V. Nonlinear state-space models

A. Extended Kalman filter
Linear state-space model:

**State equation:**

\[
\xi_{t+1} = F \xi_t + v_{t+1} \quad v_{t+1} \sim N(0, Q)
\]

**Observation equation:**

\[
y_t = A' x_t + H' \xi_t + w_t \quad w_t \sim N(0, R)
\]
Nonlinear state-space model:

State equation:
\[
\xi_{t+1} = \phi(\xi_t) + v_{t+1} \quad v_{t+1} \sim N(0, Q)
\]

Observation equation:
\[
y_t = a(x_t) + h(\xi_t) + w_t \quad w_t \sim N(0, R)
\]
Suppose at date $t$ we have approximation to distribution of $\xi_t$ conditional on

$$\Omega_t = \{y_t, y_{t-1}, \ldots, y_1, x_t, x_{t-1}, \ldots, x_1\}$$

$$\xi_t | \Omega_t \sim N(\widehat{\xi}_{t|t}, P_{t|t})$$

goal: calculate $\widehat{\xi}_{t+1|t+1}, P_{t+1|t+1}$
State equation:

\[ \xi_{t+1} = \phi(\xi_t) + \mathbf{v}_{t+1} \]

\[ \phi(\xi_t) \approx \phi_t + \Phi_t (\xi_t - \hat{\xi}_{t|t}) \]

\[ \phi_t = \phi(\hat{\xi}_{t|t}) \]

\[ \Phi_t = \frac{\partial \phi(\xi_t)}{\partial \xi_t'} \bigg|_{\xi_t = \hat{\xi}_{t|t}} \]
Forecast of state vector:

\[ \xi_{t+1} = \phi_t + \Phi_t (\xi_t - \hat{\xi}_{t|t}) + v_{t+1} \]

\[ \hat{\xi}_{t+1|t} = \phi_t = \phi(\hat{\xi}_{t|t}) \]

\[ P_{t+1|t} = \Phi_t P_{t|t} \Phi_t' + Q \]
Observation equation:

\[ y_t = a(x_t) + h(\xi_t) + w_t \]

\[ h(\xi_t) \approx h_t + H_t'(\xi_t - \hat{\xi}_{t|t-1}) \]

\[ h_t = h(\hat{\xi}_{t|t-1}) \]

\[ H_t' = \frac{\partial h(\xi_t)}{\partial \xi_t} \bigg|_{\xi_t = \hat{\xi}_{t|t-1}} \]

Note \( x_t \) is observed so no need to linearize \( a(x_t) \)
Approximating state equation:
\[ \xi_{t+1} = \phi_t + \Phi_t(\xi_t - \hat{\xi}_{t|t}) + v_{t+1} \]

Approximating observation equation:
\[ y_t = a(x_t) + h_t + H'_t(\xi_t - \hat{\xi}_{t|t-1}) + w_t \]

A state-space model with time-varying coefficients
Forecast of observation vector:

\[ y_{t+1} = a(x_{t+1}) + h_{t+1} + H'_{t+1}(\xi_{t+1} - \hat{\xi}_{t+1|t}) + w_{t+1} \]

\[ \hat{y}_{t+1|t} = a(x_{t+1}) + h_{t+1} \]

\[ = a(x_{t+1}) + h(\hat{\xi}_{t+1|t}) \]

\[ E(y_{t+1} - \hat{y}_{t+1|t})(y_{t+1} - \hat{y}_{t+1|t})' \]

\[ = H'_{t+1}P_{t+1|t}H_{t+1} + R \]
Updated inference:
\[ \hat{\xi}_{t+1|t+1} = \hat{\xi}_{t+1|t} + K_{t+1}(y_{t+1} - \hat{y}_{t+1|t}) \]
\[ K_{t+1} = P_{t+1|t}H_{t+1}(H'_{t+1}P_{t+1|t}H_{t+1} + R)^{-1} \]
Start from \( \hat{\xi}_{0|0} \) and \( P_{0|0} \) reflecting prior information
Approximate log likelihood:

\[- \frac{T_n}{2} \log 2\pi - \frac{1}{2} \sum_{t=1}^{T} \log|\Omega_t| \]

\[- \frac{1}{2} \sum_{t=1}^{T} \varepsilon_t' \Omega_t^{-1} \varepsilon_t \]

\[\Omega_t = H_t' P_{t|t-1} H_t + R \]

\[\varepsilon_t = y_t - a(x_t) - h(\hat{\xi}_{t|t-1}) \]
V. Nonlinear state-space models

A. Extended Kalman filter
B. Particle filter
State equation:
\[ \xi_{t+1} = \phi_t(\xi_t, v_{t+1}) \]
\[ r \times 1 \quad r \times 1 \]

Observation equation:
\[ y_t = h_t(\xi_t, w_t) \]
\[ n \times 1 \quad n \times 1 \]

\( \phi_t(\cdot) \) and \( h_t(\cdot) \) known functions
(may depend on unknown \( \theta \))
\{\( w_t, v_t \)\} have known distribution (e.g., i.i.d., perhaps depend on \( \theta \))
\[ \Omega_t = \{y_t, y_{t-1}, \ldots, y_1\} \]

\[ \Lambda_t = \{\xi_t, \xi_{t-1}, \ldots, \xi_0\} \]

output for step \( t \):

\[ p(\Lambda_t|\Omega_t) \]

represented by a series of particles:

\[ \{\xi^{(i)}_t, \xi^{(i)}_{t-1}, \ldots, \xi^{(i)}_0\} \] \( i=1 \ldots D \)
Particle $i$ is associated with weight $\hat{\omega}_t^{(i)}$ such that particles can be used to simulate draw from $p(\Lambda_t|\Omega_t)$, e.g.

$$E(\xi_{t-1}|\Omega_t) = \sum_{i=1}^{D} \xi_{t-1}^{(i)} \hat{\omega}_t^{(i)}$$
Output of step $t + 1$:

$$p(\Lambda_{t+1}|\Omega_{t+1})$$

keep particles $\{\xi_t^{(i)}, \xi_{t-1}^{(i)}, \ldots, \xi_0^{(i)}\}_{i=1}^D$

append $\{\xi_{t+1}^{(i)}\}_{i=1}^D$ and recalculate weights $\hat{\omega}_{t+1}^{(i)}$

and as byproduct we get an estimate of

$$p(y_{t+1}|\Omega_t)$$
Method: Sequential Importance Sampling
At end of step $t$ have generated
$\Lambda_t^{(i)} = \{\xi_t^{(i)}, \xi_{t-1}^{(i)}, \ldots, \xi_0^{(i)}\}$
from some known importance density
$g_t(\Lambda_t|\Omega_t) = \tilde{g}_t(\xi_t|\Lambda_{t-1}, \Omega_t)g_{t-1}(\Lambda_{t-1}|\Omega_{t-1})$
We will also have calculated (up to a constant that does not depend on $\xi_t$) the true value of $p_t(\Lambda_t|\Omega_t)$ so weight for particle $i$ is proportional to

$$\omega_t^{(i)} = \frac{p_t(\Lambda_t^{(i)}|\Omega_t)}{g_t(\Lambda_t^{(i)}|\Omega_t)}$$
\[ \omega_t^{(i)} = \frac{p_t(\Lambda_t^{(i)} | \Omega_t)}{g_t(\Lambda_t^{(i)} | \Omega_t)} \]

\textbf{Step } t + 1:\

\[ p_{t+1}(\Lambda_{t+1} | \Omega_{t+1}) = \frac{p(y_{t+1} | \xi_{t+1})p(\xi_{t+1} | \xi_t)p_t(\Lambda_t | \Omega_t)}{p(y_{t+1} | \Omega_t)} \]

\[ \propto p(y_{t+1} | \xi_{t+1}) p(\xi_{t+1} | \xi_t) p_t(\Lambda_t | \Omega_t) \]

-known from obs eq known from state eq known at } t
\[
\omega_{t+1}^{(i)} = \frac{p_t(L_{t+1}^{(i)}|\Omega_{t+1})}{g_t(L_{t+1}^{(i)}|\Omega_{t+1})} \\
\propto \frac{p(y_{t+1}|\xi_{t+1}^{(i)})p(\xi_{t+1}^{(i)}|\xi_t^{(i)})p_t(L_t^{(i)}|\Omega_t)}{g_t+1(\xi_{t+1}^{(i)}|\Lambda_t^{(i)},\Omega_{t+1})g_t(L_t^{(i)}|\Omega_t)} \\
= \frac{p(y_{t+1}|\xi_{t+1}^{(i)})p(\xi_{t+1}^{(i)}|\xi_t^{(i)})}{g_t+1(\xi_{t+1}^{(i)}|\Lambda_t^{(i)},\Omega_{t+1})} \frac{p_t(L_t^{(i)}|\Omega_t)}{g_t(L_t^{(i)}|\Omega_t)} \\
= \tilde{\omega}_{t+1}^{(i)} \omega_t^{(i)}
\]
\[ \hat{\omega}_t^{(i)} = \frac{\omega_t^{(i)}}{\sum_{i=1}^{D} \omega_t^{(i)}} \]

\[ \hat{E}(\xi_{t-1}|\Omega_t) = \sum_{i=1}^{D} \hat{\omega}_t^{(i)} \xi_{t-1}^{(i)} \]

\[ \hat{P}(\xi_{1,t} > 0|\Omega_t) = \sum_{i=1}^{D} \hat{\omega}_t^{(i)} \delta[\xi_{1t}>0] \]
\[ \tilde{\omega}_{t+1}^{(i)} = \frac{p(y_{t+1} | \xi_{t+1}^{(i)}) p(\xi_{t+1}^{(i)} | \xi_t^{(i)})}{\tilde{g}_{t+1}(\xi_{t+1}^{(i)} | \Lambda_t^{(i)}, \Omega_{t+1})} \]

\[ \hat{p}(y_{t+1} | \Omega_t) = \sum_{i=1}^D \tilde{\omega}_{t+1}^{(i)} \tilde{\omega}_t^{(i)} \]

\[ \hat{\mathcal{L}}(\theta) = \sum_{t=1}^T \log \hat{p}(y_t | \Omega_{t-1}) \]

Classical: choose \( \theta \) to max \( \hat{\mathcal{L}}(\theta) \)

Bayesian: draw \( \theta \) from posterior which is proportional to \( p(\theta) \exp[\hat{\mathcal{L}}(\theta)] \)
How start algorithm for $t = 0$?

Draw $\xi_0^{(i)}$ from $p(\xi_0)$

(prior distribution or hypothesized unconditional distribution)
How choose importance density $\tilde{g}_{t+1}(\xi_{t+1} | \Lambda_t, \Omega_{t+1})$?

(1) Bootstrap filter

$\tilde{g}_{t+1}(\xi_{t+1} | \Lambda_t, \Omega_{t+1}) = p(\xi_{t+1} | \xi_t)$

known from state equation

$\xi_{t+1} = \phi_t(\xi_t, v_{t+1})$

But better performance from adaptive filters that also use $y_{t+1}$
Note that for bootstrap filter

\[ \tilde{g}_{t+1} (\xi_{t+1} | \Lambda_t, \Omega_{t+1}) = p(\xi_{t+1} | \xi_t) \]

\[ \tilde{\omega}_{t+1}^{(i)} = \frac{p(y_{t+1} | \xi_{t+1}^{(i)}) p(\xi_{t+1}^{(i)} | \xi_t^{(i)})}{\tilde{g}_{t+1} (\xi_{t+1}^{(i)} | \Lambda_t^{(i)}, \Omega_{t+1})} \]

\[ = p(y_{t+1} | \xi_{t+1}^{(i)}) \]
Separate problem for particle filter: one history $\Lambda_t^{(i)}$ comes to dominate the others ($\hat{\omega}_t^{(i)} \to 1$ for some $i$)
Partial solution to degeneracy problem:
Sequential Importance Sampling with Resampling
Before finishing step $t$, now resample
\[
\{\Lambda_t^{(j)}\}_{j=1}^{D}
\]
with replacement by drawing from the distribution

\[
\Lambda_t^{(j)} = \begin{cases} 
\Lambda_t^{(1)} & \text{with probability } \hat{\omega}_t^{(1)} \\
\vdots & \\
\Lambda_t^{(D)} & \text{with probability } \hat{\omega}_t^{(D)}
\end{cases}
\]
Result: repopulate $\{\Lambda_t^{(j)}\}$ by replicating most likely elements (weights for $\Lambda_t^{(j)}$ are now $\hat{\omega}_t^{*(j)} = 1/D$).
(1) Resampling does not completely solve degeneracy because early-sample elements of
\[
\Lambda_t^{(j)} = \{\xi_t^{(j)}, \xi_{t-1}^{(j)}, \ldots, \xi_0^{(j)}\}
\]
will tend to be the same for all \(j\) as \(t\) gets large.

(2) Does help in the sense that have full set of particles to grow from \(t\) forward
(3) Have good inference about $p(\xi_{t-k} | \Omega_t)$ for small $k$

(4) Have poor inference about $p(\xi_{t-k} | \Omega_t)$ for large $k$

(separate smoothing algorithm can be used if goal is $p(\xi_t | \Omega_T)$)
Summary of bootstrap particle filter with resampling:

(1) Get initial set of \( D \) particles for date \( t = 0 \)
   
   (a) Set \( \xi_{-100}^{(j)} = 0 \) for \( j = 1 \)

   (b) Generate \( \xi_t^{(j)} = \phi_0(\xi_{t-1}^{(j)}, v_t^{(j)}) \)

   for \( t = -99, -98, \ldots, 0 \)

   (c) Value of \( \xi_0^{(j)} \) is one value for particle \( j = 1 \) for date 0

   (d) repeat (a)-(c) for \( j = 1, \ldots, D \) to populate \( \{\xi_0^{(1)}, \xi_0^{(2)}, \ldots, \xi_0^{(D)}\} \)
For any given $\theta$ set $\ell_0(\theta) = 0$ and for each $t = 1, 2, \ldots, T$ we then do the following:

(2) Compute $\tilde{\omega}_t^{(i)} = p(y_t | \xi_t^{(i)})$ and update estimate of log likelihood:

$$
\ell_t(\theta) = \ell_{t-1}(\theta) + \log \left\{ D^{-1} \sum_{j=1}^{D} \tilde{\omega}_t^{(j)} \right\}
$$
(3) Resample particles:

(a) Calculate $\hat{\omega}^{*}_{t}(j) = \check{\omega}^{(j)}_{t} / \left\{ \sum_{j=1}^{D} \check{\omega}^{(j)}_{t} \right\}$

(b) Draw $u \sim U(0, 1)$ and define $u^{(j)} = (u/D) + (j - 1)/D$ for $j = 1, \ldots, D$.

(c) Find the indexes $i^{1}, \ldots, i^{D}$ such that $\sum_{k=1}^{i^{j}-1} \hat{\omega}^{*}_{t}(k) < u^{(j)} \leq \sum_{k=1}^{i^{j}} \hat{\omega}^{*}_{t}(k)$
(4) Generate new particles:

\[ \xi_t^{(j)} \text{ from } \phi_{t+1}(\xi_t^{i,j}, v_{t+1}^{(j)}) \]

Repeat (2)-(4) for \( t = 1, \ldots, T \).
What do we do with estimate of log likelihood $\ell_T(\theta)$?

Best approach: embed within random-walk Metropolis-Hastings to generate draws of $\theta$ from posterior $p(\theta|Y)$ using prior $p(\theta)$. 
(1) Generate initial draw \( \theta^{(m)} \) for \( m = 1 \) and calculate \( \ell_T(\theta^{(m)}) \) and \( p(\theta^{(m)}) \).

(2) Generate \( \tilde{\theta}^{(m+1)} \sim N(\theta^{(m)}, c\Lambda) \) and calculate \( \ell_T(\tilde{\theta}^{(m+1)}) \) and \( p(\tilde{\theta}^{(m+1)}) \).
(3) Set

\[
\theta^{(m+1)} = \begin{cases} 
\tilde{\theta}^{(m+1)} & \text{with prob } \alpha \\
\theta^{(m)} & \text{with prob } 1 - \alpha
\end{cases}
\]

\[
\alpha = \min \left\{ \frac{\ell_T(\tilde{\theta}^{(m+1)})p(\tilde{\theta}^{(m+1)})}{\ell_T(\theta^{(m)})p(\theta^{(m)})}, 1 \right\}.
\]
Also possible to improve a lot on particle bootstrap by using better proposal density. Example: use extended Kalman filter for proposal density in place of generating $\xi_{t+1}^{(j)}$ from $\phi_t(\xi_t^{(j)}, v_{t+1})$. 