Particle Filtering

Sanjiban Choudhury

TAs: Matthew Rockett, Gilwoo Lee, Matt Schmittle
Question: What makes a good proposal distribution?
Applying importance sampling to Bayes filtering

Target distribution: Posterior

\[ bel(x_t) = \eta P(z_t|x_t) \int p(x_t|u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1} \]
Applying importance sampling to Bayes filtering

Target distribution : Posterior

$$bel(x_t) = \eta P(z_t|x_t) \int p(x_t|u_t, x_{t-1})bel(x_{t-1})dx_{t-1}$$

Proposal distribution : After applying motion model

$$\overline{bel}(x_t) = \int p(x_t|u_t, x_{t-1})bel(x_{t-1})dx_{t-1}$$

Why is this easy to sample from?
Applying importance sampling to Bayes filtering

Target distribution: Posterior

\[ \text{bel}(x_t) = \eta P(z_t|x_t) \int p(x_t|u_t, x_{t-1})\text{bel}(x_{t-1})dx_{t-1} \]

Proposal distribution: After applying motion model

\[ \overline{\text{bel}}(x_t) = \int p(x_t|u_t, x_{t-1})\text{bel}(x_{t-1})dx_{t-1} \]

Why is this easy to sample from?

Importance Ratio:

\[ r = \frac{\text{bel}(x_t)}{\overline{\text{bel}}(x_t)} = \eta P(z_t|x_t) \]
Question: What are our options for non-parametric belief representations?

1. Histogram filter
2. Normalized importance sampling
3. Particle filter
Approach 2: Normalized Importance Sampling

\[ bel(x_{t-1}) = \left\{ x_{t-1}^1, x_{t-1}^2, \ldots, x_{t-1}^M \right\} \]

for \( i = 1 \) to \( M \)

sample \( \bar{x}_t^i \sim P(x_t | u_t, x_t^i) \)

\[ w_t^i = P(z_t | \bar{x}_t^i) w_{t-1}^i \]

for \( i = 1 \) to \( M \)

\[ w_t = \frac{w_t^i}{\sum_i w_t^i} \]

\[ bel(x_t) = \left\{ \frac{\bar{x}_t^1}{w_t}, \ldots, \frac{\bar{x}_t^M}{w_t} \right\} \]
Problem: What happens after enough iterations?
**Problem:** What happens after enough iterations?

Particles don’t move - can get stuck in regions of low probability

This is the same complaint we had about histogram filters!
Key Idea: Resample!

Why? Get rid of bad particles
Approach 3: Particle Filtering

\[
bel(x_{t-1}) = \left\{ \begin{array}{c}
x_{t-1}^1, w_{t-1}^1 \\
... \\
x_{t-1}^M, w_{t-1}^M \
\end{array} \right\}
\]

for \( i = 1 \) to \( M \)

sample \( \bar{x}_t^i \sim P(x_t|u_t, x_t^i) \)

for \( i = 1 \) to \( M \)

\[
w_t^i = P(z_t|\bar{x}_t^i)w_{t-1}^i
\]

all weights \( = 1/M \)

\[
sample \ x_t^i \sim w_t^i \quad bel(x_t) = \left\{ \begin{array}{c}
x_t^1 \\
... \\
x_t^M \
\end{array} \right\}
\]
Virtues of resampling
Virtues of resampling

$u_t \rightarrow z_t$
Virtues of resampling

$u_t$  $z_t$  resample
Virtues of resampling
Virtues of resampling

\[ u_t \rightarrow z_t \rightarrow \text{resample} \rightarrow u_{t+1} \rightarrow z_{t+1} \]
Virtues of resampling
Why use particle filters?

1. Can answer any query

2. Will work for any distribution, including multi-modal (unlike Kalman filter)

3. Scale well in computational resources (embarrassingly parallel)

4. Easy to implement!
Non-parametric Filters

- Grid up state space

Histogram Filter

- Use a fixed set of samples

Normalized Importance Sampling

- Resample

Particle Filter

Same fundamental Bayes rule again and again ...
Are we done?

No!

Lots of practical problems to deal with
Problem 1: Two room challenge

Given: Particles equally distributed, no motion, no observation

What happens?
Problem 1: Two room challenge

Given: Particles equally distributed, no motion, no observation

What happens?

All particles migrate to the other room!!
Reason: Resampling increases variance

Resampling collapses particles, reduces diversity, increases variance w.r.t true posterior
Fix 1: Choose when to resample

Key idea: If variance of weights low, don’t resample
Fix 1: Choose when to resample

Key idea: If variance of weights low, don’t resample

We can implement this condition in various ways
Fix 1: Choose when to resample

Key idea: If variance of weights low, don’t resample

We can implement this condition in various ways

1. All weights are equal - don’t resample
Fix 1: Choose when to resample

Key idea: If variance of weights low, don’t resample

We can implement this condition in various ways

1. All weights are equal - don’t resample

2. Entropy of weights high - don’t resample
Fix 1: Choose when to resample

Key idea: If variance of weights low, don’t resample

We can implement this condition in various ways

1. All weights are equal - don’t resample

2. Entropy of weights high - don’t resample

3. Ratio of max to min weights low - don’t resample
Fix 2: Low variance sampling
Fix 2: Low variance sampling

1. Algorithm **systematic_resampling**(S,n):
   
2. \( S' = \emptyset, c_1 = w^1 \) \hspace{1cm} \textbf{Assumption: weights sum to 1}
3. \textbf{For } \ i = 2 \ldots n \hspace{1cm} \textbf{Generate cdf}
4. \( c_i = c_{i-1} + w^i \)
5. \( u_i \sim U[0, n^{-1}], i = 1 \) \hspace{1cm} \textbf{Initialize threshold}
6. \textbf{For } \ j = 1 \ldots n \hspace{1cm} \textbf{Draw samples …}
7. \textbf{While } ( u_j > c_i ) \hspace{1cm} \textbf{Skip until next threshold reached}
8. \( i = i + 1 \)
9. \( S' = S' \cup \{< x^i, n^{-1} > \} \) \hspace{1cm} \textbf{Insert}
10. \( u_j = u_j + n^{-1} \) \hspace{1cm} \textbf{Increment threshold}

11. \textbf{Return } S' \hspace{1cm} \textbf{Also called stochastic universal sampling}
Why does this work?

1. What happens when all weights equal?

2. What happens if you have ONE large weight and many tiny weights?

\[ w_1 = 0.5, \quad w_2 = \frac{0.5}{1000}, \quad w_3 = \frac{0.5}{1000}, \quad \ldots \quad w_{1001} = \frac{0.5}{1000} \]
Problem 2: Particle Starvation

No particles in the vicinity of the current state
Problem 2: Particle Starvation

No particles in the vicinity of the current state

Why?

1. Unlucky set of samples

2. Committed to the wrong mode in a multi-modal scenario

3. Bad set of measurements
Fix: Add new particles
Fix: Add new particles

Which distribution should be used to add new particles?
Fix: Add new particles

Which distribution should be used to add new particles?

1. Uniform distribution

2. Biased around last good measurement

3. Directly from the sensor model
Fix: Add new particles

When should we add new samples?
Fix: Add new particles

When should we add new samples?

Key Idea: As soon as importance weights become too small, add more samples
Fix: Add new particles

When should we add new samples?

Key Idea: As soon as importance weights become too small, add more samples

1. Threshold the total sum of weights

2. Fancy estimator that checks rate of change.
Problem 3: Observation model too good!

Observation model is so peaky, that all particles die!
Problem 3: Observation model too good!

Observation model is so peaky, that all particles die!

Fixes

1. Sample from a better proposal distribution than motion model!

2. Squash the observation model (apply a power of $1/m$ to all probabilities. $m$ observations count as one)

3. Last resort: Smooth your observation model with a Gaussian (you are pretending your observation model is worse than it is)
Fix 1: Sample from a better proposal distribution

Key Idea: Sample and weigh particles correctly

\[ bel(x_t) = \eta P(z_t|x_t) \int P(x_t|x_{t-1}, u_t) \, bel(x_{t-1}) \, dx_{t-1} \]

(Sample) \hspace{2cm} (Reweigh)

Contact observation may kill ALL particles!
Problem 4: How many samples is enough?

Example: We typically need more particles at the beginning of run

Key idea: KLD Sampling (Fox et al. 2002)

1. Partition the state-space into bins

2. When sampling, keep track of the number of bins

3. Stop sampling when you reach a statistical threshold that depends on the number of bins

(If all samples fall in a small number of bins -> lower threshold)
Figure 8.18  KLD-sampling: Typical evolution of number of samples for a global localization run, plotted against time (number of samples is shown on a log scale). The solid line shows the number of samples when using the robot’s laser range-finder, the dashed graph is based on sonar sensor data.
1. Particle Filter = Sample from motion model, weight by observation

2. Particle filters are for localization

3. Particle filters are to do with samples
Closing: Myth busting Particle filters

1. Particle Filter = Sample from motion model, weight by observation
   (sample from any good proposal distribution)

2. Particle filters are for localization

3. Particle filters are to do with samples
1. Particle Filter = Sample from motion model, weight by observation

(sample from any good proposal distribution)

2. Particle filters are for localization

(any continuous space estimation problem)

3. Particle filters are to do with samples
Closing: Myth busting Particle filters

1. Particle Filter = Sample from motion model, weight by observation
   (sample from any good proposal distribution)

2. Particle filters are for localization
   (any continuous space estimation problem)

3. Particle filters are to do with samples
   (normalized importance sampling also uses samples but no resampling)