Sequential learning in dynamic models: Particle filters

Hedibert F. Lopes

April 23rd 2026

Normal Dynamic Linear Model (NDLM)

Suppose that observations y_t is modeled as follows:

$$y_t = F_t^\top \theta_t + v_t, \quad v_t \sim \mathcal{N}(0, V_t), \tag{1}$$
$$\theta_t = G_t \theta_{t-1} + w_t, \quad w_t \sim \mathcal{N}(0, W_t). \tag{2}$$

Standard cases where closed-form inference about the states $\theta_{1:T}$ is availabe:

- Known $\{V_t, W_t\}$,
- W_t derived via discount factor,
- V_t learning via discount factor and variance law.

Online vs Offline Inference

$$\text{Offline: } p(\theta_t \mid y_{1:T}) \quad \text{(backward)} \tag{3}$$
$$\text{Online: } p(\theta_t \mid y_{1:t}) \quad \text{(forward)} \tag{4}$$

Key Point

NDLM allows exact inference without MCMC.

When Parameters Are Unknown

Assume:

$$V_t = V, \quad W_t = W \quad \text{unknown}$$

Inference Problem

$$p(\theta_{1:T}, V, W \mid y_{1:T})$$

- Posterior is analytically intractable
- Need simulation-based methods

MCMC Solution

$$p(\theta_{1:T} \mid V, W, y_{1:T}) \quad (5)$$

$$p(V, W \mid \theta_{1:T}, y_{1:T}) \quad (6)$$

- Gibbs Sampling cycles iteratively through (5) and (6).
- Forward Filtering Backward Sampling (FFBS)
- Full conditionals tractable

Cost

Loss of sequential (online) learning

General Dynamic Model

$$y_t = f(\theta_t, v_t, \xi) \qquad (7)$$

$$\theta_t = g(\theta_{t-1}, w_t, \xi) \qquad (8)$$

- ξ: unknown static parameters
- f, g: known functions
- Possibly nonlinear / non-Gaussian

Sequential Monte Carlo (SMC)

Also known as Particle Filters.

Goal

Approximate:

$$p(\theta_t \mid y_{1:t})$$

- Represent distribution with particles
- Update sequentially as new data arrives

Recursive Structure

1. Posterior at $t-1$:

$$\{\theta_{t-1}^{(i)}\}_{i=1}^{M} \sim p(\theta_{t-1} \mid y_{1:(t-1)})$$

2. Prior at t (propagation step): For $i = 1, \ldots, M$

$$\tilde{\theta}_t^{(i)} = g(\theta_{t-1}^{(i)}, w_t, \xi) \sim p(\theta_t \mid \theta_{t-1}^{(i)})$$

3. Compute the re-sampling weights (SIR step):

$$\omega_t^{(i)} \propto p(y_t \mid \tilde{\theta}_t^{(i)})$$

4. Resampling step:

Sample from $\{\tilde{\theta}_t^{(i)}\}_{i=1}^{M}$ with weights $\{\omega_t^{(i)}\}_{i=1}^{M}$

5. Posterior at t:

$$\{\theta_t^{(i)}\}_{i=1}^{M} \sim p(\theta_t \mid y_{1:t})$$

Bootstrap Particle Filter

Key Idea

Use transition prior as proposal.

- Simple
- Widely used
- May suffer from degeneracy

Example: Local Level Model

$$y_t = \theta_t + v_t, \quad v_t \sim \mathcal{N}(0,1) \tag{9}$$

$$\theta_t = \theta_{t-1} + w_t, \quad w_t \sim \mathcal{N}(0,0.5) \tag{10}$$

Algorithm (Bootstrap Filter)

For $i = 1, \ldots, M$

Initialization at time $t = 0$:

$$\theta_0^{(i)} \sim \mathcal{N}(0, 1)$$

Step A: Propagate

$$\tilde{\theta}_1^{(i)} = \theta_0^{(i)} + v_1^{(i)}$$

Step B: Weight

$$w_1^{(i)} \propto p(y_1 \mid \tilde{\theta}_1^{(i)})$$

Step C: Resample: For $j = 1, \ldots, M$

$$\theta_1^{(j)} \sim \{\tilde{\theta}_1^{(i)}\}_{i=1}^{M} \text{ with weights } \{\tilde{\theta}_1^{(i)}\}_{i=1}^{M}$$

Key Takeaways

- NDLM $\Rightarrow$ exact inference (Kalman Filter)
- Unknown parameters $\Rightarrow$ MCMC
- Nonlinear / non-Gaussian $\Rightarrow$ Particle Filters
- SMC enables **online Bayesian learning**