Particle Filtering: Solutions

ELG 5218 – Uncertainty Evaluation in Engineering Measurements and Machine Learning

Instructor: Miodrag Bolić, University of Ottawa

1 Resampling Techniques

Problem 1: Multinomial Resampling

Given M particles $\{\mathbf{z}^{(i)}, \tilde{w}^{(i)}\}_{i=1}^{M}$ where $\sum_i \tilde{w}^{(i)} = 1$:

1. Explain how multinomial resampling works.
2. If a particle has normalized weight $\tilde{w}^{(i)} = 0.2$ and we draw $M = 100$ resampled particles, what is the expected number of copies?
3. Show that the resampled distribution $\bar{p}(\mathbf{z}) = \sum_{i=1}^{M} \frac{N^{(i)}}{M} \delta(\mathbf{z} - \mathbf{z}^{(i)})$ is an unbiased approximation.

Solution:

1. **How Multinomial Resampling Works:**

   Each particle $\mathbf{z}^{(i)}$ is treated as a category in a multinomial distribution with parameter $\tilde{w}^{(i)}$. We draw M samples (with replacement) from this categorical distribution:

   - Each draw selects particle i with probability $\tilde{w}^{(i)}$
   - Let $N^{(i)}$ = number of times particle i is selected
   - $(N^{(1)}, \ldots, N^{(M)}) \sim \text{Multinomial}(M, \tilde{w}^{(1)}, \ldots, \tilde{w}^{(M)})$
   - The new population consists of $N^{(i)}$ copies of particle $\mathbf{z}^{(i)}$

   Result: High-weight particles are replicated multiple times; low-weight particles are likely discarded.

2. **Expected Number of Copies:** 照乘：Weight X M copies = Mean

   $$\mathbb{E}[N^{(i)}] = M \cdot \tilde{w}^{(i)} = 100 \times 0.2 = 20 \text{ expected copies}$$

   The actual realization is random: $N^{(i)} \sim \text{Binomial}(M, \tilde{w}^{(i)})$ marginally.

3. **Unbiasedness:** The resampled distribution is:

   $$\bar{p}(\mathbf{z}) = \sum_{i=1}^{M} \frac{N^{(i)}}{M} \delta(\mathbf{z} - \mathbf{z}^{(i)})$$

Taking expectation:

$$\begin{aligned}
\mathbb{E}[\bar{p}(\mathbf{z})] &= \sum_{i=1}^{M} \mathbb{E}\left[\frac{N^{(i)}}{M}\right]\delta(\mathbf{z}-\mathbf{z}^{(i)}) \\
&= \sum_{i=1}^{M} \frac{\mathbb{E}[N^{(i)}]}{M}\delta(\mathbf{z}-\mathbf{z}^{(i)}) \\
&= \sum_{i=1}^{M} \frac{M\tilde{w}^{(i)}}{M}\delta(\mathbf{z}-\mathbf{z}^{(i)}) \\
&= \sum_{i=1}^{M} \tilde{w}^{(i)}\delta(\mathbf{z}-\mathbf{z}^{(i)}) = \hat{p}(\mathbf{z})
\end{aligned}$$

Thus $\bar{p}$ is an unbiased approximation of the original discrete distribution.

Problem 2: State-Space Models and Optimal Proposal

For nonlinear state-space models:

Latent State Equation: $\mathbf{z}_k = f_z(\mathbf{z}_{k-1}, \mathbf{u}_k)$
Observation Equation: $\mathbf{y}_k = f_y(\mathbf{z}_k, \mathbf{v}_k)$
with densities:

- $p(\mathbf{z}_k|\mathbf{z}_{k-1})$: state transition
- $p(\mathbf{y}_k|\mathbf{z}_k)$: observation likelihood

1. What is the optimal proposal distribution $q^{\text{opt}}(\mathbf{z}_k|\mathbf{z}_{k-1}, \mathbf{y}_k)$?
2. What are the resulting weights?
3. When is this optimal proposal tractable?
4. Suggest practical alternatives when intractable.

Solution:

1. **Optimal Proposal:**

   From importance sampling theory (Problem 1), the optimal proposal minimizes weight variance. For sequential problems, this is:

   $$q^{\text{opt}}(\mathbf{z}_k|\mathbf{z}_{1:k-1}, \mathbf{y}_{1:k}) = p(\mathbf{z}_k|\mathbf{z}_{1:k-1}, \mathbf{y}_{1:k})$$

   Using the Markov structure of the state-space model (latent state depends only on previous state, observations depend only on current state):

   $$q^{\text{opt}}(\mathbf{z}_k|\mathbf{z}_{k-1}, \mathbf{y}_k) = p(\mathbf{z}_k|\mathbf{z}_{k-1}, \mathbf{y}_k)$$

   This incorporates both the state dynamics and the current observation.

2. **Optimal Weights:**

   Using $\phi_k(\mathbf{z}_{1:k}) = \prod_{\ell=1}^{k} p(\mathbf{y}_\ell|\mathbf{z}_\ell)p(\mathbf{z}_\ell|\mathbf{z}_{\ell-1})$ with the optimal proposal:

$$\begin{aligned} w_k &\propto \frac{p(\mathbf{y}_k|\mathbf{z}_k)p(\mathbf{z}_k|\mathbf{z}_{k-1})}{p(\mathbf{z}_k|\mathbf{z}_{k-1}, \mathbf{y}_k)} \\ &= \frac{p(\mathbf{y}_k|\mathbf{z}_k)p(\mathbf{z}_k|\mathbf{z}_{k-1})}{p(\mathbf{y}_k|\mathbf{z}_k)p(\mathbf{z}_k|\mathbf{z}_{k-1})/p(\mathbf{y}_k|\mathbf{z}_{k-1})} \\ &= p(\mathbf{y}_k|\mathbf{z}_{k-1}) \end{aligned}$$

   So: $w_k(\mathbf{z}_{1:k}) = w_{k-1}(\mathbf{z}_{1:k-1})p(\mathbf{y}_k|\mathbf{z}_{k-1})$.

   Key insight: With the optimal proposal, weights depend only on the **previous** state, not the newly sampled state.

Problem 3: Particle Filter Algorithm - Prior Proposal

For state-space model with prior proposal $q(\mathbf{z}_k|\mathbf{z}_{k-1}, \mathbf{y}_k) = p(\mathbf{z}_k|\mathbf{z}_{k-1})$:

Particle Filter Algorithm (Prior Proposal):

For each particle $i = 1, \ldots, M$:

Step 1: Particle Generation

$$\mathbf{z}_k^{(i)} \sim p(\mathbf{z}_k|\mathbf{z}_{k-1}^{(i)})$$

Step 2a: Weight Computation

$$w_k^{*(i)} = \tilde{w}_{k-1}^{(i)} \cdot p(\mathbf{y}_k|\mathbf{z}_k^{(i)})$$

Step 2b: Weight Normalization

$$\tilde{w}_k^{(i)} = \frac{w_k^{*(i)}}{\sum_{j=1}^{M} w_k^{*(j)}}$$

Step 3: Estimate

$$\widehat{\mathbb{E}}[g(\mathbf{z}_k)|\mathbf{y}_{1:k}] \approx \sum_{i=1}^{M} \tilde{w}_k^{(i)} g(\mathbf{z}_k^{(i)})$$

Solution:

For a 1D random walk example:

Model:

$$\begin{aligned} \mathbf{z}_k &= \mathbf{z}_{k-1} + \mathbf{u}_k, \quad \mathbf{u}_k \sim \mathcal{N}(0, 1) \\ \mathbf{y}_k &= \mathbf{z}_k + \mathbf{v}_k, \quad \mathbf{v}_k \sim \mathcal{N}(0, 1) \end{aligned}$$

Densities:

$$\begin{aligned} p(\mathbf{z}_k|\mathbf{z}_{k-1}) &= \mathcal{N}(\mathbf{z}_k|\mathbf{z}_{k-1}, 1) \\ p(\mathbf{y}_k|\mathbf{z}_k) &= \mathcal{N}(\mathbf{y}_k|\mathbf{z}_k, 1) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(\mathbf{y}_k - \mathbf{z}_k)^2}{2}\right) \end{aligned}$$

Problem 4: Effective Sample Size

The effective number of samples is:

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{M} (\tilde{w}^{(i)})^2}$$

1. When is $N_{\text{eff}} = M$ (maximum)?
2. When is $N_{\text{eff}} = 1$ (worst)?
3. Why trigger resampling at $N_{\text{eff}} < M/2$?

Solution:

1. **Maximum $N_{\text{eff}} = M$:**

   Occurs when all weights are equal: $\tilde{w}^{(i)} = 1/M$ for all i.

   $$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{M} (1/M)^2} = \frac{1}{M \cdot 1/M^2} = M$$

   All particles contribute equally to the posterior estimate.

2. **Minimum $N_{\text{eff}} = 1$:**

   Occurs when one particle has weight 1 and all others have weight 0:

   $$N_{\text{eff}} = \frac{1}{1^2 + 0 + \cdots + 0} = 1$$

   Only one particle carries information; all others are useless.

3. **Resampling Threshold $N_{\text{eff}} < M/2$:**

   This threshold balances two competing objectives:

   - **Prevent Degeneracy:** If $N_{\text{eff}} < M/2$, then weight concentration is severe (too much variance). Resampling discards particles in low-probability regions.
   - **Preserve Diversity:** If resampling too frequently (higher threshold), we lose particle diversity unnecessarily. The threshold $M/2$ is empirically found to be near-optimal.
   - **Adaptive:** Don't resample every step (wastes diversity). Only resample when needed.

   Why $M/2$: Empirical experience shows this threshold provides best balance between tracking accuracy and computational efficiency. Too early resampling increases sample impoverishment; too late resampling allows weight degeneracy to worsen.

2 Practical Applications

Problem 5: Bearings-Only Tracking

Problem: A passive observer (e.g., sonar) measures only bearing angles to a moving target, not range. Goal: Estimate 2D target position and velocity.

Latent State: $\mathbf{z}_k = [s_{x,k}, V_{x,k}, s_{y,k}, V_{y,k}]^\top$ (position and velocity)

Latent State Equation:

$$\mathbf{z}_{k+1} = F\mathbf{z}_k + \mathbf{u}_k$$

where:

$$F = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and $\mathbf{u}_k \sim \mathcal{N}(0, Q)$ (process noise accounting for unmodeled acceleration, wind, etc.)

Observation Equation:

$$\mathbf{y}_k = f_y(\mathbf{z}_k, \mathbf{v}_k) = \arctan\left(\frac{s_{y,k}}{s_{x,k}}\right) + \mathbf{v}_k$$

where $\mathbf{v}_k \sim \mathcal{N}(0, R)$ is measurement noise (bearing uncertainty).

Observation likelihood:

$$p(\mathbf{y}_k|\mathbf{z}_k) = \mathcal{N}\left(\mathbf{y}_k \,\middle|\, \arctan(s_{y,k}/s_{x,k}), R\right)$$

Why Kalman Filter Fails:

- **Nonlinear Observation:** The function $f_y(\mathbf{z}_k) = \arctan(s_{y,k}/s_{x,k})$ is highly nonlinear (not a linear function of $\mathbf{z}_k$)
- **Gaussian Assumption Violated:** If we pass a Gaussian distribution on position through arctan, the resulting distribution on bearing is NOT Gaussian
- **Standard Kalman Limitation:** Kalman filter assumes:
  - Linear dynamics: $\mathbf{z}_{k+1} = F\mathbf{z}_k + \mathbf{u}_k$ (satisfied)
  - Linear observations: $\mathbf{y}_k = H\mathbf{z}_k + \mathbf{v}_k$ (violated due to arctan)
  - Gaussian noise: Both $\mathbf{u}_k, \mathbf{v}_k$ Gaussian (satisfied)
- **Posterior Non-Gaussian:** Even starting with Gaussian prior on position, the posterior given bearing measurements is non-Gaussian. Kalman's Gaussian updates are suboptimal.

Particle Filter Solution:

Apply the algorithm from Problem 8:

1. **Initialization:** Sample $M = 100$ particles: $\mathbf{z}_0^{(i)} \sim \mathcal{N}(\mathbf{z}_0^{\text{prior}}, P_0)$
2. **For each time step k:**
   (a) **Prediction:** $\mathbf{z}_k^{(i)} = F\mathbf{z}_{k-1}^{(i)} + \mathbf{u}_k^{(i)}$ where $\mathbf{u}_k^{(i)} \sim \mathcal{N}(0, Q)$
   (b) **Weight:** $w_k^{*(i)} = \tilde{w}_{k-1}^{(i)} \cdot \mathcal{N}(\mathbf{y}_k | \arctan(s_{y,k}^{(i)}/s_{x,k}^{(i)}), R)$
   (c) **Normalize:** $\tilde{w}_k^{(i)} = w_k^{*(i)} / \sum_j w_k^{*(j)}$
   (d) **Estimate:** $\hat{\mathbf{z}}_k = \sum_i \tilde{w}_k^{(i)} \mathbf{z}_k^{(i)}$
   (e) **Resample:** If $N_{\text{eff}} < 50$, use Algorithm 4 (systematic resampling)

Advantages:

- Handles nonlinearity exactly (no linearization approximation)
- Posterior naturally non-Gaussian (represents multimodal uncertainty)
- No divergence issues (unlike EKF)
- Can estimate full posterior, not just mean and covariance

Problem 6: Visual Object Tracking with Non-Gaussian Noise

Latent State Equation:

$$\mathbf{z}_k = \mathbf{z}_{k-1} + \mathbf{u}_k, \quad \mathbf{u}_k \sim \mathcal{U}([-10, 10]^2)$$

where $\mathbf{z}_k = [x_k, y_k]^\top$ is 2D object position.

Observation Equation:

$$\mathbf{y}_k = \mathbf{z}_k + \mathbf{v}_k, \quad \mathbf{v}_k \sim \mathcal{N}(0, \Sigma_v), \quad \Sigma_v = \begin{bmatrix} 2 & 0.5 \\ 0.5 & 2 \end{bmatrix}$$

Why Kalman Filter Suboptimal:

Kalman filter is optimal only for LINEAR dynamics with GAUSSIAN noise on both state and observation equations:

- **Linear dynamics:** $\mathbf{z}_{k+1} = F\mathbf{z}_k + \mathbf{u}_k$ (satisfied)
- **Linear observations:** $\mathbf{y}_k = H\mathbf{z}_k + \mathbf{v}_k$ (satisfied with $H = I$)
- **Gaussian process noise:** $\mathbf{u}_k \sim \mathcal{N}(0, Q)$ (VIOLATED: $\mathbf{u}_k$ is uniform!)
- **Gaussian measurement noise:** $\mathbf{v}_k \sim \mathcal{N}(0, R)$ (satisfied)

Under uniform process noise, the posterior $p(\mathbf{z}_k|\mathbf{y}_{1:k})$ is NOT Gaussian. Kalman's assumption of Gaussian distributions breaks down, making it suboptimal.

Particle Filter Design:

Use prior proposal from Problem 8:

1. **Particle Generation:** Sample from uniform distribution:
$$\mathbf{u}_k^{(i)} \sim \mathcal{U}([-10, 10]^2), \quad \mathbf{z}_k^{(i)} = \mathbf{z}_{k-1}^{(i)} + \mathbf{u}_k^{(i)}$$
2. **Weight Update:**
$$w_k^{*(i)} = \tilde{w}_{k-1}^{(i)} \cdot \mathcal{N}(\mathbf{y}_k|\mathbf{z}_k^{(i)}, \Sigma_v)$$
where the Gaussian likelihood is:
$$\mathcal{N}(\mathbf{y}_k|\mathbf{z}_k^{(i)}, \Sigma_v) = \frac{1}{2\pi\sqrt{\det(\Sigma_v)}} \exp\left(-\frac{1}{2}(\mathbf{y}_k - \mathbf{z}_k^{(i)})^\top \Sigma_v^{-1} (\mathbf{y}_k - \mathbf{z}_k^{(i)})\right)$$
3. **Normalization and Resampling:** Standard steps from Problem 8

Comparison: With vs Without Resampling

- **With resampling:**
  - Fewer particles have significant weight (concentrated on likely region)
  - Computational efficiency: fewer useless particles
  - Better tracking quality over long horizons
  - Trade-off: some loss of diversity
- **Without resampling:**
  - More particles retain nonzero weight initially
  - Wasted computation on unlikely particles
  - Tracking degrades as time increases (weight degeneracy)
  - Advantage: preserves full diversity of prior
  - Not practical for long sequences

Recommendation: Always use resampling with threshold $N_{\text{eff}} < M/2$.

Problem 7: Degeneracy vs Sample Impoverishment

Definitions:

1. **Weight Degeneracy:**

   As time progresses, the weight distribution becomes increasingly skewed. After many steps:

   - Few particles (e.g., 1-5 out of $M = 100$) accumulate nearly all weight
   - Most particles have negligible weight ($\tilde{w}^{(i)} < 0.01$)
   - Computation is wasted on irrelevant particles
   - Posterior approximation quality suffers (small effective sample size)

   Cause: Weights are products of likelihoods; those are typically < 1, so $w_k \approx L^k$ decays exponentially.

2. **Sample Impoverishment:**

   After resampling, the particle population loses diversity:

   - High-weight particles are replicated many times
   - Posterior approximation becomes coarse (many identical particles)
   - Effective sample size drops due to redundancy
   - Affects future predictions: correlated particles lead to poor uncertainty estimates

   Cause: Resampling concentrates particles on high-probability regions; new diversity only emerges through noise in state evolution.

Mechanism of Impoverishment:
Suppose at time $k-1$ one particle has weight 0.9 and we resample with $M = 100$:

- This particle is drawn ≈ 90 times
- Others divided among remaining 10 draws
- At time k: population has ~ 90 copies of one particle, plus scattered copies of others
- If process noise is small, all 90 copies evolve nearly identically
- Posterior $p(\mathbf{z}_k|\mathbf{y}_{1:k})$ becomes concentrated at discrete points (not smooth)
- For estimation, we effectively have $\ll 100$ independent samples

When Severe:
Sample impoverishment is worst when **process noise is small** (or near-zero):

- Small Q means state evolution is nearly deterministic
- Resampled particles don't diverge during prediction
- Diversity doesn't recover
- Over multiple steps, population collapses to few distinct values

Example: $\mathbf{z}_k = \mathbf{z}_{k-1} + \mathbf{u}_k$ with $\mathbf{u}_k \sim \mathcal{N}(0, 0.001)$ (tiny noise)
After resampling once, 90 copies of a single particle stay nearly identical through many steps.
Solutions:

- **Regularized Particle Filters:** Instead of resampling to discrete points, approximate posterior with smooth kernels:

$$p(\mathbf{z}|\mathbf{y}) \approx \sum_{i=1}^{M} \tilde{w}^{(i)} K(\mathbf{z} - \mathbf{z}^{(i)})$$

  where K is a smooth kernel (e.g., Gaussian). Maintains continuous support.

- **Better Proposal:** Using observation-dependent proposal (Problem 7) places particles in likely regions without pure resampling-induced collapse.

- **Jittering:** Add small random noise to resampled particles to encourage diversity.

- **Increase Process Noise Estimate:** If you have control over Q, larger Q helps particles diverge after resampling.