CMPE 58N - Lecture 8. Monte Carlo methods

Sequential Monte Carlo

Department of Computer Engineering,

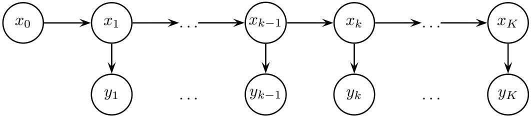
Boğaziçi University, Istanbul, Turkey

Instructor: A. Taylan Cemgil

Spring 2009

Time series models and Inference, Terminology

In signal processing, applied physics, machine learning many phenomena are modelled by dynamical models



$$\begin{aligned} x_k &\sim p(x_k|x_{k-1}) \\ y_k &\sim p(y_k|x_k) \end{aligned}$$

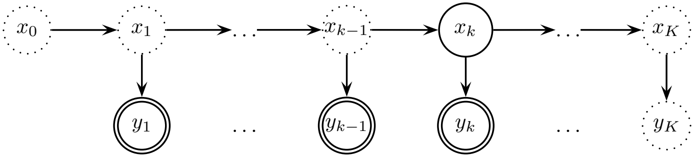
Transition Model

Observation Model

- x are the latent states
- y are the observations
- In a full Bayesian setting, x includes unknown model parameters

Online Inference, Terminology

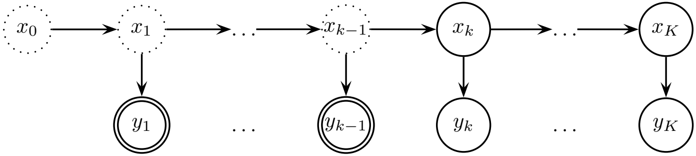
- **Filtering:** $p(x_k|y_{1:k})$
  - Distribution of current state given all past information
  - Realtime/Online/Sequential Processing



- Potentially confusing misnomer:
  - More general than “digital filtering” (convolution) in DSP – but algoritmically related for some models (KFM)

Online Inference, Terminology

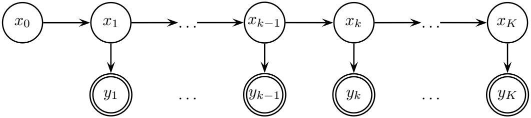
- **Prediction** $p(y_{k:K}, x_{k:K} | y_{1:k-1})$
  - evaluation of possible future outcomes; like filtering without observations



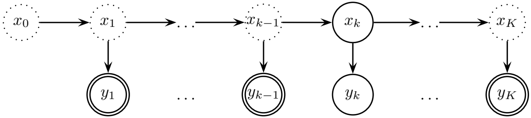
- Tracking, Restoration

Offline Inference, Terminology

- **Smoothing** $p(x_{0:K}|y_{1:K})$,
  Most likely trajectory – Viterbi path $\arg\max_{x_{0:K}} p(x_{0:K}|y_{1:K})$
  better estimate of past states, essential for learning



- **Interpolation** $p(y_k, x_k|y_{1:k-1}, y_{k+1:K})$
  fill in lost observations given past and future



Deterministic Linear Dynamical Systems

- The latent variables s_k and observations y_k are continuous
- The transition and observations models are linear
- Examples
  - A deterministic dynamical system with two state variables
  - Particle moving on the real line,

$$
\begin{aligned}
\mathbf{s}_k &= \begin{pmatrix} \text{phase} \\ \text{period} \end{pmatrix}_k = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \mathbf{s}_{k-1} = \mathbf{A}\mathbf{s}_{k-1} \\
y_k &= \text{phase}_k = \begin{pmatrix} 1 & 0 \end{pmatrix} \mathbf{s}_k = \mathbf{C}\mathbf{s}_k
\end{aligned}
$$

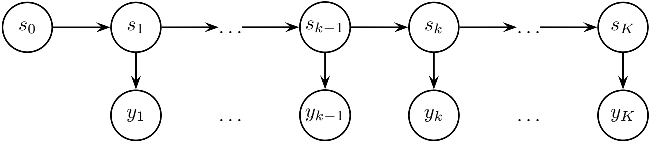
Kalman Filter Models, Stochastic Dynamical Systems

- We allow random (unknown) accelerations and observation error

$$
\begin{aligned}
\mathbf{s}_k &= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \mathbf{s}_{k-1} + \epsilon_k \\
&= \mathbf{A}\mathbf{s}_{k-1} + \epsilon_k
\end{aligned}
$$

$$
\begin{aligned}
y_k &= \begin{pmatrix} 1 & 0 \end{pmatrix} \mathbf{s}_k + \nu_k \\
&= \mathbf{C}\mathbf{s}_k + \nu_k
\end{aligned}
$$

Tracking

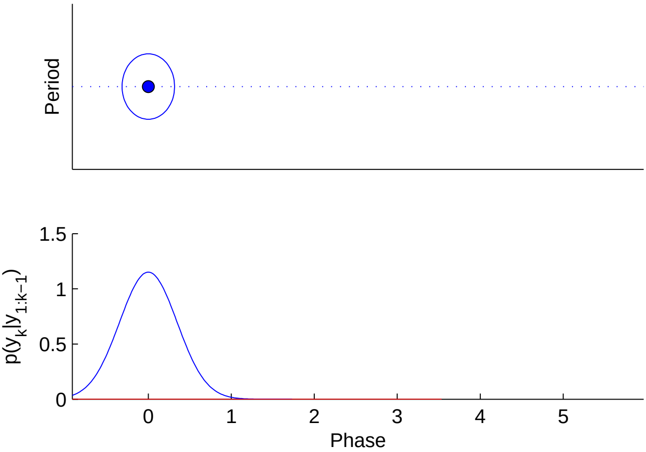


- In generative model notation

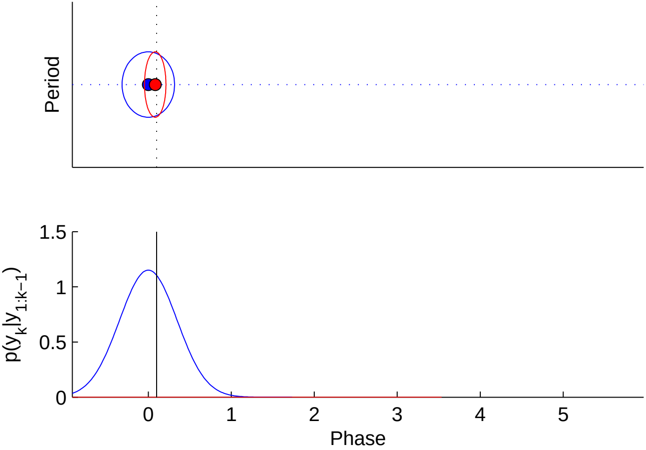
$$
\begin{aligned}
\mathbf{s}_k &\sim \mathcal{N}(\mathbf{s}_k; \mathbf{A}\mathbf{s}_{k-1}, Q) \\
y_k &\sim \mathcal{N}(y_k; \mathbf{C}\mathbf{s}_k, R)
\end{aligned}
$$

- Tracking = estimating the latent state of the system = Kalman filtering

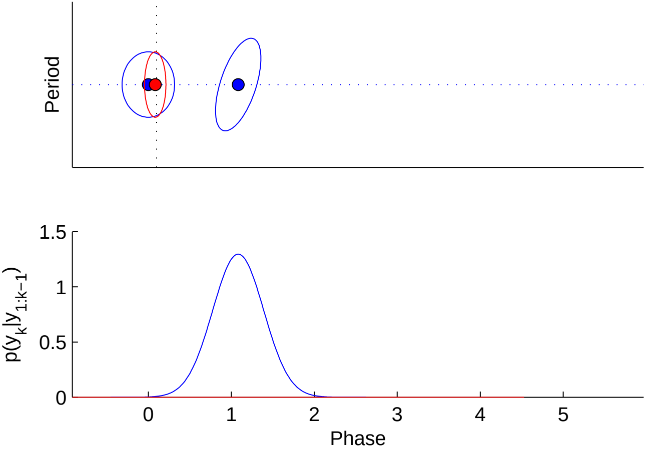
$$\alpha_{1|0} = p(x_1)$$



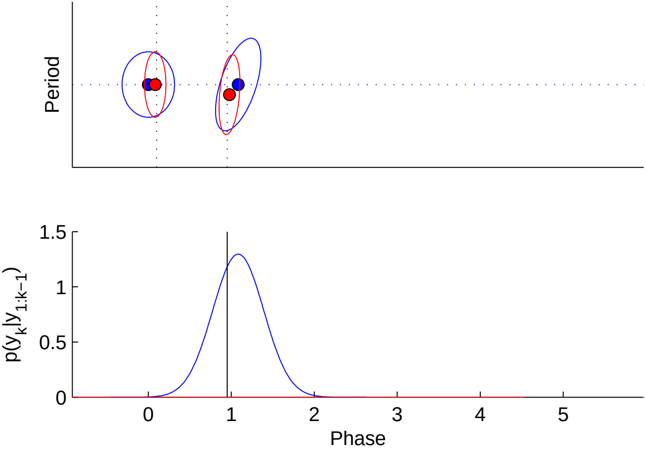
$$\alpha_{1|1} = p(y_1|x_1)p(x_1)$$



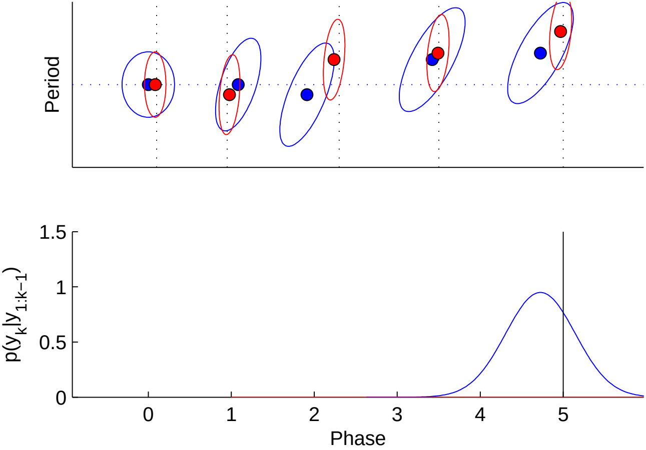
$$\alpha_{2|1} = \int dx_1 p(x_2|x_1)p(y_1|x_1)p(x_1) \propto p(x_2|y_1)$$



$$\alpha_{2|2} = p(y_2|x_2)p(x_2|y_1)$$



$$\alpha_{5|5} \propto p(x_5|y_{1:5})$$



Nonlinear/Non-Gaussian Dynamical Systems

$$
\begin{aligned}
x_k &\sim p(x_k|x_{k-1}) && \text{Transition Model} \\
y_k &\sim p(y_k|x_k) && \text{Observation Model}
\end{aligned}
$$

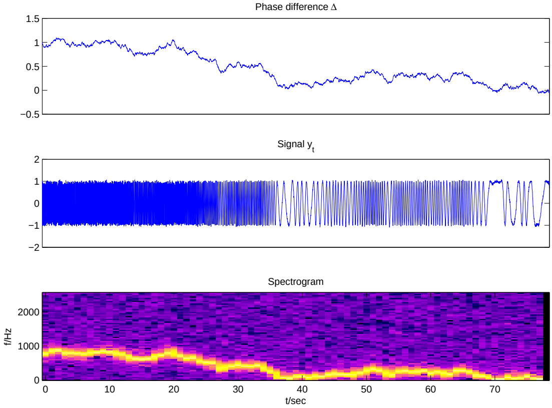
- What happens when the transition and/or observation model are non-Gaussian
- Apart from a handful of happy cases, the filtering density is not available in closed form or costs a lot of memory to represent exactly
  - $\Rightarrow$ Need efficient and flexible numeric integration techniques

Nonlinear Dynamical System Example

- Noisy Sinusoidal with frequency modulation

$$
\begin{aligned}
\Delta_k &\sim \mathcal{N}(\Delta_k; \Delta_{k-1}, Q) \\
\phi_k &= \phi_{k-1} + \Delta_k \\
y_k &\sim \mathcal{N}(y_k; \sin(\phi_k), R)
\end{aligned}
$$

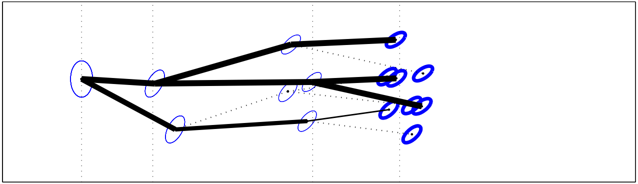
Example:



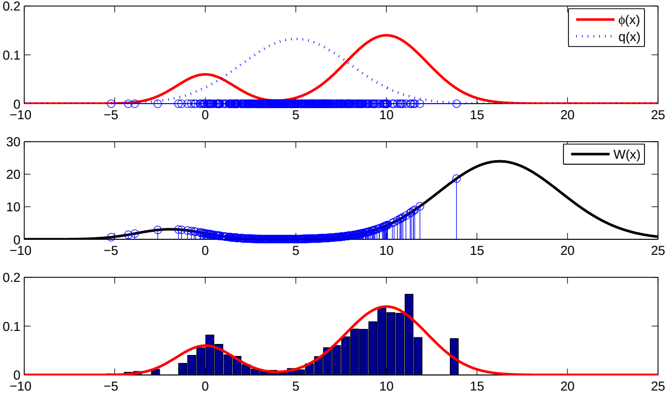
Phase difference Δ
1.5
1
0.5
0
−0.5
Signal y_t
2
1
0
−1
−2
Spectrogram
2000
1000
0
f/Hz
0
10
20
30
40
50
60
70
t/sec

Sequential Monte Carlo - Particle Filtering

- We try to approximate the so-called filtering density with a set of points/Gaussians $\equiv$ particles
- Algorithms are intuitively similar to randomised search algorithms but are best understood in terms of sequential importance sampling and resampling techniques



Importance Sampling (Review)



Resampling (Review)

- Importance sampling computes an approximation with weighted delta functions

$$p(x) \approx \sum_i \tilde{W}^{(i)} \delta(x - x^{(i)})$$

- In this representation, most of $\tilde{W}^{(i)}$ will be very close to zero and the representation may be dominated by few large weights.
- Resampling samples a set of new "particles"

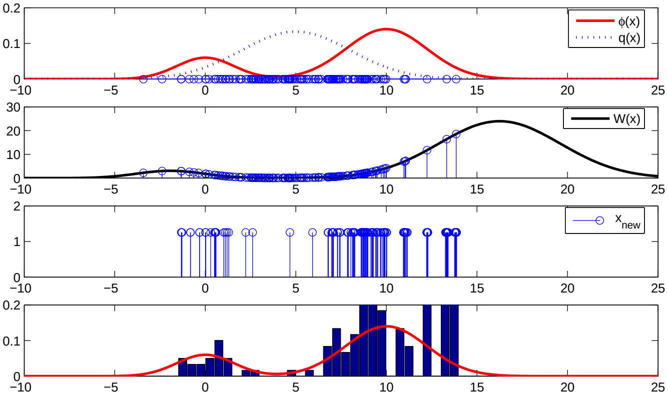
$$\begin{aligned} x_{\text{new}}^{(j)} &\sim \sum_i \tilde{W}^{(i)} \delta(x - x^{(i)}) \\ p(x) &\approx \frac{1}{N} \sum_j \delta(x - x_{\text{new}}^{(j)}) \end{aligned}$$

- Since we sample from a degenerate distribution, particle locations stay unchanged. We merely dublicate (, triplicate, ...) or discard particles according to their weight.

Resampling (Review) (cont.)

- This process is also named “selection”, “survival of the fittest”, e.t.c., in various fields (Genetic algorithms, AI..).

Resampling



$$x_{\text{new}}^{(j)} \sim \sum_i \tilde{W}^{(i)} \delta(x - x^{(i)})$$

Resampling

- Importance sampling computes an approximation with weighted delta functions

$$p(x) \approx \sum_i \tilde{w}^{(i)} \delta(x - x^{(i)})$$

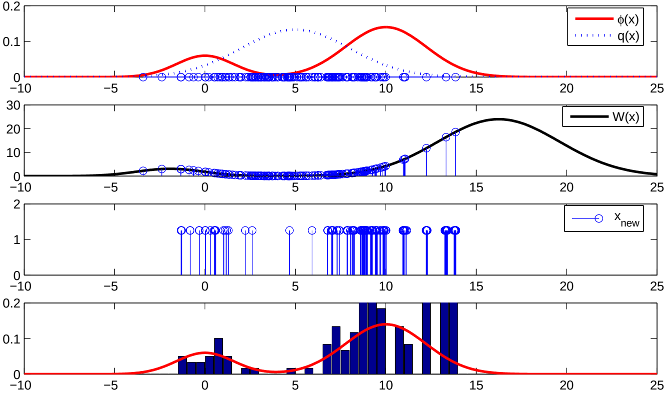
- In this representation, most of $\tilde{w}^{(i)}$ will be very close to zero and the representation may be dominated by few large weights.
- Resampling samples a set of new “particles”

$$\begin{aligned} x_{\text{new}}^{(j)} &\sim \sum_i \tilde{w}^{(i)} \delta(x - x^{(i)}) \\ p(x) &\approx \frac{1}{N} \sum_j \delta(x - x_{\text{new}}^{(j)}) \end{aligned}$$

Resampling (cont.)

- Since we sample from a degenerate distribution, particle locations stay unchanged. We merely duplicate (, triplicate, ...) or discard particles according to their weight.
- This process is also named "selection", "survival of the fittest", e.t.c., in various fields (Genetic algorithms, AI..).

Resampling



$$x_{\text{new}}^{(j)} \sim \sum_i \tilde{W}^{(i)} \delta(x - x^{(i)})$$

Popular Resampling Methods

- Multinomial Resampling
- Systematic Resampling
- Residual Resampling

Sequential Importance Sampling, Particle Filtering

Apply importance sampling to the SSM to obtain some samples from the posterior $p(x_{0:K}|y_{1:K})$.

$$p(x_{0:K}|y_{1:K}) = \frac{1}{p(y_{1:K})} p(y_{1:K}|x_{0:K}) p(x_{0:K}) \equiv \frac{1}{Z_y} \phi(x_{0:K}) \quad (1)$$

Key idea: sequential construction of the proposal distribution q, possibly using the available observations $y_{1:k}$, i.e.

$$q(x_{0:K}|y_{1:K}) = q(x_0) \prod_{k=1}^{K} q(x_k|x_{1:k-1} y_{1:k})$$

Sequential Importance Sampling

Due to the sequential nature of the model and the proposal, the importance weight function $W(x_{0:k}) \equiv W_k$ admits *recursive* computation

$$
\begin{aligned}
W_k &= \frac{\phi(x_{0:k})}{q(x_{0:k}|y_{1:k})} = \frac{p(y_k|x_k)p(x_k|x_{k-1})}{q(x_k|x_{0:k-1}y_{1:k})} \frac{\phi(x_{0:k-1})}{q(x_{0:k-1}|y_{1:k-1})} \qquad (2) \\
&= \frac{p(y_k|x_k)p(x_k|x_{k-1})}{q(x_k|x_{0:k-1}, y_{1:k})} W_{k-1} \equiv u_{k|0:k-1} W_{k-1} \qquad (3)
\end{aligned}
$$

Suppose we had an approximation to the posterior (in the sense $\langle f(x)\rangle_\phi \approx \sum_i W_{k-1}^{(i)} f(x_{0:k-1}^{(i)})$)

$$
\begin{aligned}
\phi(x_{0:k-1}) &\approx \sum_i W_{k-1}^{(i)} \delta(x_{0:k-1} - x_{0:k-1}^{(i)}) \\
x_k^{(i)} &\sim q(x_k|x_{0:k-1}^{(i)}, y_{1:k}) && \text{Extend trajectory} \\
W_k^{(i)} &= u_{k|0:k-1}^{(i)} W_{k-1} && \text{Update weight} \\
\phi(x_{0:k}) &\approx \sum_i W_k^{(i)} \delta(x_{0:k} - x_{0:k}^{(i)})
\end{aligned}
$$

Example

- Prior as the proposal density

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

- The weight is given by

$$\begin{aligned}
x_k^{(i)} &\sim p(x_k|x_{k-1}^{(i)}) && \text{Extend trajectory} \\
W_k^{(i)} &= u_{k|0:k-1}^{(i)} W_{k-1} && \text{Update weight} \\
&= \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{p(x_k^{(i)}|x_{k-1}^{(i)})} W_{k-1}^{(i)} = p(y_k|x_k^{(i)})W_{k-1}^{(i)}
\end{aligned}$$

- However, this schema will **not** work, since we blindly sample from the prior. But ...

Example (cont.)

- Perhaps surprisingly, interleaving importance sampling steps with (occasional) resampling steps makes the approach work quite well !!

$$
\begin{aligned}
x_k^{(i)} &\sim p(x_k | x_{k-1}^{(i)}) && \text{Extend trajectory} \\
W_k^{(i)} &= p(y_k | x_k^{(i)}) W_{k-1}^{(i)} && \text{Update weight} \\
\tilde{W}_k^{(i)} &= W_k^{(i)} / \tilde{Z}_k && \text{Normalize } (\tilde{Z}_k \equiv \textstyle\sum_{i'} W_k^{(i')}) \\
x_{0:k,\text{new}}^{(j)} &\sim \sum_{i=1}^{N} \tilde{W}^{(i)} \delta(x_{0:k} - x_{0:k}^{(i)}) && \text{Resample } j = 1 \dots N
\end{aligned}
$$

- This results in a new representation as

$$
x_{0:k}^{(i)} \leftarrow x_{0:k,\text{new}}^{(j)} \qquad W_k^{(i)} \leftarrow \tilde{Z}_k / N
$$

$$
\phi(x) \approx \frac{1}{N} \sum_j \tilde{Z}_k \delta(x_{0:k} - x_{0:k,\text{new}}^{(j)})
$$

Optimal proposal distribution

- The algorithm in the previous example is known as *Bootstrap particle filter* or *Sequential Importance Sampling/Resampling* (SIS/SIR).
- Can we come up with a better proposal in a sequential setting?
  - We are not allowed to move previous sampling points $x_{1:k-1}^{(i)}$ (because in many applications we can't even store them)
  - Better in the sense of minimizing the variance of weight function $W_k(x)$.
- The answer turns out to be the filtering distribution

$$q(x_k|x_{1:k-1}, y_{1:k}) = p(x_k|x_{k-1}, y_k) \tag{4}$$

Optimal proposal distribution (cont.)

- The weight is given by

$$
\begin{aligned}
x_k^{(i)} &\sim p(x_k|x_{k-1}^{(i)}, y_k) && \text{Extend trajectory} \\
W_k^{(i)} &= u_{k|0:k-1}^{(i)} W_{k-1}^{(i)} && \text{Update weight} \\
u_{k|0:k-1}^{(i)} &= \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{p(x_k^{(i)}|x_{k-1}^{(i)}, y_k)} \times \frac{p(y_k|x_{k-1}^{(i)})}{p(y_k|x_{k-1}^{(i)})} \\
&= \frac{p(y_k, x_k^{(i)}|x_{k-1}^{(i)})p(y_k|x_{k-1}^{(i)})}{p(x_k^{(i)}, y_k|x_{k-1}^{(i)})} = p(y_k|x_{k-1}^{(i)})
\end{aligned}
$$

A Generic Particle Filter

1. **Generation**:
   Compute the proposal distribution $q(x_k|x_{0:k-1}^{(i)}, y_{1:k})$.
   Generate offsprings for $i = 1 \dots N$

$$\hat{x}_k^{(i)} \sim q(x_k|x_{0:k-1}^{(i)}, y_{1:k})$$

2. **Evaluate** importance weights

$$W_k^{(i)} = \frac{p(y_k|\hat{x}_k^{(i)})p(\hat{x}_k^{(i)}|x_{k-1}^{(i)})}{q(\hat{x}_k^{(i)}|x_{0:k-1}^{(i)}, y_{1:k})} W_{k-1}^{(i)} \qquad x_{0:k}^{(i)} = (\hat{x}_k^{(i)}, x_{0:k-1}^{(i)})$$

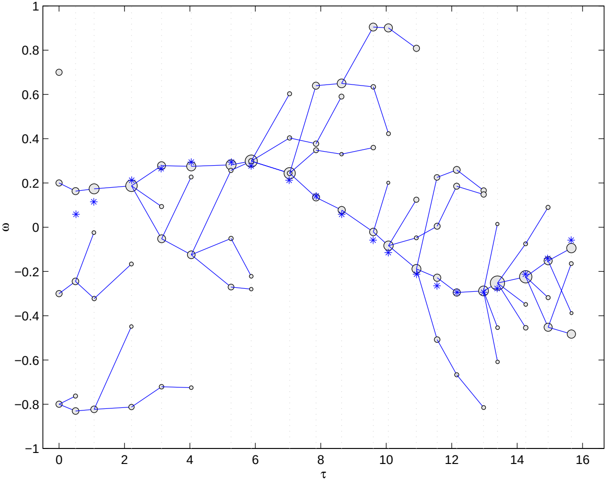
3. **Resampling** (optional but recommended)

Normalize weigts $\quad \tilde{W}_k^{(i)} = W_k^{(i)}/\tilde{Z}_k \qquad \tilde{Z}_k \equiv \sum_j W_k^{(j)}$

Resample $\quad x_{0:k,\text{new}}^{(j)} \sim \sum_{i=1}^{N} \tilde{W}^{(i)} \delta(x_{0:k} - x_{0:k}^{(i)}) \qquad j = 1 \dots N$

Reset $\quad x_{0:k}^{(i)} \leftarrow x_{0:k,\text{new}}^{(j)} \qquad W_k^{(i)} \leftarrow \tilde{Z}_k/N$

Particle Filtering



Summary

- Time Series Models and Inference
- Importance Sampling, Resampling Review
- Putting it all together, Sequential Monte Carlo