Particle Filters

M. Sami Fadali
Professor of EE
University of Nevada

Outline

- Monte Carlo integration.
- Particle filter.
- Importance sampling.
- Degeneracy
- Resampling
- Example.

Monte Carlo Integration

- Numerical integration using randomly generated points.
- No "curse of dimensionality"
- Use to calculate moments (integrals)
- Integral of $g(x)$, continuous $[a, b]$

$$\int_a^b g(x)dx = (b-a)\int_a^b \frac{1}{b-a} g(x)dx$$
$$= (b-a)E\{g(X)\}, X \sim U[a, b]$$

MC Integration Procedure

- Generate $X_i \sim U[a, b], i = 1, \ldots, N$
- Calculate $Y_i = g(X_i), i = 1, \ldots, N$
- Calculate the estimate of $E\{g(X)\}$

$$\bar{Y} = \frac{1}{N}\sum_{i=1}^{N} Y_i$$

- Calculate the estimate of $\int_a^b g(x)dx$

$$(b-a)\bar{Y}$$

Properties of $\bar{Y}$

- Use $X_i \sim U[a,b], i = 1, \dots, N$
- Unbiased

$$E\{\bar{Y}\} = \frac{1}{N}\sum_{i=1}^{N} E\{g(X_i)\} = E\{g(X)\}$$

- Consistent (shown earlier)

$$Var\{\bar{Y}\} = \frac{1}{N} Var\{g(X)\}$$

Example (Hogg et al., p. 289)

- Estimate π using Monte Carlo integration
- Use $g(X) = 4\sqrt{1-x^2}$

$$\int_0^1 g(x)dx = 2x\sqrt{1-x^2} + 2\sin^{-1}(x)\Big]_0^1 = \pi$$

- Generate $X_i \sim U[0,1], i = 1, \dots, N$
- Calculate $Y_i = 4\sqrt{1 - X_i^2}, i = 1, \dots, N$
- $\bar{Y}$ = consistent estimate of π (estimate improves as N increases)

MC Integration

- Factorization

$$g(x) = f(x)p(x), x \in [a,b]$$

$p(x)$ = pdf (integral is unity, nonnegative)

- Generate $X_i \sim p(x), i = 1, \dots, N$
- Calculate $Y_i = f(X_i), i = 1, \dots, N$
- Calculate the estimate of $E\{f(X)\} = \int_a^b g(x)dx$

$$\bar{Y} = \frac{1}{N}\sum_{i=1}^{N} Y_i$$

Importance Sampling

- Not always possible to generate samples from $p(x)$
- Generate samples from a similar pdf $q(x)$

$$p(x) > 0 \Rightarrow q(x) > 0, \qquad \forall x \in \mathcal{R}^n$$

(same support)

$$\int_a^b g(x)dx = \int_a^b f(x)p(x)dx$$

$$= \int_a^b f(x)\frac{p(x)}{q(x)} q(x)dx$$

Use MC Integration

- Generate $X_i \sim q(x), i = 1, \dots, N$
- Calculate $Y_i = f(X_i)W(X_i), i = 1, \dots, N$

$$W(X_i) = \frac{p(X_i)}{q(X_i)} \Big/ \sum_{i=1}^{N} \frac{p(X_i)}{q(X_i)}$$

$$\bar{Y} = \frac{1}{N}\sum_{i=1}^{N} Y_i = \frac{1}{N}\sum_{i=1}^{N} f(X_i)W(X_i)$$

Relation to Estimation

- Fundamental theorem of estimation theory
- Minimum mean-square error estimator:

$$\hat{\boldsymbol{x}}(k) = E\{\boldsymbol{x}(k)|\boldsymbol{z}^*(k)\}$$
$$= \int \boldsymbol{x}(k) p_{x|z}(\boldsymbol{x}(k)|\boldsymbol{z}^*(k)) d\boldsymbol{x}(k)$$
$$\boldsymbol{z}^*(k) = col\{\boldsymbol{z}(i), i = 0,1, \dots, k\}.$$

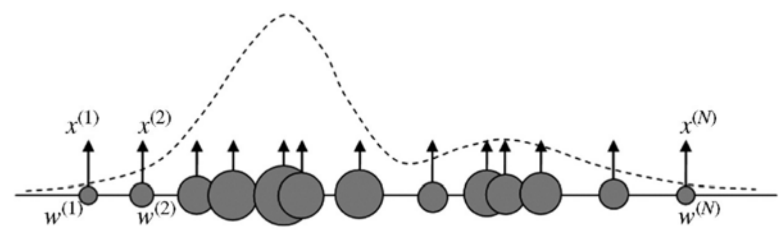
- $\boldsymbol{z}^*(k)$ = all measurements up to and including time k

Particle Filter

- View random samples as "particles"
- Can handle non-Gaussian statistics and multimodal distributions.
- Approximate a continuous pdf with appropriately weighted random samples.
- A posteriori conditional density

$$p\left(\boldsymbol{x}_k|\boldsymbol{z}_k^*\right) \approx \sum_{i=1}^{n} W_k^{(i)} \delta\left(\boldsymbol{x}_k - \boldsymbol{x}_k^{(i)}\right), \sum_{i=1}^{n} W_k^{(i)} = 1$$

Approximation of PDF Using Weighted Random Samples



$$p(x|z^*) \approx \sum_{i=1}^{N} \omega^{(i)} \delta(x - x^{(i)}) \quad \text{where} \sum_{i=1}^{N} w^{(i)} = 1$$

Figure 7.10

Model

- Nonlinear process model
$$\boldsymbol{x}_{k+1} = \phi(\boldsymbol{x}_k, \boldsymbol{w}_k)$$
- Measurement model
$$\boldsymbol{z}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{v}_k)$$
- Noise processes are not necessarily Gaussian.
- $\boldsymbol{z}_k^*$ =measurements up to and including time k

Recursive Bayesian Filter

- Predictor
$$p(\boldsymbol{x}_{k+1}|\boldsymbol{z}_k^*) = \int p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{z}_k^*)d\boldsymbol{x}_k$$
- Corrector
$$p(\boldsymbol{x}_k|\boldsymbol{z}_k^*) = \kappa_k p(\boldsymbol{z}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{z}_{k-1}^*)$$
- Initial condition $p(\boldsymbol{x}_0)$ for recursion
$$p(\boldsymbol{x}_k|\boldsymbol{z}_k^*) = \kappa_k p(\boldsymbol{z}_k|\boldsymbol{x}_k)\int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{k-1}^*)\,d\boldsymbol{x}_{k-1}$$

Corrector Modified

$$p(\boldsymbol{x}_k|\boldsymbol{z}_k^*) = \kappa_k p(\boldsymbol{z}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{z}_{k-1}^*)$$

- Rewrite using Markov property

$$p(\boldsymbol{x}_k|\boldsymbol{z}_k^*) = \kappa_k p(\boldsymbol{z}_k|\boldsymbol{x}_k)\int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{k-1}^*)\,d\boldsymbol{x}_{k-1}$$

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{k-1}^*) = \int p(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}|\boldsymbol{z}_{k-1}^*)\,d\boldsymbol{x}_{k-1}$$

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{z}_{k-1}^*)p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{k-1}^*) = \frac{p(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}, \boldsymbol{z}_{k-1}^*)p(\boldsymbol{x}_{k-1}, \boldsymbol{z}_{k-1}^*)}{p(\boldsymbol{x}_{k-1}, \boldsymbol{z}_{k-1}^*)p(\boldsymbol{z}_{k-1}^*)} = p(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}|\boldsymbol{z}_{k-1}^*)$$

Introduce Importance Density

$$p(\boldsymbol{x}_k|\boldsymbol{z}_k^*) = \kappa_k p(\boldsymbol{z}_k|\boldsymbol{x}_k)\int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{k-1}^*)\,d\boldsymbol{x}_{k-1}$$

$$\propto p(\boldsymbol{z}_k|\boldsymbol{x}_k)\int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})\frac{p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{k-1}^*)}{q(\boldsymbol{x}_k|\boldsymbol{z}_k^*)}q(\boldsymbol{x}_k|\boldsymbol{z}_k^*)\,d\boldsymbol{x}_{k-1}$$

Choose Importance Density

- Choose to allow the factorization
$$q(\boldsymbol{x}_k|\boldsymbol{z}_k^*) = q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{z}_k^*).\, q(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{k-1}^*)$$
- Observe that
$$q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{z}_k^*) = q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{z}_k)$$
- Substitute
$$q(\boldsymbol{x}_k|\boldsymbol{z}_k^*) = q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{z}_k).\, q(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{k-1}^*)$$
$$\frac{p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{k-1}^*)}{q(\boldsymbol{x}_k|\boldsymbol{z}_k^*)} q(\boldsymbol{x}_k|\boldsymbol{z}_k^*)$$
$$= \frac{p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{k-1}^*)}{q(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{k-1}^*)} . \frac{q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{z}_k).\, q(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{k-1}^*)}{q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{z}_k)}$$

Back to Integral

$$p(\boldsymbol{x}_k|\boldsymbol{z}_k^*) = \kappa_k p(\boldsymbol{z}_k|\boldsymbol{x}_k) \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{k-1}^*)\, d\boldsymbol{x}_{k-1}$$
$$\propto p(\boldsymbol{z}_k|\boldsymbol{x}_k) \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \widetilde{w}_{k-1} q(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{k-1}^*)\, d\boldsymbol{x}_{k-1}$$
$$\widetilde{w}_{k-1} = \frac{p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{k-1}^*)}{q(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{k-1}^*)}$$
$$p(\boldsymbol{x}_k|\boldsymbol{z}_k^*) \propto \sum_{i=1}^{n} \widetilde{w}_k^{(i)} \delta\left(\boldsymbol{x}_k - \boldsymbol{x}_k^{(i)}\right)$$
$$= \sum_{i=1}^{n} \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k) p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})}{q(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{k-1}^*)} \widetilde{w}_{k-1}^{(i)} \delta\left(\boldsymbol{x}_k - \boldsymbol{x}_k^{(i)}\right)$$

Weights

$$\widetilde{w}_{k-1} \propto \frac{p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{k-1}^*)}{q(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{k-1}^*)}$$
$$\widetilde{w}_k^{(i)} = \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k) p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})}{q(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{k-1}^*)} \widetilde{w}_{k-1}^{(i)}$$

Monte Carlo Estimate (normalize)

$$W_k^{(j)} = \frac{\widetilde{w}_k^{(j)}}{\sum_{i=1}^{N} \widetilde{w}_{k-1}^{i}}$$

Substitute

$$p(\boldsymbol{x}_k|\boldsymbol{z}_k^*) = \sum_{i=1}^{n} W_k^{(i)} \delta\left(\boldsymbol{x}_k - \boldsymbol{x}_k^{(i)}\right)$$
$$W_k^{(i)} = W_{k-1}^{(i)} \frac{p\left(\boldsymbol{z}_k|\boldsymbol{x}_k^{(i)}\right) p\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}\right)}{q\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{z}_k\right)}$$

Choice of Importance Density

- Critical step in particle filter design.
- Optimal importance density: minimizes the variance of importance weights conditioned upon $\boldsymbol{x}_{k-1}^{(i)}$ and $\boldsymbol{z}_k$

$$q\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{z}_k\right) = p\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{z}_k\right)$$

Choice of Importance Density

- Popular choice: bootstrap filter

$$q\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{z}_k\right) = p\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}\right)$$

$$W_k^{(i)} \propto W_{k-1}^{(i)} p\left(\boldsymbol{z}_k|\boldsymbol{x}_k^{(i)}\right)$$

- Assume that the weights are resampled every cycle such that they become uniformly distributed.

$$W_k^{(i)} = p\left(\boldsymbol{z}_k|\boldsymbol{x}_k^{(i)}\right)$$

Importance Sampling

- Unlike Monte Carlo and ensemble filters, does not use a simple distribution to generate the random samples.
- Generate points that fit a general pdf.
- Importance Density: proposal density that can easily generate samples.
- Algorithm used: Sequential Importance Sampling (SIS)

Sequential Importance Sampling

- Weight recursion

$$W_k^{(i)} = W_{k-1}^{(i)} \frac{p\left(\boldsymbol{z}_k|\boldsymbol{x}_k^{(i)}\right) p\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}\right)}{q\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{z}_k\right)}$$

$p\left(\boldsymbol{z}_k|\boldsymbol{x}_k^{(i)}\right)$ =likelihood function

$p\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}\right)$ =transition prior

$q\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{z}_k\right)$ =importance density

Estimates

- Based on the approximate discretize pdf
- State Estimate

$$\hat{\boldsymbol{x}}_k^+ = \sum_{i=1}^{n} W_k^{(i)} \boldsymbol{x}_k^{(i)}$$

- Error Covariance

$$P_k^+ = \sum_{i=1}^{n} W_k^{(i)} \left(\boldsymbol{x}_k^{(i)} - \hat{\boldsymbol{x}}_k^+\right)\left(\boldsymbol{x}_k^{(i)} - \hat{\boldsymbol{x}}_k^+\right)^T$$

SIS Algorithm

- Input $\left\{\boldsymbol{x}_k^{(i-1)}, W_k^{(i-1)}, i = 1, \dots, N\right\}, \boldsymbol{z}_k$
- Output $\left\{\boldsymbol{x}_k^{(i)}, W_k^{(i)}, i = 1, \dots, N\right\}$

$Sum_w = 0$

for $i = 1{:}N$

Draw $\boldsymbol{x}_k^{(i)}$ from $q\left(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{z}_k\right)$

$W_k^{(i)} = W_{k-1}^{(i)} p\left(\boldsymbol{z}_k | \boldsymbol{x}_k^{(i)}\right) p\left(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}^{(i)}\right) / q\left(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{z}_k\right)$

$Sum_w = Sum_w + W_k^{(i)}$

end

for $i = 1{:}N$

$W_k^{(i)} = W_k^{(i)} / Sum_w$

end

Degeneracy Phenomenon

- Normalized weights tend to concentrate into one particle after a number of recursions (all other particles degenerate)
- Measure of degeneracy N_{eff}

$$\hat{N}_{eff} = 1 \Big/ \sum_{i=1}^{n} \left(w_k^{(i)}\right)^2, 1 \le \hat{N}_{eff} \le N$$

$$\hat{N}_{eff} = N, w_k^{(i)} = 1/N, i = 1, 2, \dots, N$$

$$\hat{N}_{eff} = 1, w_k^{(j)} = 1,$$
$$w_k^{(i)} = 0, i \ne j, i = 1, \dots, j-1, j+1, \dots, N$$

Degeneracy After 2 Cycles

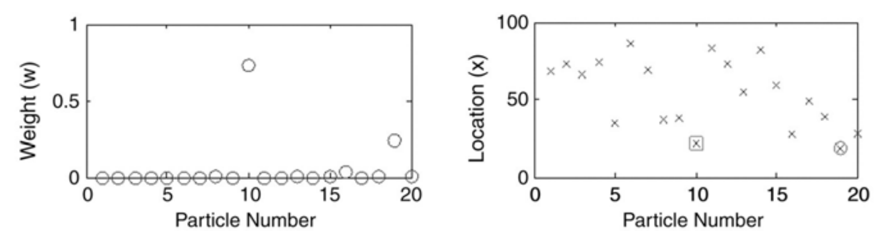


Figure 7.11

Resampling

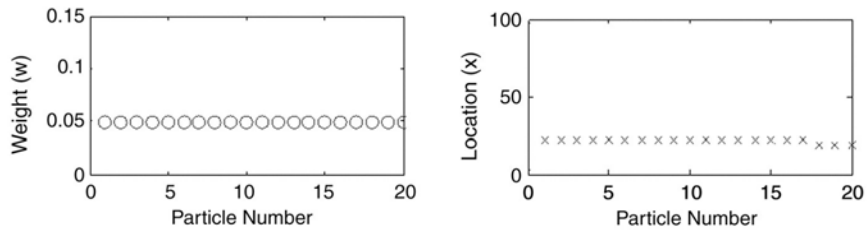


Figure 7.12

Resampling

- Required if $Neff < N_{thresh}$
- Particles get more meaningful weights (uniformly distributed weights)
- Eliminates samples with low weights and produces a set with uniform weights.

Resampling Algorithms Idea

- Generate a random number $u_1 \sim U[0,1]$
- Add the weights until their sum exceeds u_1 at which point choose the corresponding particle $\boldsymbol{x}_k^i$ as a resampled point $\boldsymbol{x}_k^{j*}$ and save the index i^j of its parent
- Can show that the resampled particles' pdf converges to $p(\boldsymbol{x}_k|\boldsymbol{z}_k)$

Resampling Algorithm

- Input $\left\{\boldsymbol{x}_k^{(i)}, w_k^{(i)}, i = 1, \dots, N\right\}$
- Output $\left\{\boldsymbol{x}_k^{(j)*}, w_k^{(j)}, i^j, i = 1, \dots, N\right\}$

$c_1 = w_k^{(1)}$
for $i = 2\colon N$
$c_i = c_{i-1} + w_k^{(i)}$
end
Draw a starting point $u_1 \sim U\left[0, N^{-1}\right]$
for $j = 1\colon N$
$u_j = u_1 + N^{-1}(j-1)$
While $u_j > c_i$
$i = i + 1$
end
$\boldsymbol{x}_k^{(j)*} = \boldsymbol{x}_k^{(i)}$
$w_k^{(i)} = N^{-1}$
$i^j = i$
end

Particle Filter Algorithm (Simon)

- Generate N initial particles $\boldsymbol{x}_0^{+(i)}, i = 1, \dots, N$, using the pdf $p(x^0)$
- Predictor step to obtain

$$\boldsymbol{x}_k^{-(i)} = \phi\left(\boldsymbol{x}_{k-1}^{+(i)}, \boldsymbol{w}_{k-1}^{(i)}\right), i = 1, \dots, N$$

- Compute the weights $W_k^{(i)}$ then scale them so that their sum is unity.
- Corrector step:

$$\boldsymbol{x}_k^{+(i)}, i = 1, \dots, N, \quad \boldsymbol{x}_k^{+(i)} = \frac{1}{N}\sum_{i-1}^{N} \boldsymbol{x}_k^{+(i)}$$

SIR Particle Filter Algorithm

- Input $\left\{\boldsymbol{x}_{k-1}^{(i)}, i = 1, \dots, N\right\}, \boldsymbol{z}_k$
- Output $\left\{\boldsymbol{x}_k^{(i)}, i = 1, \dots, N\right\}$

$Sum_w = 0$
for $i = 1{:}N$
Draw $\boldsymbol{x}_k^{(i)} \sim p\left(\boldsymbol{x}_k \middle| \boldsymbol{x}_{k-1}^{(i)}\right)$
Calculate $W_k^{(i)} = W_{k-1}^{(i)} p\left(\boldsymbol{z}_k | \boldsymbol{x}_k^{(i)}\right)$
$Sum_w = Sum_w + W_k^{(i)}$
End
for $j = 1{:}N$
$W_k^{(i)} = W_k^{(i)} / Sum_w$
end
$\boldsymbol{x}_k^{(j)*} = \boldsymbol{x}_k^{(i)}$
Resample using SIR

References

- R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, 4ed, J. Wiley, NY, 2012.
- R. V. Hogg, J. W. McKean, A. T. Craig, *Introduction to Mathematical Statistics*, 6th Ed., Pearson/Prentice-Hall, Upper Saddle River, NJ, 2005.
- B. Ristic, S. Arulampalam, N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House, Boston, 2004.