An introduction to particle filters

SMC Down Under 2023

Imke Botha

Queensland University of Technology

Outline

1. Hidden Markov models (HMMs)
2. The filtering problem
3. Particle filtering
4. Beyond the basics

Hidden Markov Models

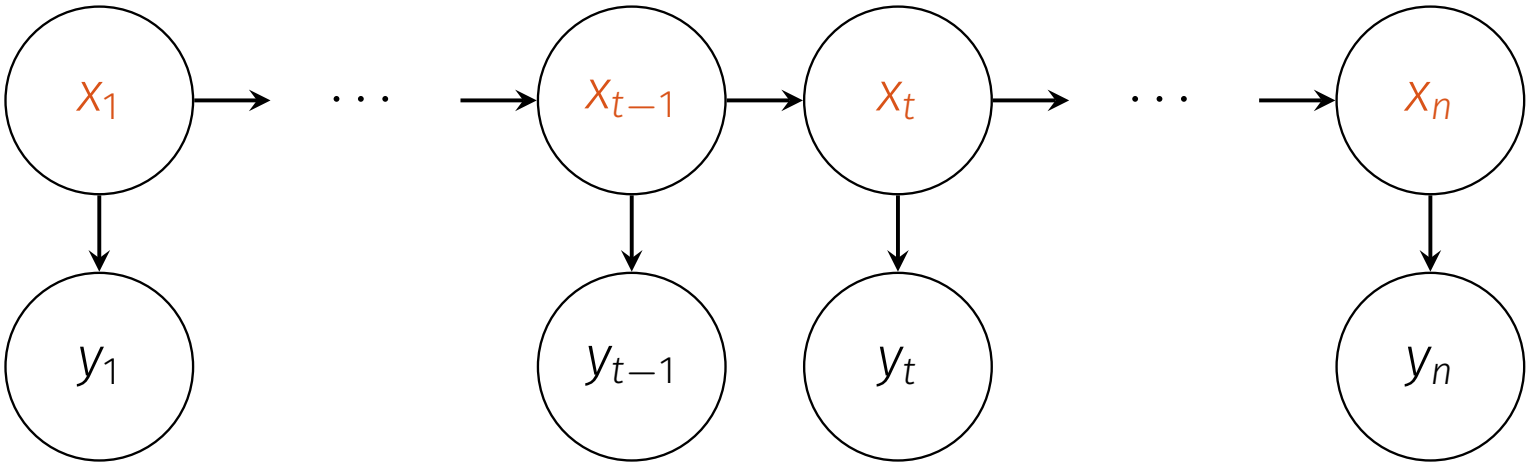
Hidden Markov Models

We have two stochastic processes:

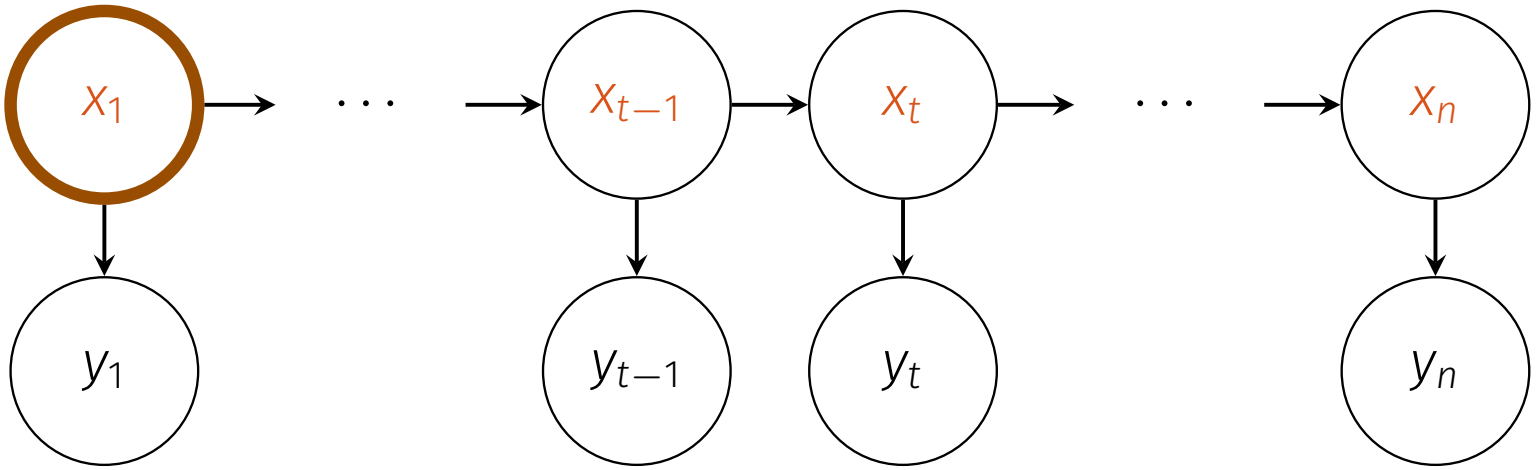
1. Latent or unobserved process $x_t \in \mathsf{X} = \mathbb{R}^{d_\mathsf{X}}$
2. Observed process $y_t \in \mathsf{Y} = \mathbb{R}^{d_\mathsf{Y}}$

for $t \in \{1, \ldots, n\}$.

Hidden Markov Models (HMM)

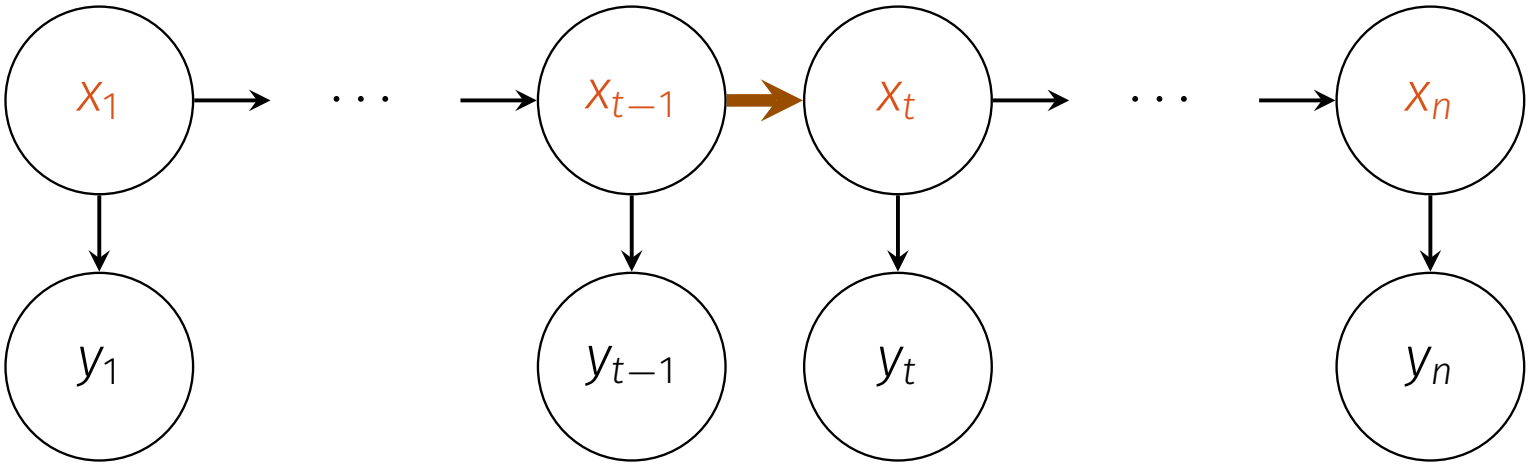


Hidden Markov Models (HMM)



$X_1 \sim p(x_1)$ initial density

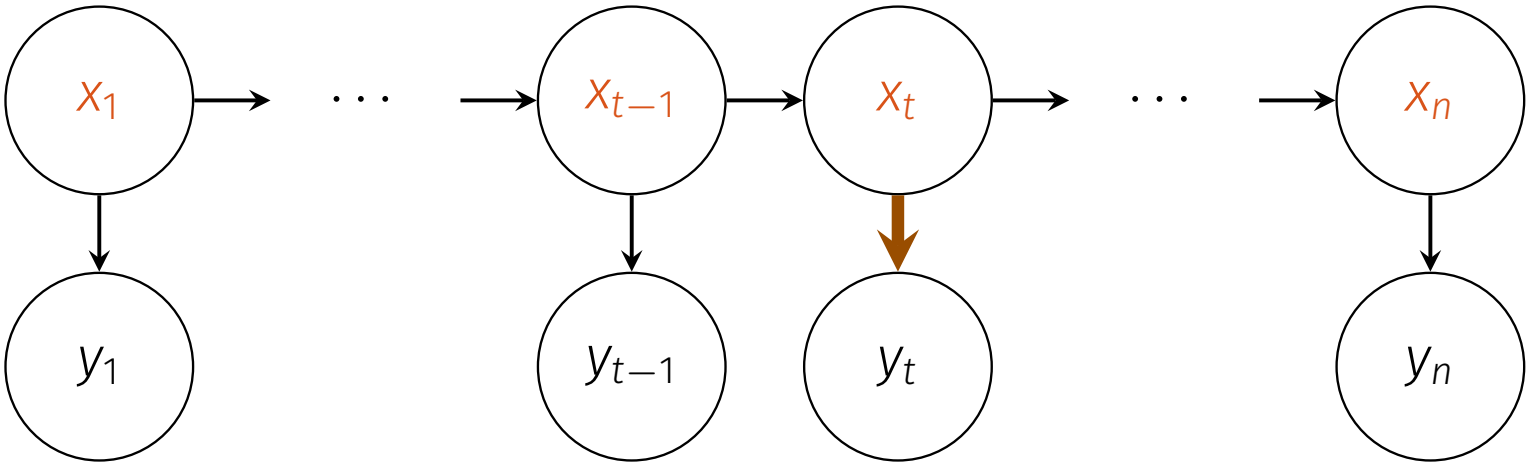
Hidden Markov Models (HMM)



$$X_1 \sim p(x_1) \qquad \text{initial density}$$

$$X_t \mid X_{t-1} = x_{t-1} \sim p(x_t \mid x_{t-1}) \qquad \text{transition density}$$

Hidden Markov Models (HMM)



$$X_1 \sim p(x_1) \qquad \text{initial density}$$

$$X_t \mid X_{t-1} = x_{t-1} \sim p(x_t \mid x_{t-1}) \qquad \text{transition density}$$

$$Y_t \mid X_t = x_t \sim p(y_t \mid x_t) \qquad \text{observation density}$$

Lotka-Volterra Example

We have

- Latent process: $X_t = (X_{\text{prey},t}, X_{\text{pred},t})^\top = (X_{1,t}, X_{2,t})^\top$
- Observed process: $Y_t = (Y_{\text{prey},t}, Y_{\text{pred},t})^\top = (Y_{1,t}, Y_{2,t})^\top$

Observation density

$$Y_{1,t} \mid X_{1,t} = x_{1,t} \sim \text{Poisson}(0.5 \cdot x_{1,t})$$
$$Y_{2,t} \mid X_{2,t} = x_{2,t} \sim \text{Poisson}(0.8 \cdot x_{2,t})$$

Initial states are known: $X_0 = \{71, 79\}^\top$

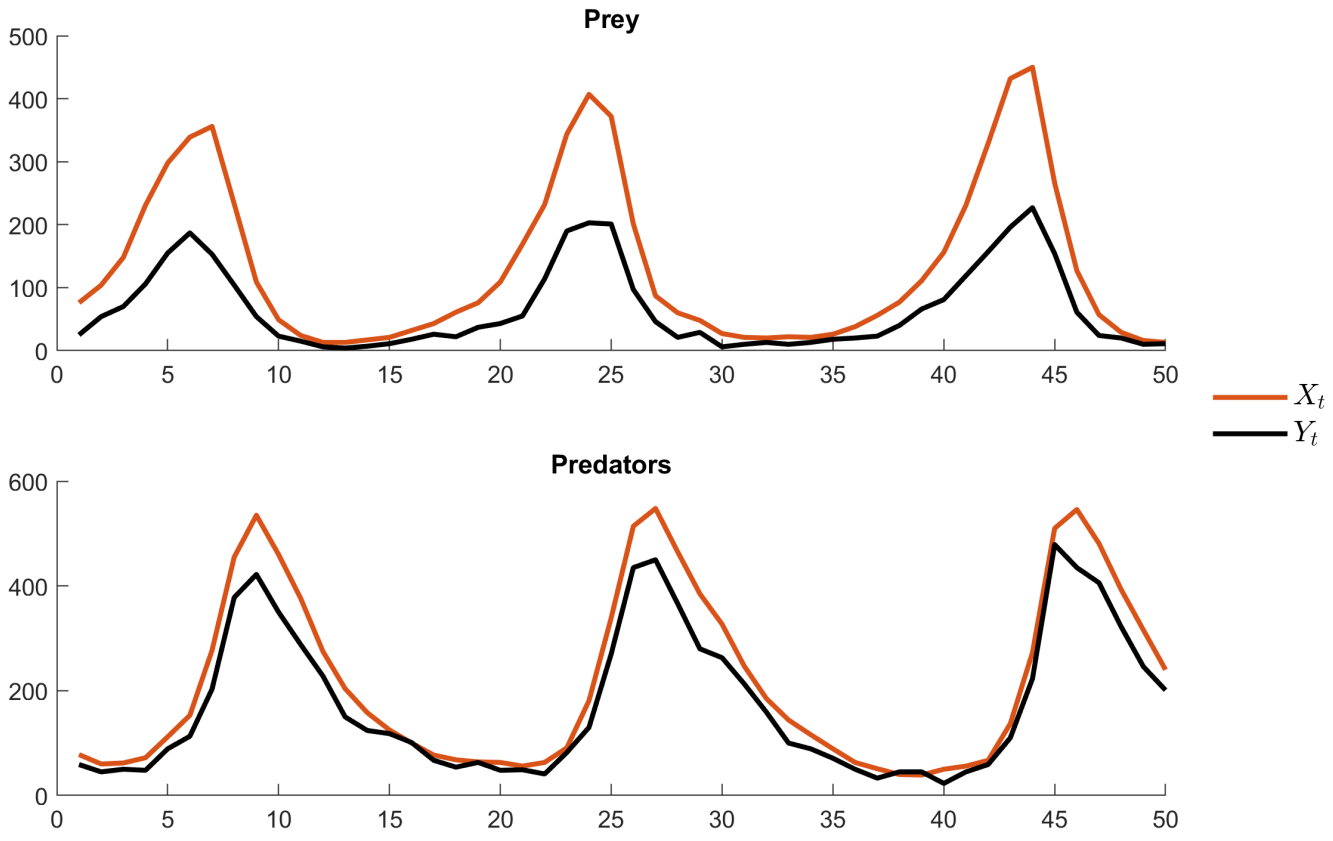
Lotka-Volterra Example

Transition density is a Markov jump process (MJP) $\rightarrow$ 3 possible reactions that occur at a particular rate

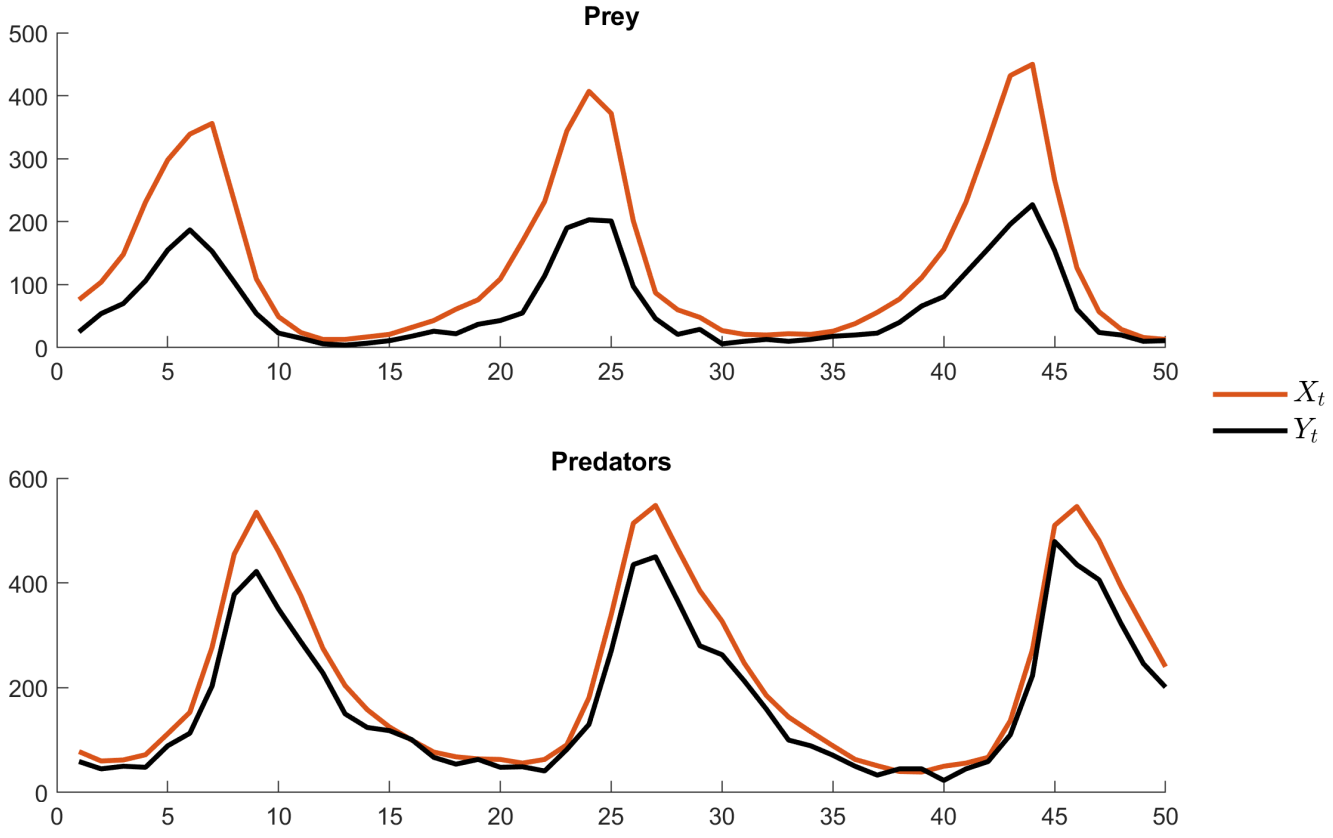
For small dt, the probabilities of a transition over $(t, t + dt]$ are

$$
\begin{aligned}
P(\text{prey reproduction}) &\approx 0.5 \cdot x_{1,t} dt \\
P(\text{prey death, predator reproduction}) &\approx 0.0025 \cdot x_{1,t} x_{2,t} dt \\
P(\text{predator death}) &\approx 0.3 \cdot x_{2,t} dt
\end{aligned}
$$

Lotka-Volterra Example



Lotka-Volterra Example



Want to track the true population of the prey/predators at time t given the observations.

Hidden Markov Models

Our goal is state estimation $\rightarrow$ want to learn $x_{1:n}$ from $y_{1:n}$

Hidden Markov Models

Our goal is state estimation $\rightarrow$ want to learn $x_{1:n}$ from $y_{1:n}$

$$p(x_{1:n} \mid y_{1:n}) = \frac{p(x_{1:n}, y_{1:n})}{p(y_{1:n})} \propto p(x_{1:n}, y_{1:n})$$

Hidden Markov Models

Our goal is state estimation $\rightarrow$ want to learn $x_{1:n}$ from $y_{1:n}$

$$p(x_{1:n} \mid y_{1:n}) = \frac{p(x_{1:n}, y_{1:n})}{p(y_{1:n})} \propto p(x_{1:n}, y_{1:n})$$

$$p(x_{1:n}, y_{1:n}) = p(y_{1:n} \mid x_{1:n}) p(x_{1:n})$$

Hidden Markov Models

Our goal is state estimation $\rightarrow$ want to learn $x_{1:n}$ from $y_{1:n}$

$$p(x_{1:n} \mid y_{1:n}) = \frac{p(x_{1:n}, y_{1:n})}{p(y_{1:n})} \propto p(x_{1:n}, y_{1:n})$$

$$\begin{aligned} p(x_{1:n}, y_{1:n}) &= p(y_{1:n} \mid x_{1:n}) p(x_{1:n}) \\ &= \text{likelihood} \times \text{prior} \end{aligned}$$

Hidden Markov Models

Our goal is state estimation $\rightarrow$ want to learn $x_{1:n}$ from $y_{1:n}$

$$p(x_{1:n} \mid y_{1:n}) = \frac{p(x_{1:n}, y_{1:n})}{p(y_{1:n})} \propto p(x_{1:n}, y_{1:n})$$

$$\begin{aligned} p(x_{1:n}, y_{1:n}) &= p(y_{1:n} \mid x_{1:n}) p(x_{1:n}) \\ &= \text{likelihood} \times \text{prior} \end{aligned}$$

Can construct $p(x_{1:n}, y_{1:n})$ using $p(x_1)$, $p(x_t \mid x_{t-1})$ and $p(y_t \mid x_t)$

Hidden Markov Models

Full joint density

$$p(x_{1:n}, y_{1:n}) = \underbrace{p(y_{1:n} \mid x_{1:n})}_{\text{likelihood}} p(x_{1:n})$$

$$p(y_{1:n} \mid x_{1:n})$$

Hidden Markov Models

Full joint density

$$p(x_{1:n}, y_{1:n}) = \underbrace{p(y_{1:n} \mid x_{1:n})}_{\text{likelihood}} p(x_{1:n})$$

$$p(y_{1:n} \mid x_{1:n}) = \prod_{t=1}^{n} p(y_t \mid x_{1:n})$$

Hidden Markov Models

Full joint density

$$p(x_{1:n}, y_{1:n}) = \underbrace{p(y_{1:n} \mid x_{1:n})}_{\text{likelihood}} p(x_{1:n})$$

$$\begin{aligned} p(y_{1:n} \mid x_{1:n}) &= \prod_{t=1}^{n} p(y_t \mid x_{1:n}) \\ &= \prod_{t=1}^{n} p(y_t \mid x_t) \quad \text{conditional independence} \end{aligned}$$

Hidden Markov Models

Full joint density

$$p(x_{1:n}, y_{1:n}) = p(y_{1:n} \mid x_{1:n}) \underbrace{p(x_{1:n})}_{\text{state prior}}$$

$p(x_{1:n})$

Hidden Markov Models

Full joint density

$$p(x_{1:n}, y_{1:n}) = p(y_{1:n} \mid x_{1:n}) \underbrace{p(x_{1:n})}_{\text{state prior}}$$

$$p(x_{1:n}) = p(x_n \mid x_{1:n-1}) p(x_{1:n-1})$$

Hidden Markov Models

Full joint density

$$p(x_{1:n}, y_{1:n}) = p(y_{1:n} \mid x_{1:n}) \underbrace{p(x_{1:n})}_{\text{state prior}}$$

$$\begin{aligned} p(x_{1:n}) &= p(x_n \mid x_{1:n-1}) p(x_{1:n-1}) \\ &= p(x_n \mid x_{n-1}) p(x_{1:n-1}) \quad \text{conditional independence} \end{aligned}$$

Hidden Markov Models

Full joint density

$$p(x_{1:n}, y_{1:n}) = p(y_{1:n} \mid x_{1:n}) \underbrace{p(x_{1:n})}_{\text{state prior}}$$

$$\begin{aligned} p(x_{1:n}) &= p(x_n \mid x_{1:n-1}) p(x_{1:n-1}) \\ &= p(x_n \mid x_{n-1}) p(x_{1:n-1}) \quad \text{conditional independence} \\ &= p(x_n \mid x_{n-1}) p(x_{n-1} \mid x_{n-2}) p(x_{1:n-2}) \end{aligned}$$

Hidden Markov Models

Full joint density

$$p(x_{1:n}, y_{1:n}) = p(y_{1:n} \mid x_{1:n}) \underbrace{p(x_{1:n})}_{\text{state prior}}$$

$$\begin{aligned}
p(x_{1:n}) &= p(x_n \mid x_{1:n-1}) p(x_{1:n-1}) \\
&= p(x_n \mid x_{n-1}) p(x_{1:n-1}) \quad \text{conditional independence} \\
&= p(x_n \mid x_{n-1}) p(x_{n-1} \mid x_{n-2}) p(x_{1:n-2}) \\
&= p(x_1) \prod_{t=2}^{n} p(x_t \mid x_{t-1})
\end{aligned}$$

Hidden Markov Models

Full joint density

$$
\begin{aligned}
p(x_{1:n}, y_{1:n}) &= p(y_{1:n} \mid x_{1:n}) p(x_{1:n}) \\
&= \prod_{t=1}^{n} p(y_t \mid x_t) \left[\prod_{t=2}^{n} p(x_t \mid x_{t-1}) \right] p(x_1)
\end{aligned}
$$

Hidden Markov Models

Full joint density

$$
\begin{aligned}
p(x_{1:n}, y_{1:n}) &= p(y_{1:n} \mid x_{1:n}) p(x_{1:n}) \\
&= \prod_{t=1}^{n} p(y_t \mid x_t) \left[\prod_{t=2}^{n} p(x_t \mid x_{t-1}) \right] p(x_1)
\end{aligned}
$$

Filtering	$p(x_t \mid y_{1:t})$
Predictive	$p(x_t \mid y_{1:t-1})$
Joint filtering	$p(x_{1:t} \mid y_{1:t})$
Smoothing	$p(x_t \mid y_{1:n})$
Joint smoothing	$p(x_{1:t} \mid y_{1:n})$

Hidden Markov Models

Full joint density

$$
\begin{aligned}
p(x_{1:n}, y_{1:n}) &= p(y_{1:n} \mid x_{1:n}) p(x_{1:n}) \\
&= \prod_{t=1}^{n} p(y_t \mid x_t) \left[\prod_{t=2}^{n} p(x_t \mid x_{t-1}) \right] p(x_1)
\end{aligned}
$$

Filtering	$p(x_t \mid y_{1:t})$
Predictive	$p(x_t \mid y_{1:t-1})$
Joint filtering	$p(x_{1:t} \mid y_{1:t})$
Smoothing	$p(x_t \mid y_{1:n})$
Joint smoothing	$p(x_{1:t} \mid y_{1:n})$

Lotka-Volterra Example

Filtering distribution $p(x_t \mid y_{1:t})$

- Population at time t given observations up to time t

Predictive distribution $p(x_t \mid y_{1:t-1})$

- Population at time t given observations up to time $t-1$

Joint filtering distribution $p(x_{1:t} \mid y_{1:t})$

- Evolution of population up to time t given observations up to time t

Filtering Problem

Filtering Problem

The filtering problem $\rightarrow$ learn the filtering distribution $p(x_t \mid y_{1:t})$.

Filtering Problem

The filtering problem $\rightarrow$ learn the filtering distribution $p(x_t \mid y_{1:t})$.

Can learn $p(x_t \mid y_{1:t})$ by recursively calculating the predictive density $p(x_t \mid y_{1:t-1})$ and then updating with the information in y_t.

Filtering Problem

The filtering problem $\rightarrow$ learn the filtering distribution $p(x_t \mid y_{1:t})$.

Can learn $p(x_t \mid y_{1:t})$ by recursively calculating the predictive density $p(x_t \mid y_{1:t-1})$ and then updating with the information in y_t.

At time $t = 1$, $p(x_1)$ is our initial predictive density,
$p(x_t \mid y_{1:t-1}) = p(x_1 \mid y_{1:0}) = p(x_1)$

Filtering Problem

For time $t \geq 1$:

Filtering Problem

For time $t \geq 1$:

1. **Measurement update** to incorporate y_t

$$p(x_t \mid y_{1:t}) = \frac{p(y_t \mid x_t)p(x_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})}$$

$$p(y_t \mid y_{1:t-1}) = \int p(y_t \mid x_t)p(x_t \mid y_{1:t-1})dx_t$$

Filtering Problem

For time $t \geq 1$:

1. **Measurement update** to incorporate y_t

$$p(x_t \mid y_{1:t}) = \frac{p(y_t \mid x_t)p(x_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})}$$

$$p(y_t \mid y_{1:t-1}) = \int p(y_t \mid x_t)p(x_t \mid y_{1:t-1})dx_t$$

2. **Time update** to get the predictive distribution

$$p(x_{t+1} \mid y_{1:t}) = \int p(x_{t+1} \mid x_t)p(x_t \mid y_{1:t})dx_t$$

Filtering Problem

We generally can't solve the filtering problem analytically

Linear Gaussian models $\rightarrow$ the Kalman filter gives an exact solution

Non-linear and non-Gaussian models $\rightarrow$ particle filter

Summary

A hidden Markov model (HMM) has latent states $x_{1:n}$ (unknown) and observations $y_{1:n}$ (known)

Summary

A hidden Markov model (HMM) has latent states $x_{1:n}$ (unknown) and observations $y_{1:n}$ (known)

The model is completely specified by the densities

$$
\begin{aligned}
X_1 &\sim p(x_1) && \text{initial density} \\
X_t \mid X_{t-1} = x_{t-1} &\sim p(x_t \mid x_{t-1}) && \text{transition density} \\
Y_t \mid X_t = x_t &\sim p(y_t \mid x_t) && \text{observation density}
\end{aligned}
$$

Summary

We want to solve the filtering problem $\rightarrow$ learn x_t conditional on $y_{1:t}$

Summary

We want to solve the filtering problem $\rightarrow$ learn x_t conditional on $y_{1:t}$

- Measurement update (update with y_t)

$$p(x_t \mid y_{1:t}) \propto p(y_t \mid x_t) p(x_t \mid y_{1:t-1})$$

- Time update (predict for $t+1$)

$$p(x_{t+1} \mid y_{1:t}) = \int p(x_{t+1} \mid x_t) p(x_t \mid y_{1:t}) dx_t$$

Summary

We want to solve the filtering problem $\rightarrow$ learn x_t conditional on $y_{1:t}$

- Measurement update (update with y_t)

$$p(x_t \mid y_{1:t}) \propto p(y_t \mid x_t)p(x_t \mid y_{1:t-1})$$

- Time update (predict for $t+1$)

$$p(x_{t+1} \mid y_{1:t}) = \int p(x_{t+1} \mid x_t)p(x_t \mid y_{1:t})dx_t$$

A particle filter can be used to solve the filtering problem for non-linear and non-Gaussian models

Particle Filtering

Particle Filter

A particle filter builds the filtering distribution recursively using N samples/particles

Uses importance sampling (self-normalised) to construct the joint filtering distribution

- $w_t^i \rightarrow$ unnormalised weight
- $W_t^i = w_t^i / \sum_{i=1}^N w_t^i \rightarrow$ normalised weight

At a minimum, we need to be able to

- sample from $p(x_1)$ and $p(x_t \mid x_{t-1})$
- evaluate $p(y_t \mid x_t)$

Lotka-Volterra Example

Initial states are known: $X_0 = \{71, 79\}^\top$

For small dt, the probabilities of a transition over $(t, t + dt]$ are

$$
\begin{aligned}
&P(\text{prey reproduction}) && \approx 0.5 \cdot x_{1,t} dt \\
&P(\text{prey death, predator reproduction}) && \approx 0.0025 \cdot x_{1,t} x_{2,t} dt \\
&P(\text{predator death}) && \approx 0.3 \cdot x_{2,t} dt
\end{aligned}
$$

Observation density

$$
\begin{aligned}
Y_{1,t} \mid X_{1,t} = x_{1,t} &\sim \text{Poisson}(0.5 \cdot x_{1,t}) \\
Y_{2,t} \mid X_{2,t} = x_{2,t} &\sim \text{Poisson}(0.8 \cdot x_{2,t})
\end{aligned}
$$

Particle Filter

At time $t = 1$

- $\zeta_1^i \overset{\text{iid}}{\sim} p(x_1)$ for $i \in \{1, \ldots, N\}$
- Set the weights as $w_0^i = \frac{1}{N}$ for $i \in \{1, \ldots, N\}$

We now have the predictive sample for time $t = 1$:
$\{\zeta_1^i, w_0^i = \frac{1}{N}\}_{i=1}^N \sim p(x_1 \mid y_{1:0}) = p(x_1)$

Particle Filter

Measurement update

$$p(x_t \mid y_{1:t}) \propto p(y_t \mid x_t) p(x_t \mid y_{1:t-1})$$

Particle Filter

Measurement update

$$p(x_t \mid y_{1:t}) \propto p(y_t \mid x_t) p(x_t \mid y_{1:t-1})$$

Input: predictive $\{\zeta_t^i, w_{t-1}^i\}_{i=1}^N \sim p(x_t \mid y_{1:t-1})$

Particle Filter

Measurement update

$$p(x_t \mid y_{1:t}) \propto p(y_t \mid x_t) p(x_t \mid y_{1:t-1})$$

Input: predictive $\{\zeta_t^i, w_{t-1}^i\}_{i=1}^N \sim p(x_t \mid y_{1:t-1})$

Update the weights $w_t^i = w_{t-1}^i p(y_t \mid x_t = \zeta_t^i)$ for $i \in \{1, \ldots, N\}$

Particle Filter

Measurement update

$$p(x_t \mid y_{1:t}) \propto p(y_t \mid x_t) p(x_t \mid y_{1:t-1})$$

Input: predictive $\{\zeta_t^i, w_{t-1}^i\}_{i=1}^N \sim p(x_t \mid y_{1:t-1})$

Update the weights $w_t^i = w_{t-1}^i p(y_t \mid x_t = \zeta_t^i)$ for $i \in \{1, \ldots, N\}$

Output: filtering $\{\zeta_t^i, w_t^i\}_{i=1}^N \sim p(x_t \mid y_{1:t})$

Particle Filter

Time update

$$p(x_{t+1} \mid y_{1:t}) = \int p(x_{t+1} \mid x_t) p(x_t \mid y_{1:t}) dx_t$$

Particle Filter

Time update

$$p(x_{t+1} \mid y_{1:t}) = \int p(x_{t+1} \mid x_t) p(x_t \mid y_{1:t}) dx_t$$

Input: filtering $\{\zeta_t^i, w_t^i\}_{i=1}^N \sim p(x_t \mid y_{1:t})$

Particle Filter

Time update

$$p(x_{t+1} \mid y_{1:t}) = \int p(x_{t+1} \mid x_t) p(x_t \mid y_{1:t}) dx_t$$

Input: filtering $\{\zeta_t^i, w_t^i\}_{i=1}^N \sim p(x_t \mid y_{1:t})$

Sample predictions $\zeta_{t+1}^i \sim p(x_{t+1} \mid x_t = \zeta_t^i)$ for $i \in \{1, \ldots, N\}$

Particle Filter

Time update

$$p(x_{t+1} \mid y_{1:t}) = \int p(x_{t+1} \mid x_t) p(x_t \mid y_{1:t}) dx_t$$

Input: filtering $\{\zeta_t^i, w_t^i\}_{i=1}^N \sim p(x_t \mid y_{1:t})$

Sample predictions $\zeta_{t+1}^i \sim p(x_{t+1} \mid x_t = \zeta_t^i)$ for $i \in \{1, \ldots, N\}$

Output: predictive $\{\zeta_{t+1}^i, w_t^i\}_{i=1}^N \sim p(x_{t+1} \mid y_{1:t})$

Particle Filter

Putting it all together

1. Sample initial $\zeta_1^i \overset{\text{iid}}{\sim} p(x_1)$ and set $w_0^i = \frac{1}{N}$ for $i \in \{1, \ldots, N\}$
2. For each time $t = 1, \ldots, n-1$
   (i) Update the weights $w_t^i = w_{t-1}^i p(y_t \mid x_t = \zeta_t^i)$ for $i \in \{1, \ldots, N\}$
   (ii) Sample predictions $\zeta_{t+1}^i \sim p(x_{t+1} \mid x_t = \zeta_t^i)$ for $i \in \{1, \ldots, N\}$
3. Calculate terminal weights w_n^i as in step 2(i)

Output: Particles $\zeta_t^{1:N}$ and weights $w_t^{1:N}$ for $t = 1, \ldots, n$

Particle Filter

Putting it all together

1. Sample initial $\zeta_1^i \overset{\text{iid}}{\sim} p(x_1)$ and set $w_0^i = \frac{1}{N}$ for $i \in \{1, \ldots, N\}$
2. For each time $t = 1, \ldots, n-1$
   (i) Update the weights $w_t^i = w_{t-1}^i p(y_t \mid x_t = \zeta_t^i)$ for $i \in \{1, \ldots, N\}$
   (ii) Sample predictions $\zeta_{t+1}^i \sim p(x_{t+1} \mid x_t = \zeta_t^i)$ for $i \in \{1, \ldots, N\}$
3. Calculate terminal weights w_n^i as in step 2(i)

Output: Particles $\zeta_t^{1:N}$ and weights $w_t^{1:N}$ for $t = 1, \ldots, n$

Not quite a particle filter yet... still just importance sampling

Particle Filter

At iteration t:

Particle Filter

At iteration t:

The target is the joint filtering distribution (posterior of $x_{1:t}$)

$$p(x_{1:t} \mid y_{1:t}) \propto \underbrace{p(y_{1:t} \mid x_{1:t})}_{\text{likelihood}} \underbrace{p(x_{1:t})}_{\text{state prior}}$$

Particle Filter

At iteration t:

The target is the joint filtering distribution (posterior of $x_{1:t}$)

$$p(x_{1:t} \mid y_{1:t}) \propto \underbrace{p(y_{1:t} \mid x_{1:t})}_{\text{likelihood}} \underbrace{p(x_{1:t})}_{\text{state prior}}$$

The proposal is the state prior $p(x_{1:t}) = p(x_1) \prod_{j=2}^{t} p(x_j \mid x_{j-1})$

Particle Filter

At iteration t:

The target is the joint filtering distribution (posterior of $x_{1:t}$)

$$p(x_{1:t} \mid y_{1:t}) \propto \underbrace{p(y_{1:t} \mid x_{1:t})}_{\text{likelihood}} \underbrace{p(x_{1:t})}_{\text{state prior}}$$

The proposal is the state prior $p(x_{1:t}) = p(x_1) \prod_{j=2}^{t} p(x_j \mid x_{j-1})$

The unnormalised weights are

$$w_t = \frac{\text{target}}{\text{proposal}} = \frac{p(y_{1:t} \mid x_{1:t}) p(x_{1:t})}{p(x_{1:t})}$$

$$= p(y_{1:t} \mid x_{1:t}) = \prod_{j=1}^{t} p(y_j \mid x_j)$$

Particle Filter

Not very efficient...

Particle Filter

Not very efficient...

Since we're sampling over the joint filtering distribution $\{\zeta_{1:t}, w_t^i\}_{i=1}^N \sim p(x_{1:t} \mid y_{1:t})$, the dimension increases at each iteration

Particle Filter

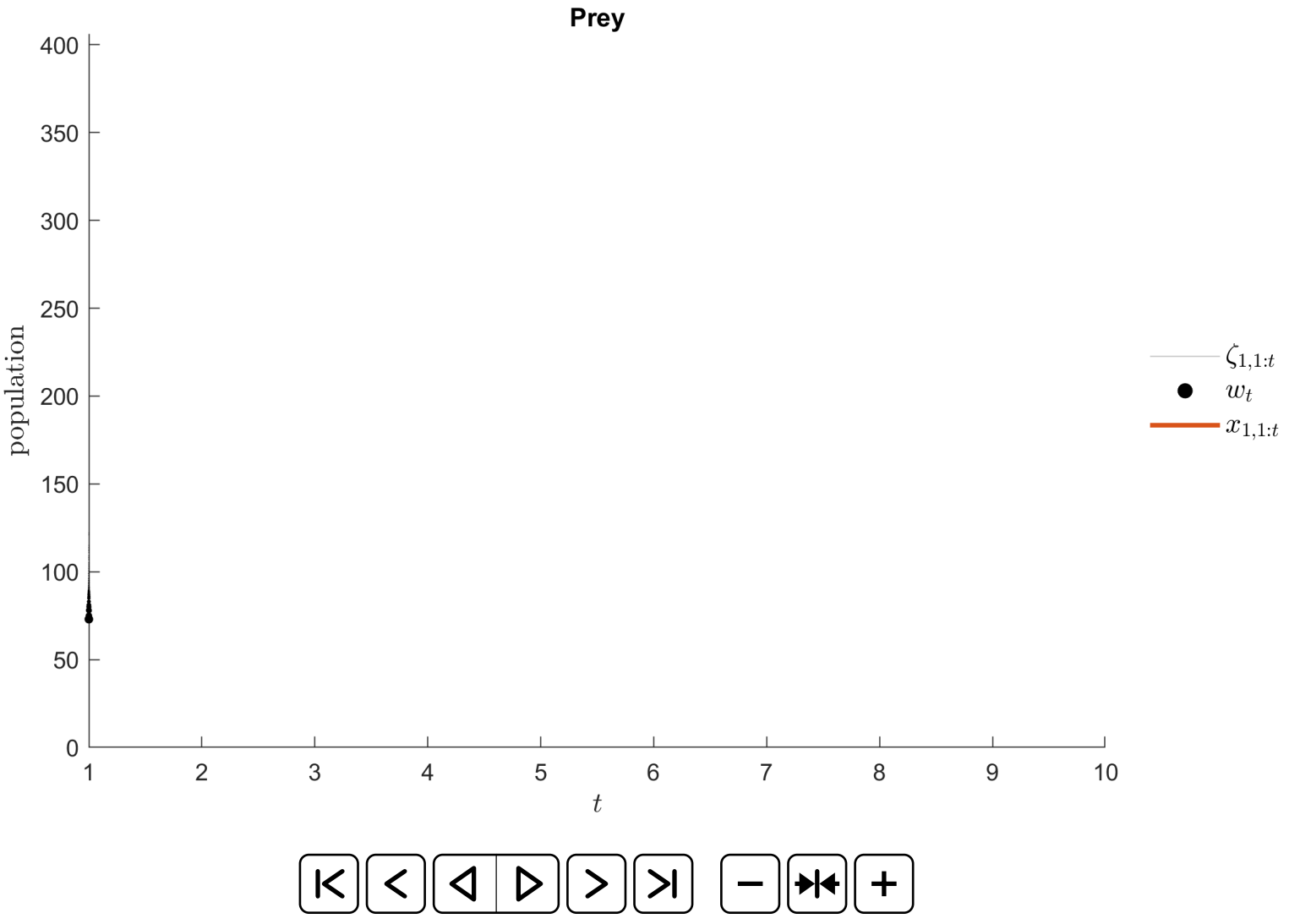
Not very efficient...

Since we're sampling over the joint filtering distribution $\{\zeta_{1:t}, w_t^i\}_{i=1}^N \sim p(x_{1:t} \mid y_{1:t})$, the dimension increases at each iteration

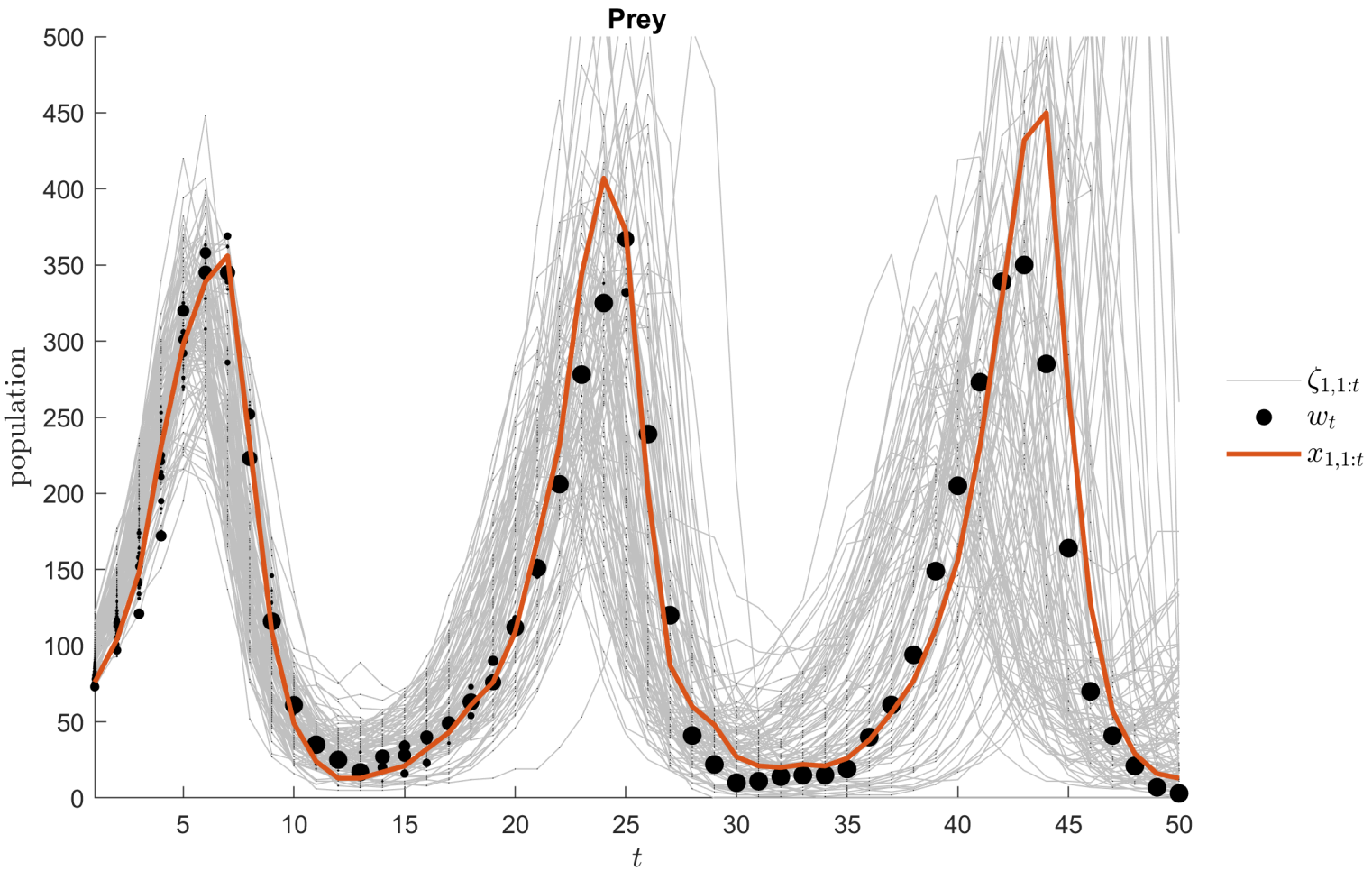
The joint proposal is diverging more and more from the target

- particle degeneracy $\rightarrow$ most of the particles have 0 weight
- the effective sample size decreases

Lotka-Volterra Example



Lotka-Volterra Example



Particle Filter

Possible solutions

- Increase $N \rightarrow$ expensive
- Resampling

Particle Filter

Time update

$$p(x_{t+1} \mid y_{1:t}) = \int p(x_{t+1} \mid x_t) p(x_t \mid y_{1:t}) dx_t$$

Input: filtering $\{\zeta_t^i, w_t^i\}_{i=1}^N \sim p(x_t \mid y_{1:t})$

Particle Filter

Time update

$$p(x_{t+1} \mid y_{1:t}) = \int p(x_{t+1} \mid x_t) p(x_t \mid y_{1:t}) dx_t$$

Input: filtering $\{\zeta_t^i, w_t^i\}_{i=1}^N \sim p(x_t \mid y_{1:t})$

Sample ancestors $A_t^i \sim \text{Cat}\left(w_t^1, \ldots, w_t^N\right)$ for $i \in \{1, \ldots, N\}$

Particle Filter

Time update

$$p(x_{t+1} \mid y_{1:t}) = \int p(x_{t+1} \mid x_t) p(x_t \mid y_{1:t}) dx_t$$

Input: filtering $\{\zeta_t^i, w_t^i\}_{i=1}^N \sim p(x_t \mid y_{1:t})$

Sample ancestors $A_t^i \sim \text{Cat}\left(w_t^1, \ldots, w_t^N\right)$ for $i \in \{1, \ldots, N\}$

We now have $\{\hat{\zeta}_t^i = \zeta_t^{A_t^i}, w_t^i = \frac{1}{N}\}_{i=1}^N \sim p(x_t \mid y_{1:t})$

Particle Filter

Time update

$$p(x_{t+1} \mid y_{1:t}) = \int p(x_{t+1} \mid x_t) p(x_t \mid y_{1:t}) dx_t$$

Input: filtering $\{\zeta_t^i, w_t^i\}_{i=1}^N \sim p(x_t \mid y_{1:t})$

Sample ancestors $A_t^i \sim \text{Cat}\left(w_t^1, \ldots, w_t^N\right)$ for $i \in \{1, \ldots, N\}$

We now have $\{\hat{\zeta}_t^i = \zeta_t^{A_t^i}, w_t^i = \frac{1}{N}\}_{i=1}^N \sim p(x_t \mid y_{1:t})$

Sample predictions $\zeta_{t+1}^i \sim p(x_{t+1} \mid x_t = \hat{\zeta}_t^i)$ for $i \in \{1, \ldots, N\}$

Particle Filter

At iteration t:

Particle Filter

At iteration t:

The target is still the joint filtering distribution (posterior of $x_{1:t}$)

$$p(x_{1:t} \mid y_{1:t}) \propto \underbrace{p(y_{1:t} \mid x_{1:t})}_{\text{likelihood}} \underbrace{p(x_{1:t})}_{\text{state prior}}$$

Particle Filter

At iteration t:

The target is still the joint filtering distribution (posterior of $x_{1:t}$)

$$p(x_{1:t} \mid y_{1:t}) \propto \underbrace{p(y_{1:t} \mid x_{1:t})}_{\text{likelihood}} \underbrace{p(x_{1:t})}_{\text{state prior}}$$

The proposal is $p(x_{1:t} \mid y_{1:t-1})$ (much closer to $p(x_{1:t} \mid y_{1:t})$!)

Particle Filter

At iteration t:

The target is still the joint filtering distribution (posterior of $x_{1:t}$)

$$p(x_{1:t} \mid y_{1:t}) \propto \underbrace{p(y_{1:t} \mid x_{1:t})}_{\text{likelihood}} \underbrace{p(x_{1:t})}_{\text{state prior}}$$

The proposal is $p(x_{1:t} \mid y_{1:t-1})$ (much closer to $p(x_{1:t} \mid y_{1:t})$!)

The weights are

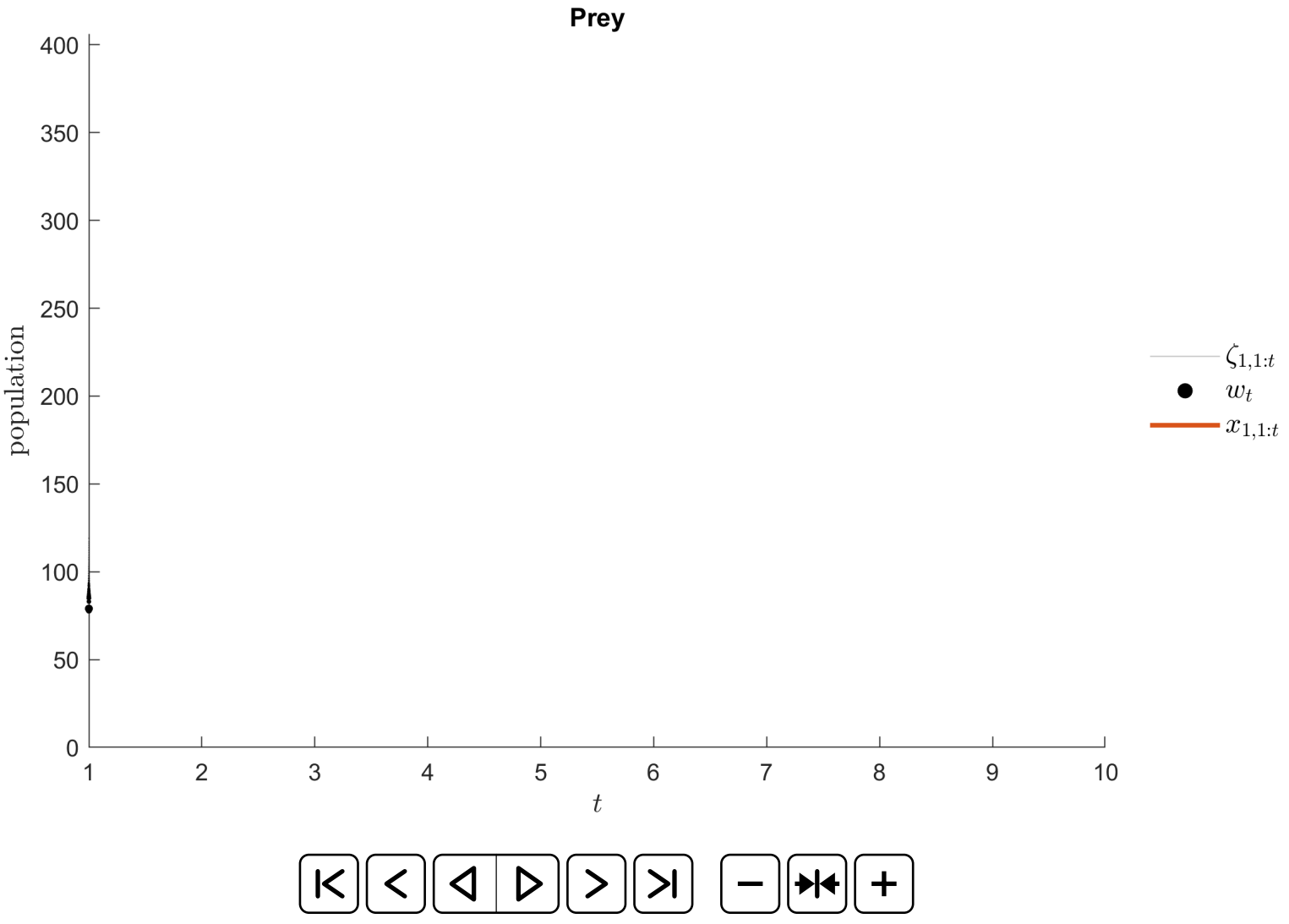
$$w_t = \frac{p(x_{1:t}, y_{1:t})}{p(x_{1:t}, y_{1:t-1})} = p(y_t \mid x_{1:t}, y_{1:t-1}) = p(y_t \mid x_t)$$

Particle Filter

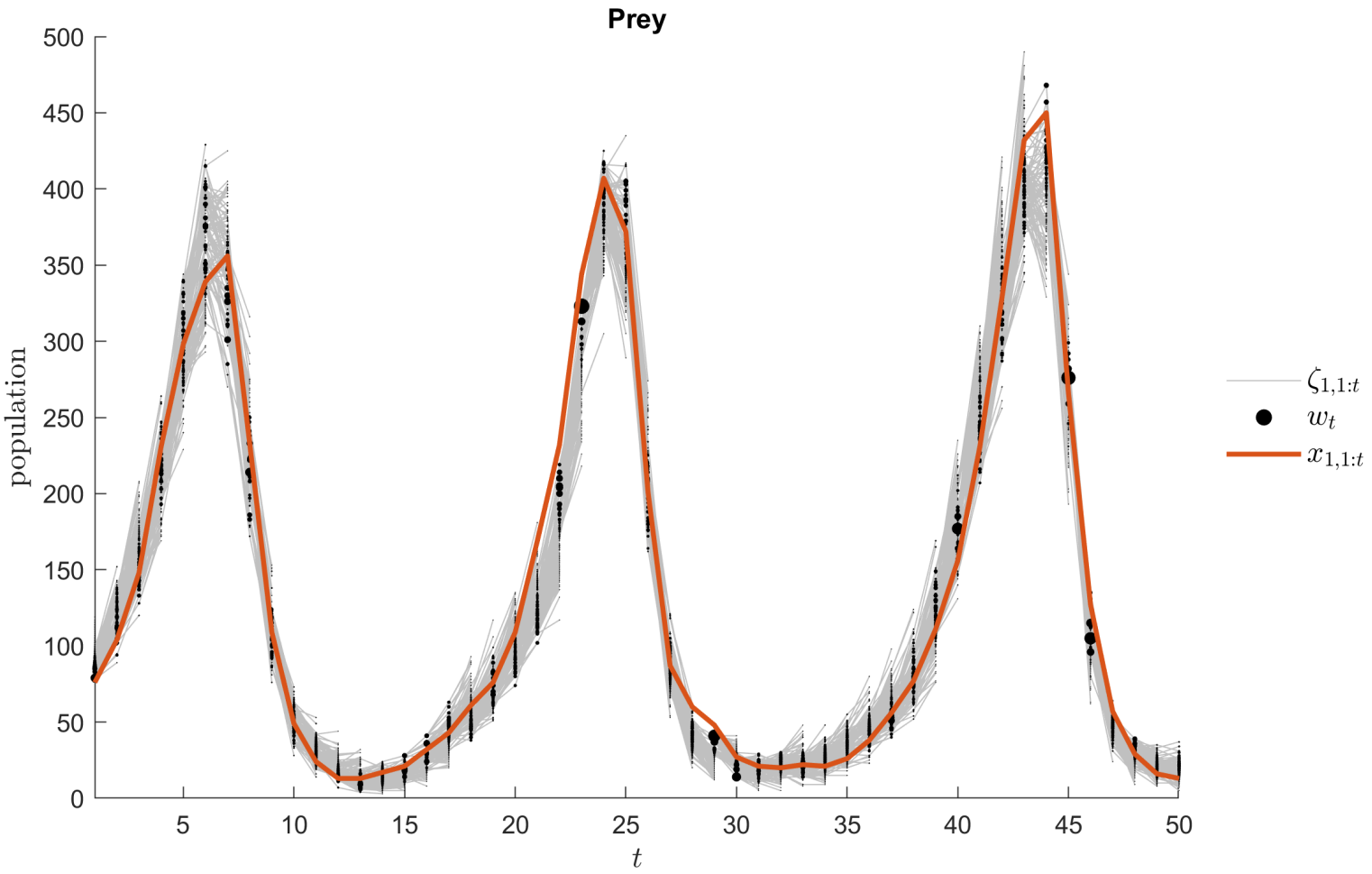
1. Sample initial $\zeta_1^i \overset{\text{iid}}{\sim} p(x_1)$ for $i \in \{1, \ldots, N\}$
2. For each time $t = 1, \ldots, n-1$
   (i) Calculate weights $w_t^i = p(y_t \mid x_t = \zeta_t^i)$ for $i \in \{1, \ldots, N\}$
   (ii) Sample ancestors $A_t^i \sim \text{Cat}\left(w_t^1, \ldots, w_t^N\right)$ for $i \in \{1, \ldots, N\}$
   (iii) Sample prediction $\zeta_{t+1}^i \sim p(x_{t+1} \mid x_t = \zeta_t^{A_t^i})$ for $i \in \{1, \ldots, N\}$
3. Calculate terminal weights w_n^i as in step 2(i)

Output: Particles $\zeta_t^{1:N}$ and weights $w_t^{1:N}$ for $t = 1, \ldots, n$

Lotka-Volterra Example



Lotka-Volterra Example



Particle Filter Output

The particle filter gives a weighted sample from $p(x_{1:n} \mid y_{1:n}) \rightarrow \{\zeta^i_{1:n}, w^i_{1:n}\}_{i=1}^N \sim p(x_{1:n} \mid y_{1:n})$

Particle Filter Output

The particle filter gives a weighted sample from $p(x_{1:n} \mid y_{1:n}) \to$
$\{\zeta^i_{1:n}, w^i_{1:n}\}_{i=1}^N \sim p(x_{1:n} \mid y_{1:n})$

At any iteration t:

$$\begin{aligned}
&\text{(i)} && \{\zeta^i_t, w^i_t\}_{i=1}^N && \sim p(x_t \mid y_{1:t}) && \text{filtering distribution} \\
&\text{(ii)} && \{\zeta^i_{t+1}, w^i_t\}_{i=1}^N && \sim p(x_{t+1} \mid y_{1:t}) && \text{predictive distribution}
\end{aligned}$$

Particle Filter Output

The particle filter gives a weighted sample from $p(x_{1:n} \mid y_{1:n}) \rightarrow \{\zeta^i_{1:n}, w^i_{1:n}\}_{i=1}^N \sim p(x_{1:n} \mid y_{1:n})$

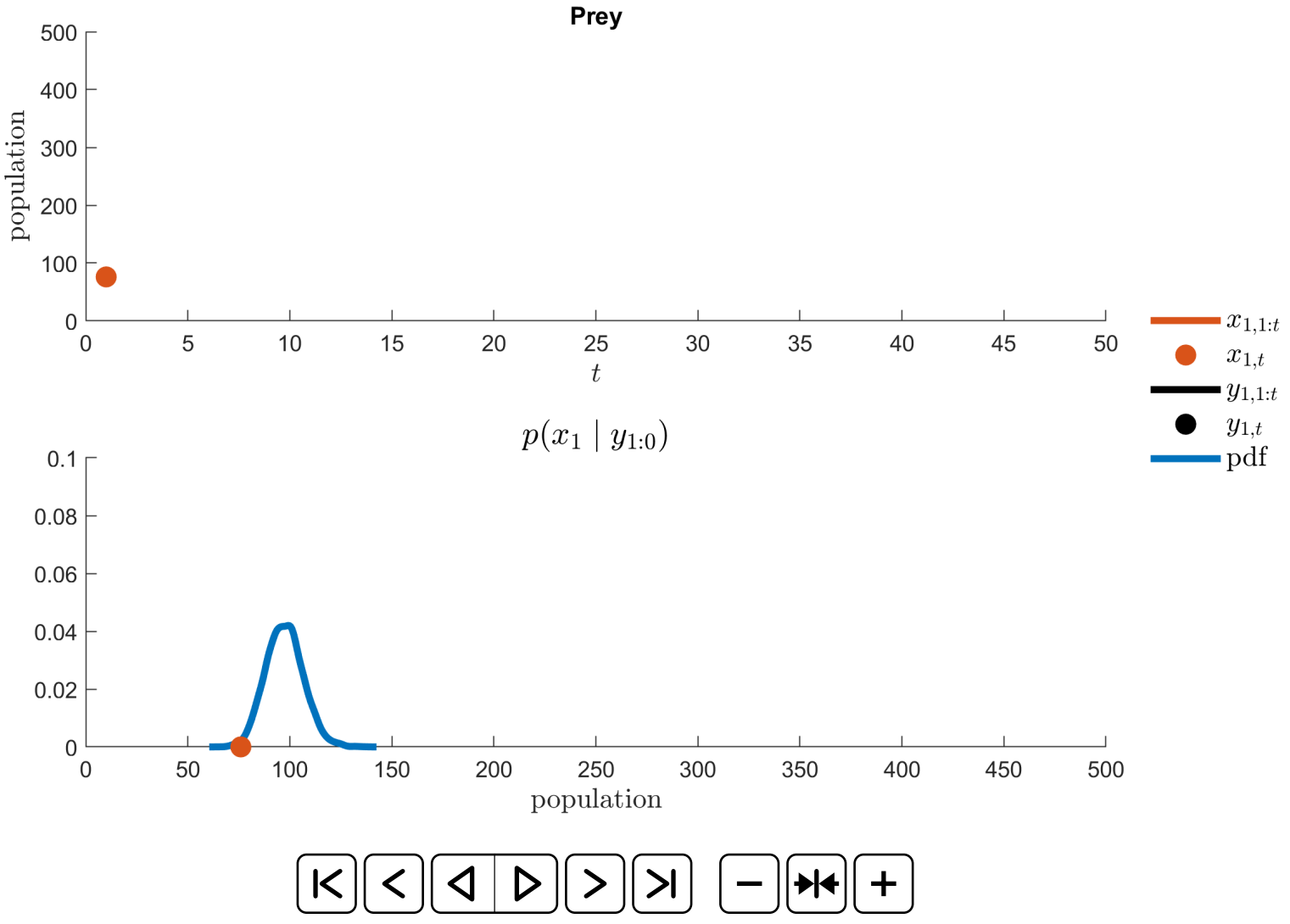
At any iteration t:

$$\begin{array}{lll} \text{(i)} & \{\zeta^i_t, w^i_t\}_{i=1}^N \sim p(x_t \mid y_{1:t}) & \text{filtering distribution} \\ \text{(ii)} & \{\zeta^i_{t+1}, w^i_t\}_{i=1}^N \sim p(x_{t+1} \mid y_{1:t}) & \text{predictive distribution} \end{array}$$

Can estimate the marginal likelihood

$$p(y_{1:n}) = \prod_{t=1}^{n} p(y_t \mid y_{1:t-1}) \approx \prod_{t=1}^{n} \sum_{i=1}^{N} w^i_t$$

Lotka-Volterra Example

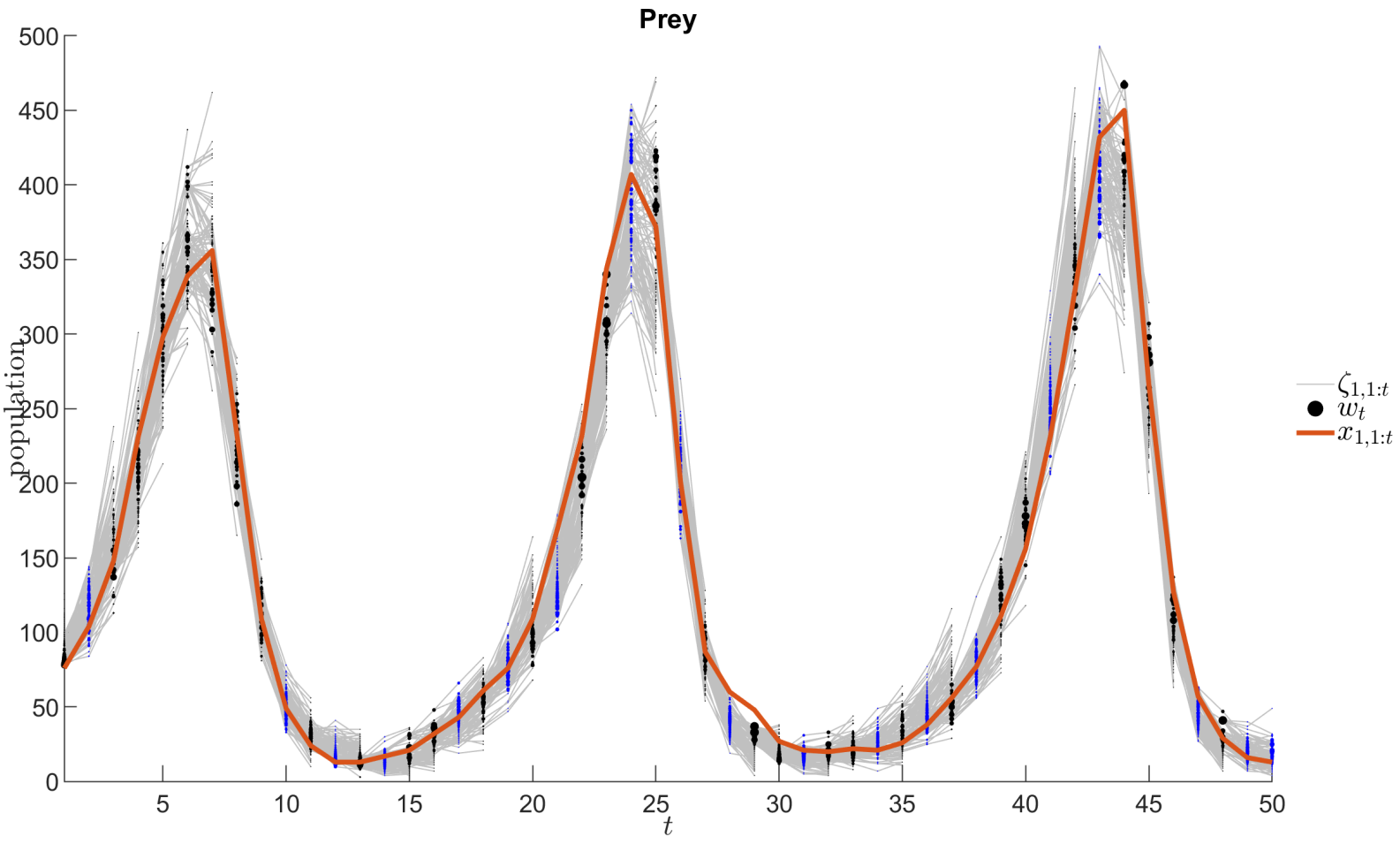


Beyond the basics

Adaptive Resampling

- Resampling helps with particle degeneracy, but you lose diversity of the samples
- It's usually unnecessary to resample at each iteration
- The goal of resampling is to remove particles with very low weights
- Can use the estimated ESS ($1/\sum_{i=1}^{N}(W_t^i)^2$) to determine when resampling will be helpful
- Only resample when the ESS falls below some threshold, e.g. $N/2$

Lotka-Volterra Example



By-passed the resampling step 18 times

Different Proposals

Can use other functions to predict $x_t \mid x_{t-1} \rightarrow$ it doesn't have to be the transition density

Currently, the proposal is

$$p(x_{1:t} \mid y_{1:t-1}) = p(x_t \mid x_{t-1}) p(x_{1:t-1} \mid y_{1:t-1})$$

Might get a better result if the proposal also depends on y_t

$$q(x_{1:t} \mid y_{1:t}) = q(x_t \mid x_{t-1}, y_t) p(x_{1:t-1} \mid y_{1:t-1})$$

Different Proposals

At iteration t:

Different Proposals

At iteration t:

The target is still the joint filtering distribution (posterior of $x_{1:t}$)

$$p(x_{1:t} \mid y_{1:t}) \propto \underbrace{p(y_{1:t} \mid x_{1:t})}_{\text{likelihood}} \underbrace{p(x_{1:t})}_{\text{state prior}}$$

Different Proposals

At iteration t:

The target is still the joint filtering distribution (posterior of $x_{1:t}$)

$$p(x_{1:t} \mid y_{1:t}) \propto \underbrace{p(y_{1:t} \mid x_{1:t})}_{\text{likelihood}} \underbrace{p(x_{1:t})}_{\text{state prior}}$$

The proposal is

$$q(x_{1:t} \mid y_{1:t}) = q(x_t \mid x_{t-1}, y_t) p(x_{1:t-1} \mid y_{1:t-1})$$

Different Proposals

At iteration t:

The target is still the joint filtering distribution (posterior of $x_{1:t}$)

$$p(x_{1:t} \mid y_{1:t}) \propto \underbrace{p(y_{1:t} \mid x_{1:t})}_{\text{likelihood}} \underbrace{p(x_{1:t})}_{\text{state prior}}$$

The proposal is

$$q(x_{1:t} \mid y_{1:t}) = q(x_t \mid x_{t-1}, y_t) p(x_{1:t-1} \mid y_{1:t-1})$$

The weights are

$$\begin{aligned} w_t &= \frac{p(x_{1:t}, y_{1:t})}{q(x_{1:t}, y_{1:t})} = \frac{p(x_{1:t}, y_{1:t})}{q(x_t \mid x_{t-1}, y_t) p(x_{1:t-1}, y_{1:t-1})} \\ &= \frac{p(y_t \mid x_t) p(x_t \mid x_{t-1})}{q(x_t \mid x_{t-1}, y_t)}, \quad x_{t-1:t} \sim q(\bullet \mid y_{1:t}) \end{aligned}$$

Final Remarks

We've covered the bootstrap particle filter (Gordon et al., 1993) with multinomial resampling, which has a nice correspondence to the filtering problem.

Output

- weighted sample from $p(x_{1:n} \mid y_{1:n})$
- unbiased estimate of the marginal likelihood $p(y_{1:n})$
- samples marginally distributed according to the filtering distribution at every iteration t

For improved efficiency

- adaptive resampling
- using a different proposal (no longer the bootstrap filter)
- different resampling strategies

Further Reading i

Chopin, N. and Papaspiliopoulos, O. (2020). *An introduction to sequential Monte Carlo*. Springer International Publishing.

Doucet, A., Johansen, A. M., et al. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F Radar and Signal Processing*, 140(2):107.

Further Reading ii

Naesseth, C. A., Lindsten, F., and Schön, T. B. (2019). Elements of Sequential Monte Carlo. *Foundations and Trends® in Machine Learning*, 12(3):307–392.

Schön, T. B. and Lindsten, F. (2017). Learning of dynamical systems–Particle filters and Markov chain methods. *Draft available*.