CSE-571
Robotics

Bayes Filter Implementations

Particle filters
Motivation

- So far, we discussed the
  - Kalman filter: Gaussian, linearization problems, multi-modal beliefs

- Particle filters are a way to **efficiently** represent **non-Gaussian distributions**

- Basic principle
  - Set of state hypotheses ("particles")
  - Survival-of-the-fittest
Sample-based Localization (sonar)
Density Approximation

- Particle sets can be used to approximate densities

- The more particles fall into an interval, the higher the probability of that interval

- How to draw samples form a function/distribution?
Rejection Sampling

- Let us assume that $f(x) \leq 1$ for all $x$
- Sample $x$ from a uniform distribution
- Sample $c$ from $[0,1]$
- If $f(x) > c$ keep the sample
  otherwise reject the sample
Importance Sampling Principle

- We can even use a different distribution $g$ to generate samples from $f$
- By introducing an importance weight $w$, we can account for the “differences between $g$ and $f$”
- $w = f / g$
- $f$ is often called target
- $g$ is often called proposal
Resampling

- **Given**: Set $S$ of weighted samples.

- **Wanted**: Random sample, where the probability of drawing $x_i$ is given by $w_i$.

- Typically done $n$ times with replacement to generate new sample set $S'$. 

Resampling

- Roulette wheel
- Binary search, $n \log n$
- Stochastic universal sampling
- Systematic resampling
- Linear time complexity
- Easy to implement, low variance
Particle Filters
Sensor Information: Importance Sampling

\[ Bel(x) \leftarrow \alpha \, p(z \mid x) \, Bel^{-}(x) \]

\[ w \leftarrow \frac{\alpha \, p(z \mid x) \, Bel^{-}(x)}{Bel^{-}(x)} = \alpha \, p(z \mid x) \]
Robot Motion

\[
Bel^{-}(x) \leftarrow \int p(x \mid u, x') Bel(x') \, dx'
\]
Sensor Information: Importance Sampling

\[ Bel(x) \leftarrow \alpha p(z \mid x) Bel^{-}(x) \]

\[ w \leftarrow \frac{\alpha p(z \mid x) Bel^{-}(x)}{Bel^{-}(x)} = \alpha p(z \mid x) \]
Robot Motion

\[ Bel^{-}(x) \leftarrow \int p(x \mid u, x') Bel(x') \, dx' \]
Particle Filter Algorithm

1. Algorithm \texttt{particle\_filter}( S_{t-1}, u_{t-1} z_t ):
2. \( S_t = \emptyset, \quad \eta = 0 \)
3. \textbf{For } \( i = 1 \ldots n \) \hspace{1cm} \textbf{Generate new samples}
4. \hspace{1cm} \text{Sample index } j(i) \text{ from the discrete distribution given by } w_{t-1}
5. \hspace{1cm} \text{Sample } x_t^i \text{ from } p(x_t | x_{t-1}, u_{t-1}) \text{ using } x_t^{j(i)} \text{ and } u_{t-1}
6. \hspace{1cm} w_t^i = p(z_t | x_t^i) \hspace{1cm} \textbf{Compute importance weight}
7. \hspace{1cm} \eta = \eta + w_t^i \hspace{1cm} \textbf{Update normalization factor}
8. \hspace{1cm} S_t = S_t \cup \{< x_t^i, w_t^i >\} \hspace{1cm} \textbf{Insert}
9. \textbf{For } \( i = 1 \ldots n \)
10. \hspace{1cm} w_t^i = w_t^i / \eta \hspace{1cm} \textbf{Normalize weights}
Particle Filter Algorithm

\[ Bel(x_t) = \eta p(z_t | x_t) \int p(x_t | x_{t-1}, u_{t-1}) Bel(x_{t-1}) \, dx_{t-1} \]

- draw \( x_{t-1}^i \) from \( Bel(x_{t-1}) \)
- draw \( x_t^i \) from \( p(x_t | x_{t-1}^i, u_{t-1}) \)

Importance factor for \( x_t^i \):

\[
w_t^i = \frac{\text{target distribution}}{\text{proposal distribution}}
\]

\[ = \eta p(z_t | x_t) \frac{p(x_t | x_{t-1}, u_{t-1}) Bel(x_{t-1})}{p(x_t | x_{t-1}, u_{t-1}) Bel(x_{t-1})} \]

\[ \propto p(z_t | x_t) \]
Proximity Sensor Model Reminder

Laser sensor

Sonar sensor
Using Ceiling Maps for Localization

[Dellaert et al. 99]
Vision-based Localization
Under a Light

Measurement $z$: $P(z|x)$:
Next to a Light

Measurement $z$:  

$P(z|x)$:
Elsewhere

Measurement $z$: $P(z|x)$:
Global Localization Using Vision
Recovery from Failure
Localization for AIBO robots
Adaptive Sampling

Robot position

Start
KLD-Sampling Sonar

Adapt number of particles on the fly based on statistical approximation measure
KLD-Sampling Laser
Particle Filter Projection

![Graphs and diagrams related to Particle Filter Projection](image-url)
Density Extraction
Sampling Variance
Bayes Filter Implementations

Discrete filters
Piecewise Constant
Discrete Bayes Filter Algorithm

1. Algorithm `Discrete_Bayes_filter( Bel(x),d )`:
2. \( \eta = 0 \)
3. If \( d \) is a perceptual data item \( z \) then
4.     For all \( x \) do
5.     \( Bel'(x) = P(z | x) Bel(x) \)
6.     \( \eta = \eta + Bel'(x) \)
7.     For all \( x \) do
8.     \( Bel'(x) = \eta^{-1} Bel'(x) \)
9. Else if \( d \) is an action data item \( u \) then
10.    For all \( x \) do
11.    \( Bel'(x) = \sum_{x'} P(x | u, x') Bel(x') \)
12. Return \( Bel'(x) \)
Piecewise Constant Representation

$$Bel(x_t = <x, y, \theta>)$$
Grid-based Localization
Sonars and Occupancy Grid Map
Tree-based Representation

**Idea:** Represent density using a variant of Octrees
Tree-based Representations

- Efficient in space and time
- Multi-resolution