Time series models, Importance sampling and Sequential Monte Carlo

A. Taylan Cemgil

Signal Processing and Communications Lab.

UNIVERSITY OF CAMBRIDGE
Department of Engineering

08 March 2007
Outline

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

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

\[ x_0, x_1, \ldots, x_{k-1}, x_k, \ldots, x_K \]

\[ y_1, \ldots, y_{k-1}, y_k, \ldots, y_K \]

\[ x_k \sim p(x_k|x_{k-1}) \quad \text{Transition Model} \]

\[ y_k \sim p(y_k|x_k) \quad \text{Observation Model} \]

- \( x \) 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

```
x_0 \rightarrow x_1 \rightarrow \ldots \rightarrow x_{k-1} \rightarrow x_k \rightarrow \ldots \rightarrow x_K
```

```
y_1 \downarrow \ldots \downarrow (y_{k-1}) \downarrow y_k \downarrow \ldots \downarrow y_K
```

- Potentially confusing misnomer:
  - More general than “digital filtering” (convolution) in DSP – but algorithmically 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,

\[
\begin{align*}
s_k &= \begin{pmatrix} \text{phase} \\ \text{period} \end{pmatrix}_k = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} s_{k-1} = As_{k-1} \\
y_k &= \text{phase}_k = \begin{pmatrix} 1 & 0 \end{pmatrix} s_k = Cs_k
\end{align*}
\]
Kalman Filter Models, Stochastic Dynamical Systems

- We allow random (unknown) accelerations and observation error

\[
\begin{align*}
\mathbf{s}_k &= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \mathbf{s}_{k-1} + \epsilon_k \\
&= \mathbf{A} \mathbf{s}_{k-1} + \epsilon_k \\
\mathbf{y}_k &= \begin{pmatrix} 1 & 0 \end{pmatrix} \mathbf{s}_k + \nu_k \\
&= \mathbf{C} \mathbf{s}_k + \nu_k
\end{align*}
\]
• In generative model notation

\[ s_k \sim \mathcal{N}(s_k; As_{k-1}, Q) \]
\[ y_k \sim \mathcal{N}(y_k; Cs_k, R) \]

• Tracking = estimating the latent state of the system = Kalman filtering
\[ \alpha_{1|0} = p(x_1) \]
$$\alpha_{1|1} = p(y_{1|x_1})p(x_1)$$
\[ \alpha_{2|1} = \int dx_1 p(x_2|x_1)p(y_1|x_1)p(x_1) \propto p(x_2|y_1) \]
\[ \alpha_{2|2} = p(y_2|x_2)p(x_2|y_1) \]
$$\alpha_{5|5} \propto p(x_5|y_{1:5})$$

![Diagram showing phase and period with probability distributions and points.](image-url)
Nonlinear/Non-Gaussian Dynamical Systems

\[ x_k \sim p(x_k|x_{k-1}) \quad \text{Transition Model} \]
\[ y_k \sim p(y_k|x_k) \quad \text{Observation Model} \]

- 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

⇒ Need efficient and flexible numeric integration techniques
Nonlinear Dynamical System Example

- Noisy Sinusoidal with frequency modulation

\[
\begin{align*}
\Delta_k & \sim \mathcal{N}(\Delta_k; \Delta_{k-1}, Q) \\
\phi_k & = \phi_{k-1} + \Delta_k \\
y_k & \sim \mathcal{N}(y_k; \sin(\phi_k), R)
\end{align*}
\]
Example:

Phase difference $\Delta$

Signal $y_t$

Spectrogram
Dynamical Systems with switching

- Complicated processes can be modeled by using simple processes with occasional regime switches
  - Piecewise constant
Segmentation and Changepoint detection

- Piecewise linear

- Used for tracking, segmentation, changepoint detection ...
  - What is the true state of the process given noisy data ?
  - Where are the changepoints ?
  - How many changepoints ?
Example: Conditionally Gaussian Changepoint Model

\[ r_k \sim p(r_k | r_{k-1}) \]

\[ \theta_k \sim [r_k = \text{reg}] f(\theta_k | \theta_{k-1}) + [r_k = \text{new}] \pi(\theta_k) \]

\[ y_k \sim p(y_k | \theta_k) \]

Changepoint flags \( \in \{\text{new, reg}\} \)

Latent State

Observations
Example: Piecewise constant signal

\[ \theta_0 \sim \mathcal{N}(\mu, P) \]

\[ r_k | r_{k-1} \sim p(r_k | r_{k-1}) \]

\[ \theta_k | \theta_{k-1}, r_k \sim \left[ r_k = 0 \right] \delta(\theta_k - \theta_{k-1}) + \left[ r_k = 1 \right] \mathcal{N}(m, V) \]

\[ y_k | \theta_k \sim \mathcal{N}(\theta_k, R) \]
Switching State space model

\[ r_k \sim p(r_k | r_{k-1}) \]  \hspace{1cm} \text{Regime label}

\[ \theta_k \sim \mathcal{N}(\theta_k; A r_k \theta_{k-1}, Q r_k) \]

\[ y_k \sim \mathcal{N}(y_k; C \theta_k, R) \]  \hspace{1cm} \text{Observations}
Exact Inference in switching state space models

• In general, exact inference is intractable (NP hard)
  – Conditional Gaussians are not closed under marginalization

  ⇒ Unlike HMM’s or KFM’s, summing over $r_k$ does not simplify the filtering density
  ⇒ Number of Gaussian kernels to represent exact filtering density $p(r_k, \theta_k | y_{1:k})$ increases exponentially
Sequential Monte Carlo - Particle Filtering

- We try to approximate the so-called filtering density with a set of points/Gaussians ≡ particles

- Algorithms are intuitively similar to randomised search algorithms but are best understood in terms of sequential importance sampling and resampling techniques
Importance Sampling
Importance Sampling (IS)

Consider a probability distribution with (possibly unknown) normalisation constant

\[ p(x) = \frac{1}{Z} \phi(x) \quad \text{where} \quad Z = \int dx \phi(x). \]

IS: Estimate expectations (or features) of \( p(x) \) by a weighted sample

\[ \langle f(x) \rangle_{p(x)} = \int dx f(x) p(x) \]

\[ \langle f(x) \rangle_{p(x)} \approx \sum_{i=1}^{N} \tilde{w}^{(i)} f(x^{(i)}) \]
Importance Sampling (cont.)

- Change of measure with weight function $W(x) \equiv \phi(x)/q(x)$

$$\langle f(x) \rangle_{p(x)} = \frac{1}{Z} \int dx f(x) \frac{\phi(x)}{q(x)} q(x) = \frac{1}{Z} \left\langle f(x) \frac{\phi(x)}{q(x)} \right\rangle_{q(x)} \equiv \frac{1}{Z} \langle f(x)W(x) \rangle_{q(x)}$$

- If $Z$ is unknown, as is often the case in Bayesian inference

$$Z = \int dx \phi(x) = \int dx \frac{\phi(x)}{q(x)} q(x) = \langle W(x) \rangle_{q(x)}$$

$$\langle f(x) \rangle_{p(x)} = \frac{\langle f(x)W(x) \rangle_{q(x)}}{\langle W(x) \rangle_{q(x)}}$$
Importance Sampling (cont.)

- Draw $i = 1, \ldots, N$ independent samples from $q$
  \[ x^{(i)} \sim q(x) \]

- We calculate the **importance weights**
  \[ W^{(i)} = W(x^{(i)}) = \phi(x^{(i)}) / q(x^{(i)}) \]

- Approximate the normalizing constant
  \[ Z = \langle W(x) \rangle_{q(x)} \approx \sum_{i=1}^{N} W^{(i)} \]

- Desired expectation is approximated by
  \[ \langle f(x) \rangle_{p(x)} = \frac{\langle f(x) W(x) \rangle_{q(x)}}{\langle W(x) \rangle_{q(x)}} \approx \frac{\sum_{i=1}^{N} W^{(i)} f(x^{(i)})}{\sum_{i=1}^{N} W^{(i)}} \equiv \sum_{i=1}^{N} \tilde{w}^{(i)} f(x^{(i)}) \]

Here $\tilde{w}^{(i)} = W^{(i)} / \sum_{j=1}^{N} W^{(j)}$ are **normalized importance weights**.
Importance Sampling (cont.)

\begin{align*}
\phi(x) & \quad q(x) \\
W(x) & \quad \text{(Graphs)}
\end{align*}
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”

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

\[
p(x) \approx \frac{1}{N} \sum_j \delta(x - x_{\text{new}}^{(j)})
\]

- 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

\[ x_{\text{new}}^{(j)} \sim \sum_i \tilde{W}^{(i)} \delta(x - x^{(i)}) \]
Examples of Proposal Distributions

\[ p(x|y) \propto p(y|x)p(x) \]

- Prior as the proposal. \( q(x) = p(x) \)

\[ W(x) = \frac{p(y|x)p(x)}{p(x)} = p(y|x) \]
Examples of Proposal Distributions

\[ p(x|y) \propto p(y|x)p(x) \]

- Likelihood as the proposal. \( q(x) = \frac{p(y|x)}{\int dx p(y|x)} = \frac{p(y|x)}{c(y)} \)

\[ W(x) = \frac{p(y|x)p(x)}{p(y|x)/c(y)} = \frac{p(x)c(y)}{c(y)} \propto p(x) \]

- Interesting when sensors are very accurate and \( \text{dim}(y) \gg \text{dim}(x) \).

Since there are many proposals, is there a “best” proposal distribution?
Optimal Proposal Distribution

\[ p(x|y) \propto p(y|x)p(x) \]

Task: Estimate \( \langle f(x) \rangle_{p(x|y)} \)

- IS constructs the estimator \( I(f) = \langle f(x)W(x) \rangle_{q(x)} \)

- Minimize the variance of the estimator

\[
\left\langle \left( f(x)W(x) - \langle f(x)W(x) \rangle \right)^2 \right\rangle_{q(x)} = \langle f^2(x)W^2(x) \rangle_{q(x)} - \langle f(x)W(x) \rangle_{q(x)}^2 \tag{1}
\]
\[
= \langle f^2(x)W^2(x) \rangle_{q(x)} - \langle f(x) \rangle_{p(x)}^2 \tag{2}
\]
\[
= \langle f^2(x)W^2(x) \rangle_{q(x)} - I^2(f) \tag{3}
\]

- Minimize the first term since only it depends upon \( q \)
Optimal Proposal Distribution

• (By Jensen’s inequality) The first term is lower bounded:

\[
\langle f^2(x)W^2(x) \rangle_{q(x)} \geq \langle |f(x)|W(x) \rangle_{q(x)}^2 = \left( \int |f(x)| p(x|y) dx \right)^2
\]

• We will look for a distribution \( q^* \) that attains this lower bound. Take

\[
q^*(x) = \frac{|f(x)|p(x|y)}{\int |f(x')|p(x'|y) dx'}
\]
Optimal Proposal Distribution (cont.)

- The weight function for this particular proposal $q^*$ is

$$W_*(x) = p(x|y)/q^*(x) = \frac{\int |f(x')|p(x'|y)dx'}{|f(x)|}$$

- We show that $q^*$ attains its lower bound

$$\langle f^2(x)W^2_*(x) \rangle_{q^*(x)} = \langle f^2(x) \frac{\left(\int |f(x')|p(x'|y)dx'\right)^2}{|f(x)|^2} \rangle_{q^*(x)}$$

$$= \left(\int |f(x')|p(x'|y)dx'\right)^2 = \langle |f(x)| \rangle_{p(x|y)}^2$$

$$= \langle |f(x)|W_*(x) \rangle_{q^*(x)}^2$$

- $\Rightarrow$ There are distributions $q^*$ that are even “better” than the exact posterior!
A link to alpha divergences

The $\alpha$-divergence between two distributions is defined as

$$D_\alpha(p||q) \equiv \frac{1}{\beta(1 - \beta)} \left( 1 - \int dx p(x)^{\beta} q(x)^{1 - \beta} \right)$$

where $\beta = (1 + \alpha)/2$ and $p$ and $q$ are two probability distributions

- $\lim_{\beta \to 0} D_\alpha(p||q) = KL(q||p)$
- $\lim_{\beta \to 1} D_\alpha(p||q) = KL(p||q)$
- $\beta = 2, (\alpha = 3)$

$$D_3(p||q) \equiv \frac{1}{2} \int dx p(x)^2 q(x)^{-1} - \frac{1}{2} = \frac{1}{2} \langle W(x)^2 \rangle_q(x) - \frac{1}{2}$$

Best $q$ (in a constrained family) is typically a heavy-tailed approximation to $p$
Examples of Proposal Distributions

\[ p(x|y) \propto p(y_1|x_1)p(x_1)p(y_2|x_2)p(x_2|x_1) \]

Task: Obtain samples from the posterior \( p(x_{1:2}|y_{1:2}) = \frac{1}{Z_y} \phi(x_{1:2}) \)

- Prior as the proposal. \( q(x_{1:2}) = p(x_1)p(x_2|x_1) \)

\[
W(x_{1:2}) = \frac{\phi(x_{1:2})}{q(x_{1:2})} = p(y_1|x_1)p(y_2|x_2)
\]

- We sample from the prior as follows:

\[
x^{(i)}_1 \sim p(x_1) \quad x^{(i)}_2 \sim p(x_2|x_1 = x^{(i)}_1) \quad W(x^{(i)}) = p(y_1|x^{(i)}_1)p(y_2|x^{(i)}_2)
\]
Examples of Proposal Distributions

\[ \phi(x_{1:2}) = p(y_1 | x_1)p(x_1)p(y_2 | x_2)p(x_2 | x_1) \]

- State prediction as the proposal. 
  \[ q(x_{1:2}) = p(x_1 | y_1)p(x_2 | x_1) \]

  \[ W(x_{1:2}) = \frac{p(y_1 | x_1)p(x_1)p(y_2 | x_2)p(x_2 | x_1)}{p(x_1 | y_1)p(x_2 | x_1)} = p(y_1)p(y_2 | x_2) \]

- We sample from the proposal and compute the weight

  \[ x_1^{(i)} \sim p(x_1 | y_1) \quad x_2^{(i)} \sim p(x_2 | x_1 = x_1^{(i)}) \quad W(x^{(i)}) = p(y_1)p(y_2 | x_2^{(i)}) \]

- Note that this weight does not depend on \( x_1 \)
Examples of Proposal Distributions

\[ \phi(x_{1:2}) = p(y_1|x_1)p(x_1)p(y_2|x_2)p(x_2|x_1) \]

- Filtering distribution as the proposal. \( q(x_{1:2}) = p(x_1|y_1)p(x_2|x_1, y_2) \)

\[ W(x_{1:2}) = \frac{p(y_1|x_1)p(x_1)p(y_2|x_2)p(x_2|x_1)}{p(x_1|y_1)p(x_2|x_1, y_2)} = p(y_1)p(y_2|x_1) \]

- We sample from the proposal and compute the weight

\[ x_1^{(i)} \sim p(x_1|y_1) \quad x_2^{(i)} \sim p(x_2|x_1 = x_1^{(i)}, y_2) \quad W(x^{(i)}) = p(y_1)p(y_2|x_1^{(i)}) \]

- Note that this weight does not depend on \( x_2 \)
Examples of Proposal Distributions

\[ \phi(x_{1:2}) = p(y_1|x_1)p(x_1)p(y_2|x_2)p(x_2|x_1) \]

- Exact posterior as the proposal. 
  \[ q(x_{1:2}) = p(x_1|y_1, y_2)p(x_2|x_1, y_2) \]

\[ W(x_{1:2}) = \frac{p(y_1|x_1)p(x_1)p(y_2|x_2)p(x_2|x_1)}{p(x_1|y_1)p(x_2|x_1, y_2)} = p(y_1)p(y_2|y_1) \]

- Note that this weight is constant, i.e. 
  \[ \langle W(x_{1:2})^2 \rangle - \langle W(x_{1:2}) \rangle^2 = 0 \]
Variance reduction

\[ q(x) \quad W(x) = \frac{\phi(x)}{q(x)} \]

\[
\begin{align*}
p(x_1)p(x_2|x_1) & \quad p(y_1|x_1)p(y_2|x_2) \\
p(x_1|y_1)p(x_2|x_1) & \quad p(y_1)p(y_2|x_2) \\
p(x_1|y_1)p(x_2|x_1, y_2) & \quad p(y_1)p(y_2|x_1) \\
p(x_1|y_1, y_2)p(x_2|x_1, y_2) & \quad p(y_1)p(y_2|y_1)
\end{align*}
\]

Accurate proposals

- gradually decrease the variance
- but take more time to compute
Sequential Importance Sampling, Particle Filtering

Apply importance sampling to the SSM to obtain some samples from the posterior $p(x_0:K|y_{1:K})$.

$$p(x_0:K|y_{1:K}) = \frac{1}{p(y_{1:K})} p(y_{1:K}|x_{0:K}) p(x_0:K) \equiv \frac{1}{Z_y} \phi(x_{0:K}) \tag{4}$$

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{align*}
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}) q(x_{0:k-1} | y_{1:k-1})} \frac{\phi(x_{0:k-1})}{u_{k|0:k-1} W_{k-1}} \\
&= \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}
\end{align*}

\begin{align*}
\phi(x_{0:k-1}) &\approx \sum_i W^{(i)}_{k-1} \delta(x_{0:k-1} - x^{(i)}_{0:k-1}) \\
x^{(i)}_k &\sim q(x_k | x^{(i)}_{0:k-1}, y_{1:k}) \quad \text{Extend trajectory} \\
W^{(i)}_k &= u^{(i)}_{k|0:k-1} W_{k-1} \quad \text{Update weight} \\
\phi(x_{0:k}) &\approx \sum_i W^{(i)}_k \delta(x_{0:k} - x^{(i)}_{0:k})
\end{align*}
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

\[
\begin{align*}
    x_k^{(i)} & \sim p(x_k|x^{(i)}_{k-1}) & \text{Extend trajectory} \\
    W_k^{(i)} & = u_k^{(i)} W_{k-1} & \text{Update weight} \\
    & = \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x^{(i)}_{k-1})}{p(x_k^{(i)}|x^{(i)}_{k-1})} W_{k-1} = p(y_k|x_k^{(i)}) W_{k-1}
\end{align*}
\]

- 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!!

\[
x_k^{(i)} \sim p(x_k|x_{k-1}^{(i)})
\]

\[
W_k^{(i)} = p(y_k|x_k^{(i)})W_{k-1}^{(i)}
\]

\[
\tilde{W}_k^{(i)} = W_k^{(i)}/\tilde{Z}_k
\]

\[
x_{0:k,\text{new}}^{(j)} \sim \sum_{i=1}^{N} \tilde{W}_k^{(i)}\delta(x_{0:k} - x_{0:k}^{(i)})
\]

- This results in a new representation as

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

\[
x_{0:k}^{(i)} \leftarrow x_{0:k,\text{new}}^{(j)}
\]

\[
W_k^{(i)} \leftarrow \tilde{Z}_k/N
\]
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 \( W_k(x) \).
    (remember the optimality story in Eq.(3) and set \( f(x) = 1 \)).

- The answer turns out to be the filtering distribution

  \[
  q(x_k|x_{1:k-1}, y_{1:k}) = p(x_k|x_{k-1}, y_k)
  \]  

  (7)
Optimal proposal distribution (cont.)

- The weight is given by

\[
x^{(i)}_k \sim p(x_k|x^{(i)}_{k-1}, y_k)
\]

\[
W^{(i)}_k = u^{(i)}_{k|0:k-1} W^{(i)}_{k-1}
\]

\[
u^{(i)}_{k|0:k-1} = \frac{p(y_k|x^{(i)}_k) p(x^{(i)}_k|x^{(i)}_{k-1})}{p(x^{(i)}_k|x^{(i)}_{k-1}, y_k)} \times \frac{p(y_k|x^{(i)}_{k-1})}{p(y_k|x^{(i)}_{k-1})}
\]

\[
= \frac{p(y_k, x^{(i)}_k|x^{(i)}_{k-1}) p(y_k|x^{(i)}_{k-1})}{p(x^{(i)}_k, y_k|x^{(i)}_{k-1})} = p(y_k|x^{(i)}_{k-1})
\]

Extend trajectory

Update weight
A Generic Particle Filter

1. **Generation:**
   Compute the proposal distribution $q(x_k|x_{0:k-1}, y_{1:k})$.
   Generate offsprings for $i = 1 \ldots N$
   \[ \hat{x}_k^{(i)} \sim q(x_k|x_{0:k-1}, y_{1:k}) \]

2. **Evaluate** importance weights
   \[ W_k^{(i)} = \frac{p(y_k|\hat{x}_k^{(i)})p(\hat{x}_k^{(i)}|x_{k-1}^{(i)})}{q(\hat{x}_k^{(i)}|x_{0:k-1}^{(i)}, y_{1:k})} W_{k-1}^{(i)} \]
   \[ x_0:k = (\hat{x}_k^{(i)}, x_{0:k-1}^{(i)}) \]

3. **Resampling** (optional but recommended)
   Normalize weights
   \[ \tilde{W}_k^{(i)} = \frac{W_k^{(i)}}{\tilde{Z}_k} \]
   \[ \tilde{Z}_k \equiv \sum_j W_k^{(j)} \]
   Resample
   \[ x_{0:k,new} \sim \sum_{i=1}^N \tilde{W}_k^{(i)} \delta(x_{0:k} - x_0:k^{(i)}) \]
   \[ j = 1 \ldots N \]
   Reset
   \[ x_0:k \leftarrow x_{0:k,new} \]
   \[ W_k^{(i)} \leftarrow \tilde{Z}_k/N \]
Summary

- Time Series Models and Inference
  - Nonlinear Dynamical systems
  - Conditionally Gaussian Switching State Space Models
  - Change-point models

- Importance Sampling, Resampling

- Putting it all together, Sequential Monte Carlo
The End

Slides are online