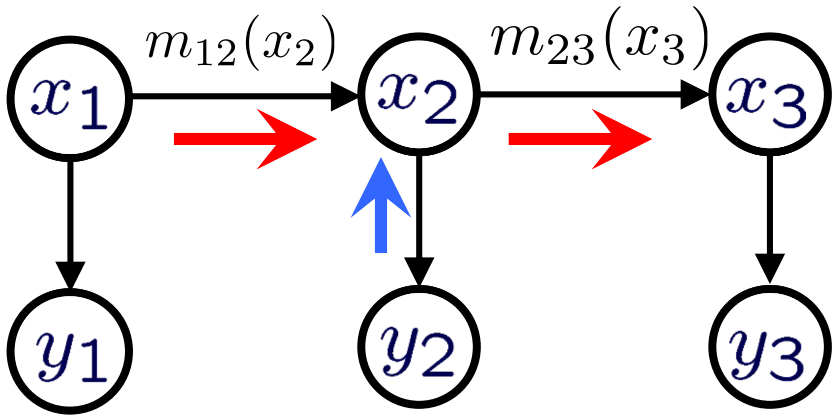


Particle Filter Marginal KDEs

Full Sequence Importance Sampling

What is the probability that a state sequence, sampled from the prior model, is consistent with all observations?

A More General Particle Filter



- Assume sample-based approximation of previous state's marginal:

$$p(x_{t-1} \mid y_{t-1}, \ldots, y_1) \approx \sum_{\ell=1}^{L} \frac{1}{L} \delta_{x_{t-1}^{(\ell)}}(x_{t-1})$$

- Sample from a *proposal distribution q*:

$$x_t^{(\ell)} \sim q(x_t \mid x_{t-1}^{(\ell)}, y_t) \approx p(x_t \mid x_{t-1}^{(\ell)}, y_t)$$

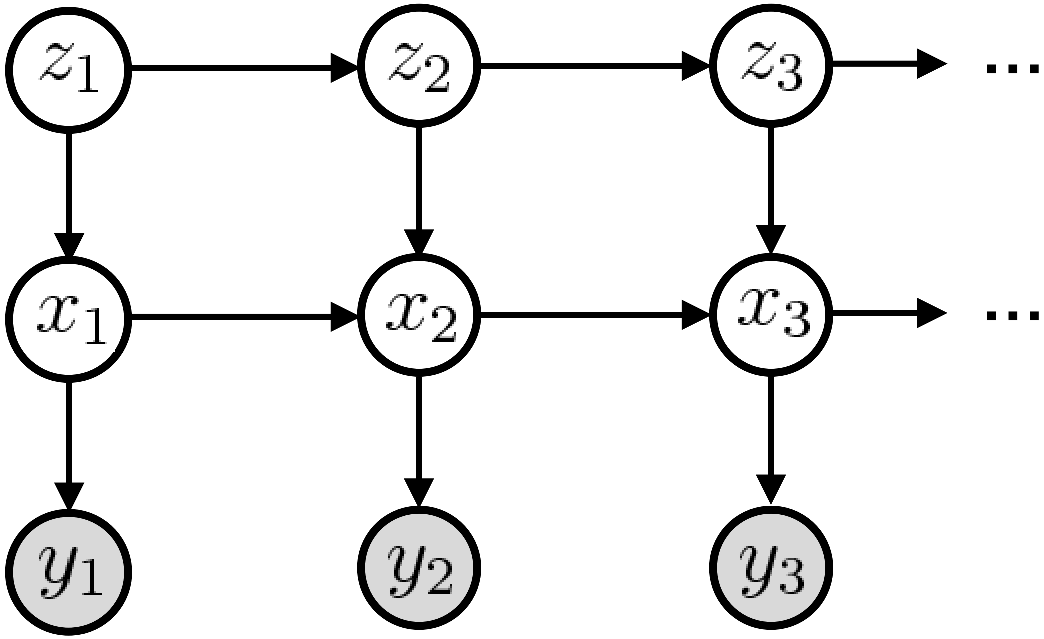
- Account for *observation and proposal* via importance weights:

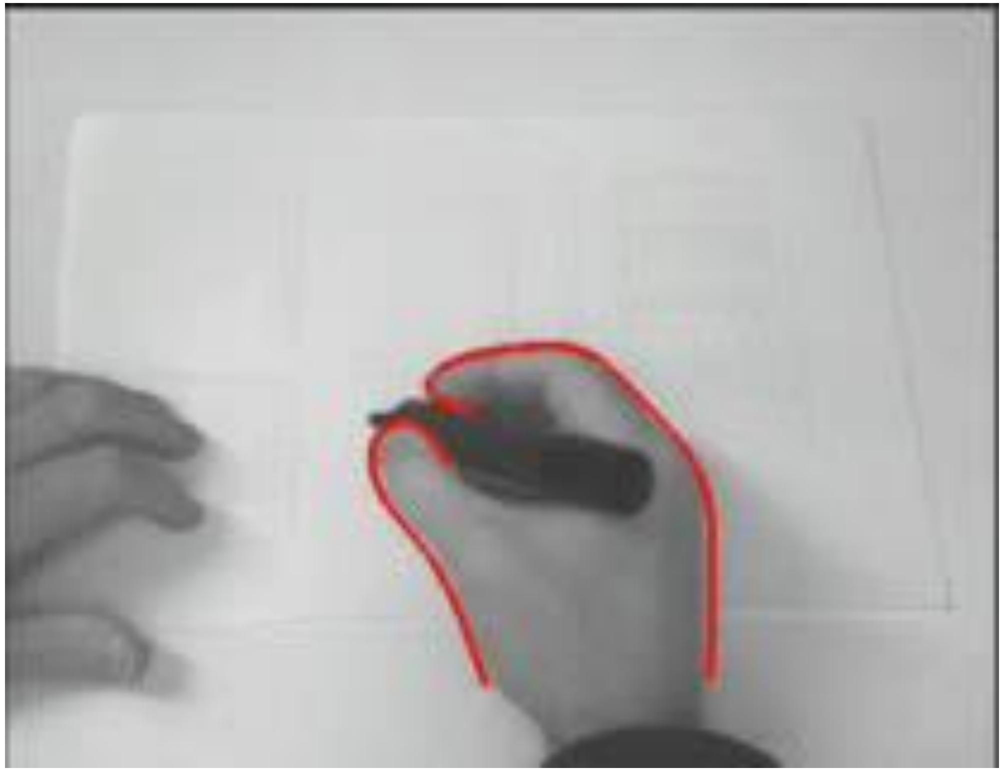
$$w_t^{(\ell)} \propto \frac{p(x_t^{(\ell)} \mid x_{t-1}^{(\ell)}) p(y_t \mid x_t^{(\ell)})}{q(x_t^{(\ell)} \mid x_{t-1}^{(\ell)}, y_t)}$$

- Resample to avoid particle degeneracy:

$$p(x_t \mid y_t, \ldots, y_1) \approx \sum_{\ell=1}^{L} \frac{1}{L} \delta_{x_t^{(\ell)}}(x_t) \qquad x_t^{(\ell)} \sim \sum_{m=1}^{L} w_t^{(m)} \delta_{x_t^{(m)}}(x_t)$$

Switching State-Space Model





Colors indicate 3 writing modes

[Video: Isard & Blake, ICCV 1998.]

Discrete switching state:

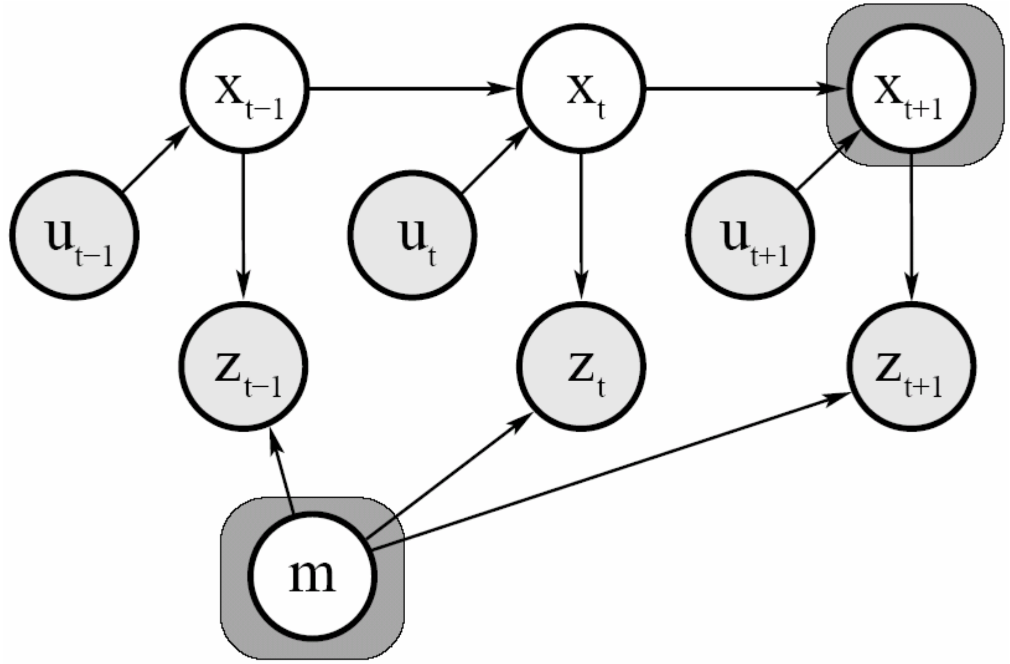
$$z_t \mid z_{t-1} \sim \mathrm{Cat}(\pi(z_{t-1}))$$ With stochastic transition matrix π

Switching state selects dynamics:

$$x_t \mid x_{t-1} \sim \mathcal{N}(A_{z_t} x_{t-1}, \Sigma_{z_t})$$ (e.g. Nonlinear Gaussian)

Example: Particle Filters for SLAM

Simultaneous Localization & Mapping (FastSLAM, Montemerlo 2003)



$p(x_t, m \mid z_{1:t}, u_{1:t})$

x_t = State of the robot at time t

m = Map of the environment

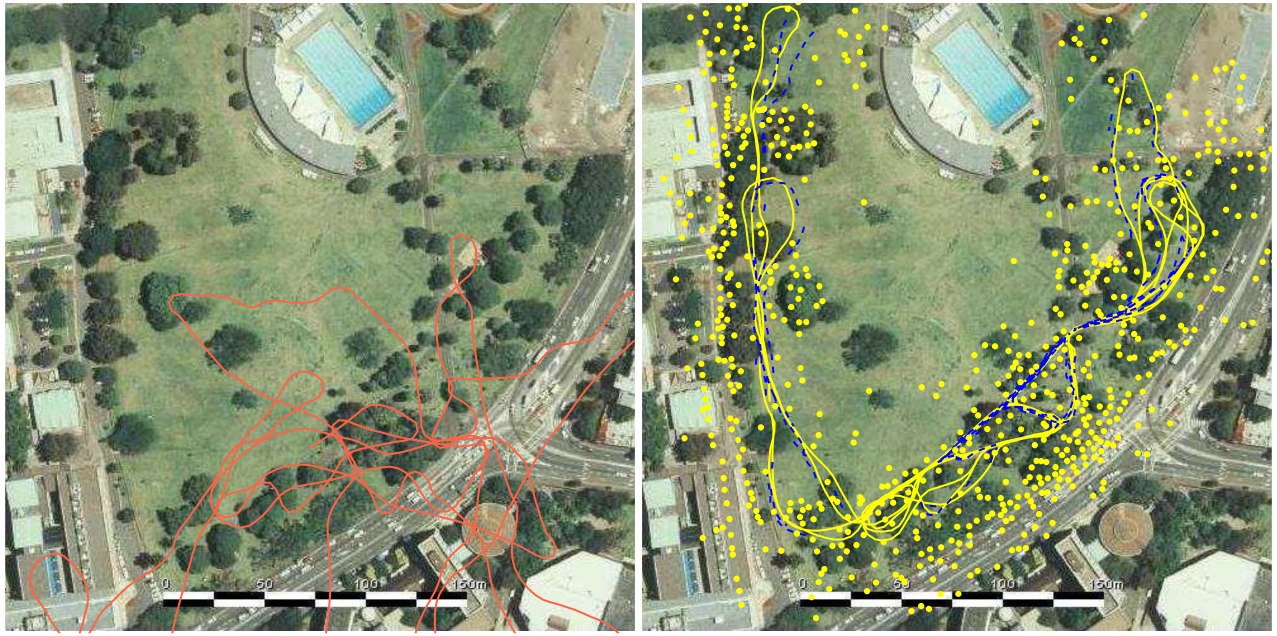
$z_{1:t}$ = Sensor inputs from time 1 to t

$u_{1:t}$ = Control inputs from time 1 to t

Raw odometry (controls)

True trajectory (GPS)

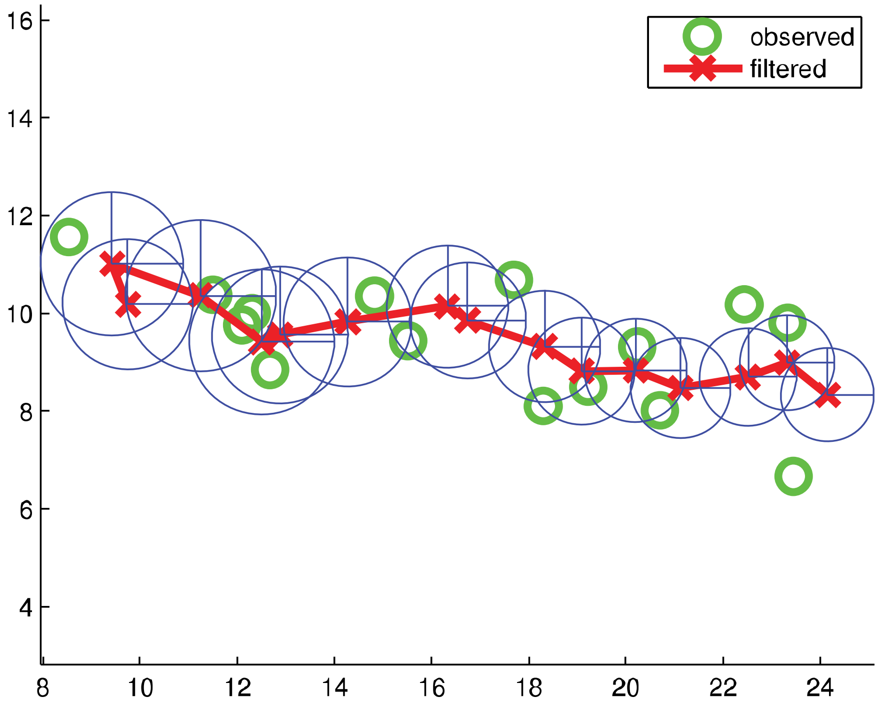
Inferred trajectory & landmarks



Dynamical System Inference

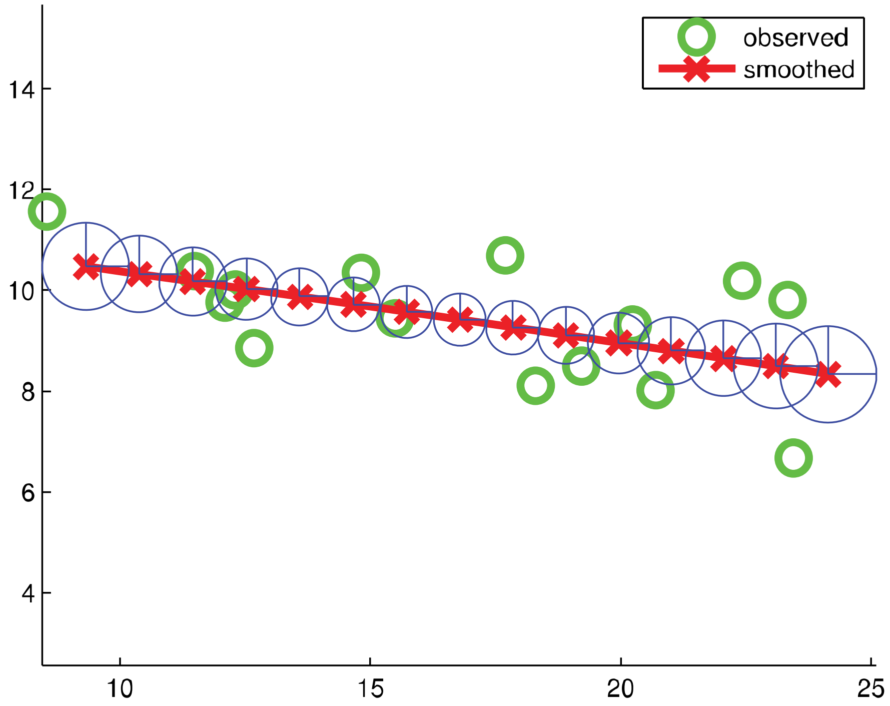
Define shorthand notation: $y_1^{t-1} \triangleq \{y_1, \ldots, y_{t-1}\}$

Filtering



Compute $p(x_t \mid y_1^t)$ at each time t

Smoothing

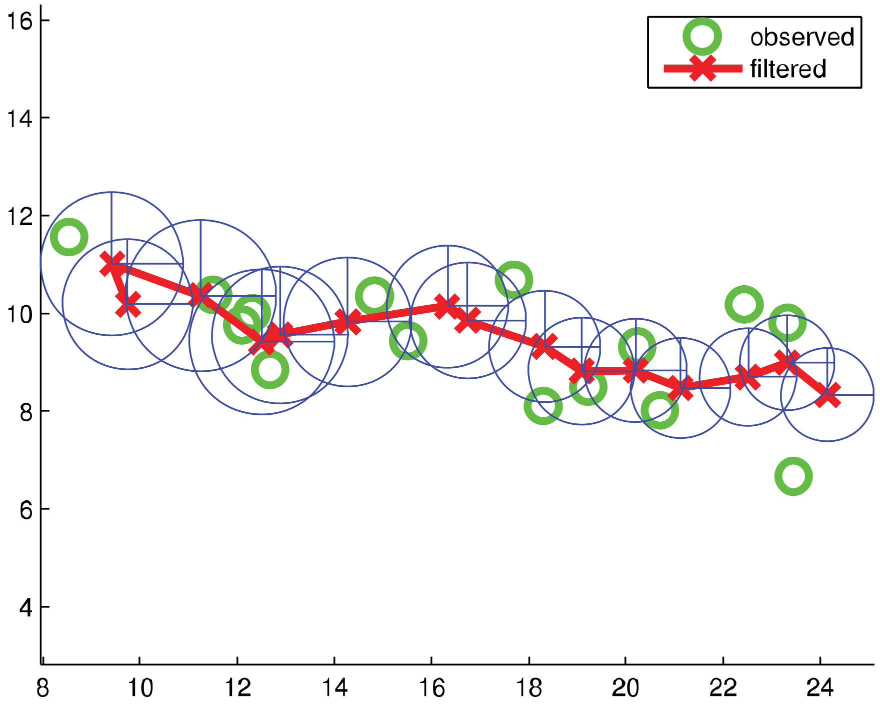


Compute full posterior marginal $p(x_t \mid y_1^T)$ at each time t

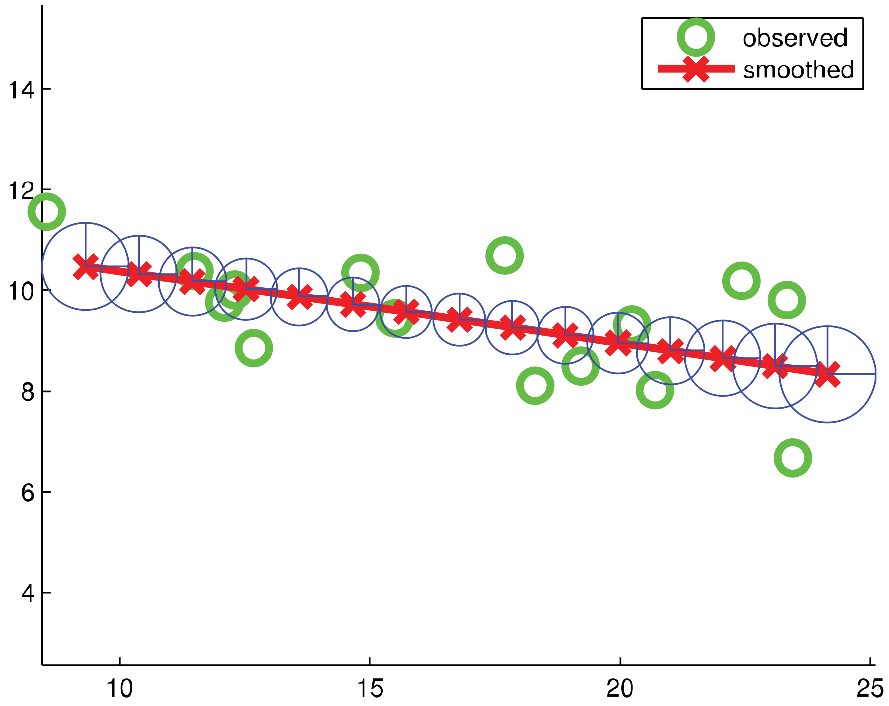
Dynamical System Inference

Define shorthand notation: $y_1^{t-1} \triangleq \{y_1, \ldots, y_{t-1}\}$

Filtering

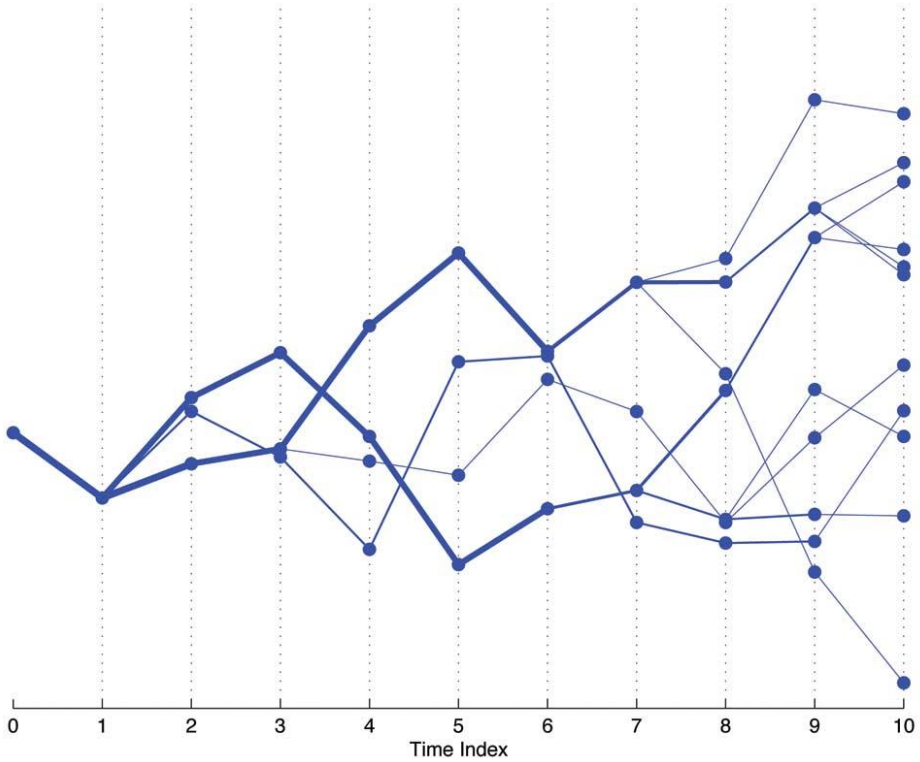


Smoothing



If estimates at time t are not needed *immediately*, then better *smoothed* estimates are possible by incorporating future observations

A Note On Smoothing



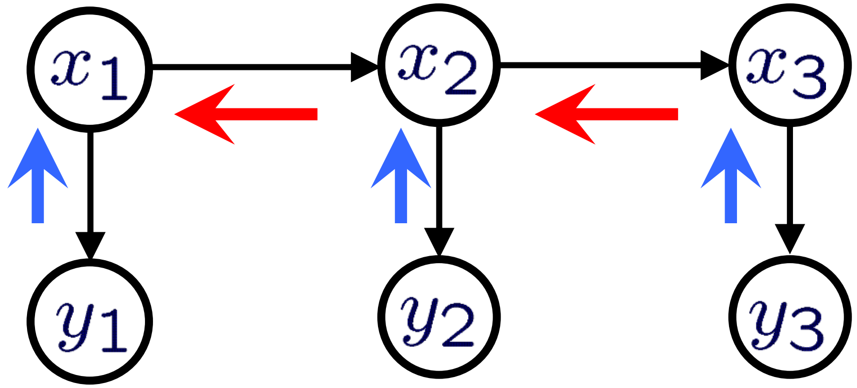
- ➢ Each resampling step discards states and they cannot subsequently restored
- ➢ Resampling introduces dependence across trajectories (common ancestors)
- ➢ Smoothed marginal estimates are generally poor
- ➢ Backwards simulation improves estimates of smoothed trajectories

Particle Filter Smoothing

Smoothing distribution factorizes as,

$$p(x_1^T \mid y_1^T) = p(x_T \mid y_1^T) \prod_{t=1}^{T-1} p(x_t \mid x_{t+1}, y_1^T)$$

$$= p(x_T \mid y_1^T) \prod_{t=1}^{T-1} p(x_t \mid x_{t+1}, y_1^t)$$



Markov property removes dependence on $y_{t+1} \dots y_T$

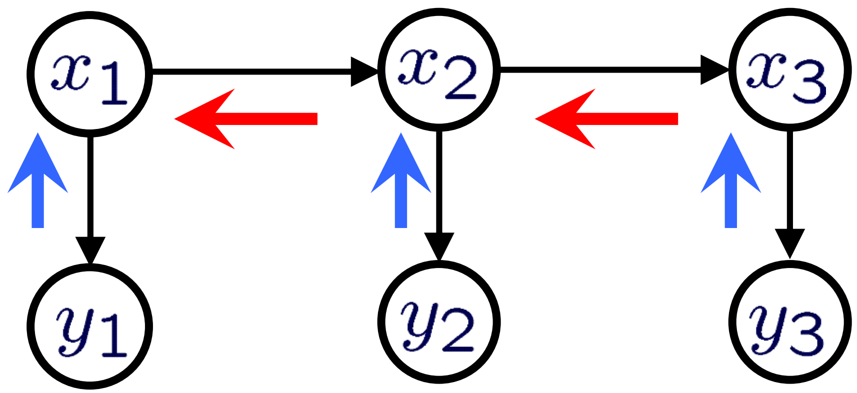
Filter distribution at time T

Suggests an algorithm to sample from $p(x_1^T \mid y_1^T)$:

1. Compute and store filter marginals, $p(x_t \mid y_1^t)$ for t=1,…,T
2. Sample final state from full posterior marginal, $x_T \sim p(x_T \mid y_1^T)$
3. Sample in reverse for t=(T-1),(T-2),…,2,1 from, $x_t \sim p(x_t \mid x_{t+1}, y_1^t)$

Use resampling idea to sample from current particle trajectories in reverse

Particle Filter Smoothing



Reverse conditional given by def'n of conditional prob.:

$$p(x_t \mid x_{t+1}, y_1^t) = \frac{p(x_{t+1} \mid x_t)p(x_t \mid y_1^t)}{p(x_{t+1} \mid y_1^t)}$$

$$\propto p(x_{t+1} \mid x_t)p(x_t \mid y_1^t)$$

Forward pass sample-based filter marginal estimates:

$$p(x_t \mid y_1^t) \approx \sum_{\ell=1}^{L} w_t^{(\ell)} \delta(x_t - x_t^{(\ell)})$$

Thus particle estimate of reverse prediction is:

$$p(x_t \mid x_{t+1}, y_1^T) \approx \sum_{\ell=1}^{L} \rho_t^{(\ell)}(x_{t+1}) \delta(x_t - x_t^{(i)}) \quad \text{where} \quad \rho_t^{(i)}(x_{t+1}) = \frac{w_t^{(i)} p(x_{t+1} \mid x_t^{(i)})}{\sum_{l=1}^{L} w_t^{(l)} p(x_{t+1} \mid x_t^{(l)})}$$

Particle Filter Smoothing

Algorithm 5 Particle Smoother

for $t = 0$ to T **do** ▷ Forward Pass Filter

Run Particle filter, storing at each time step the particles and weights $\{x_t^{(i)}, \omega_t^{(i)}\}_{1 \le i \le L}$

end for

Choose $\widetilde{x}_T = x_T^{(i)}$ with probability $\omega_t^{(i)}$.

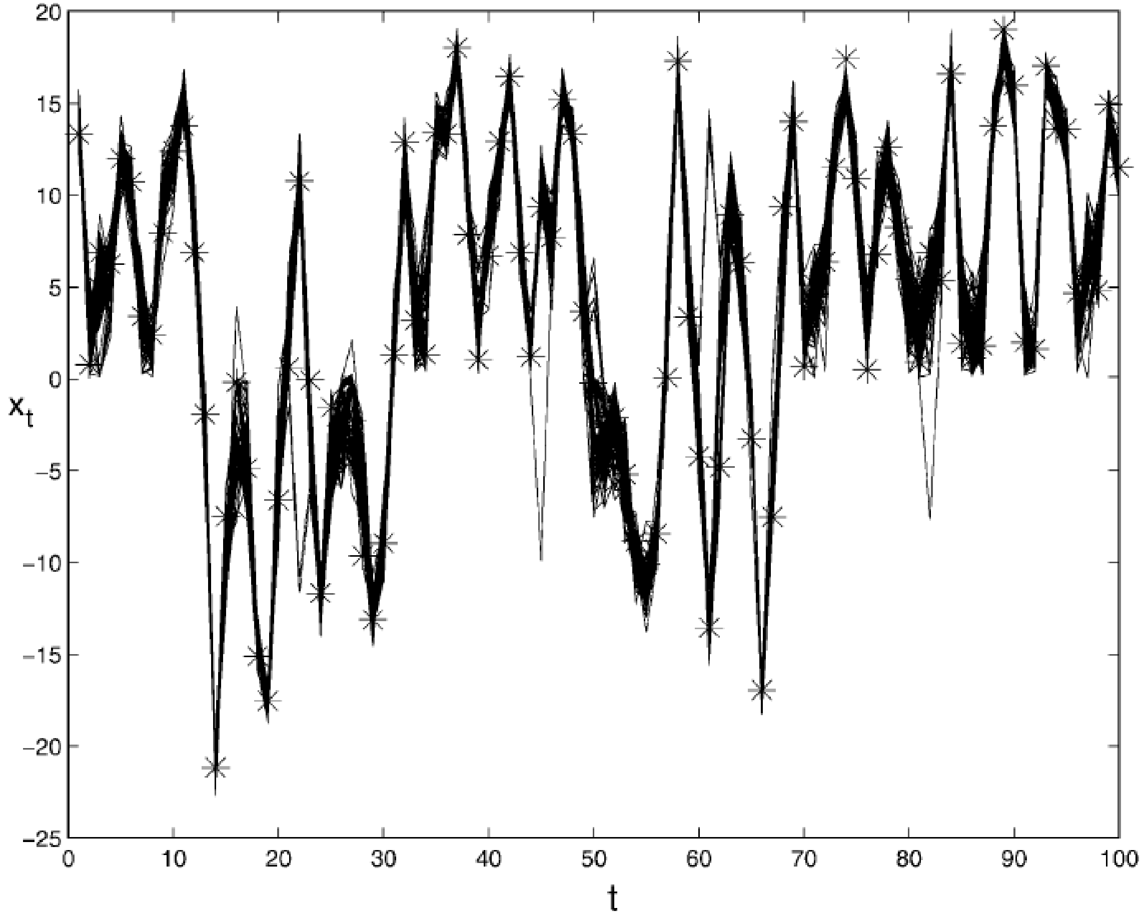
for $t = T - 1$ to 1 **do** ▷ Backward Pass Smoother

Calculate $\rho_t^{(i)} \propto \omega_t^{(i)} p(\tilde{x}_{t+1} \mid x_t^{(i)})$ for $i = 1, \ldots, L$ and normalize the modified weights.

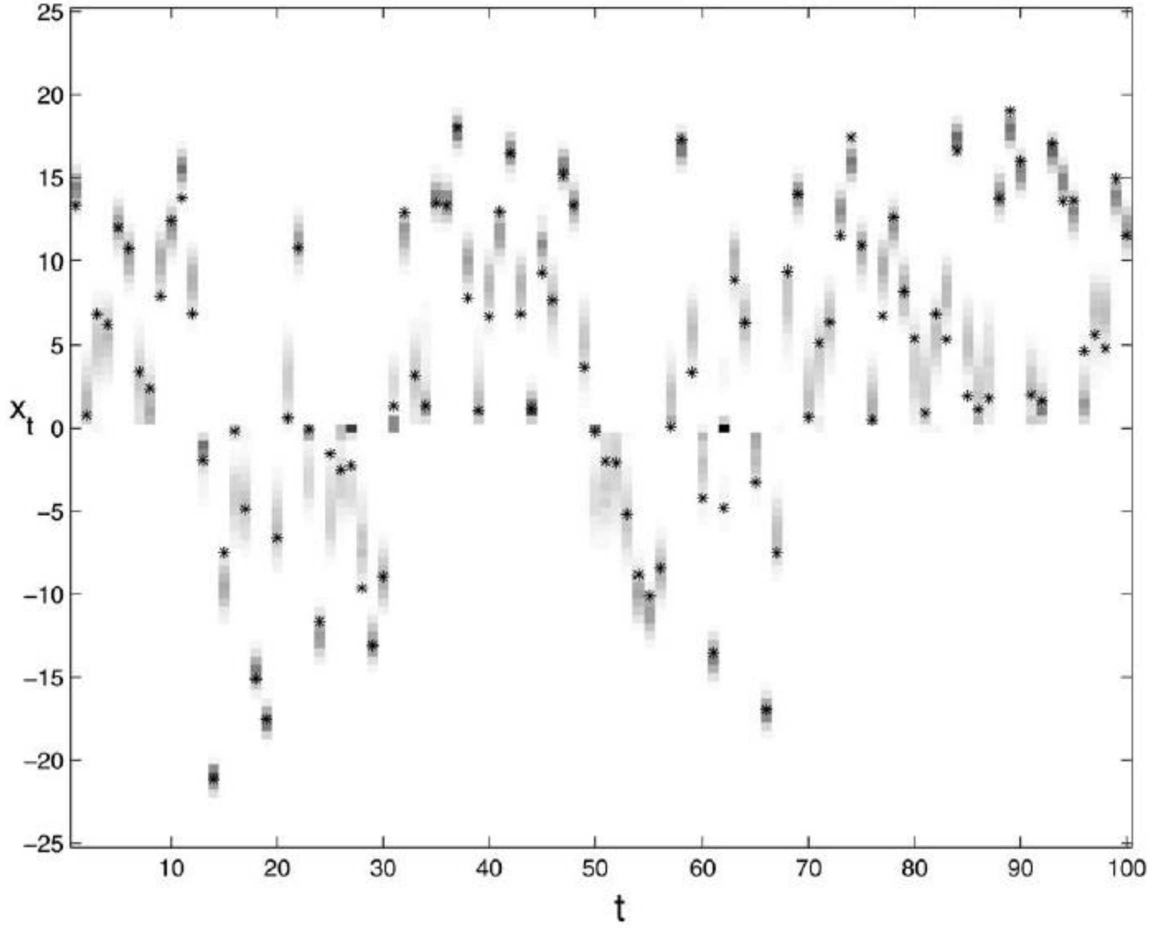
Choose $\widetilde{x}_t = x_t^{(i)}$ with probability $\rho_t^{(i)}$.

end for

Particle Smoothing Example



Smoothing trajectories for T=100. True states (*).



Kernel density estimates based on smoothed trajectories.True states (*).

Additional Particle Filter Topics

- Auxiliary particle filter – bias samples towards those more likely to "survive"

- Rao-Blackwell PF – analytically marginalize tractable sub-components of the state (e.g. linear Gaussian terms)

- MCMC PF – apply MC kernel with correct target $p(x_1^t \mid y_1^t)$ to sample trajectory prior to the resampling step

- Other smoothing topics:
  - Generalized two-filter smoothing
  - MC approximation of posterior marginals $p(x_t \mid y_1^T)$

- *Maximum a posteriori* (MAP) particle filter

- Maximum likelihood parameter estimation using PF