Introduction to Mobile Robotics

Bayes Filter – Particle Filter and Monte Carlo Localization

Wolfram Burgard, Cyrill Stachniss, Maren Bennewitz, Giorgio Grisetti, Kai Arras
Motivation

- Recall: Discrete filter
  - Discretize the continuous state space
  - High memory complexity
  - Fixed resolution (does not adapt to the belief)

- Particle filters are a way to efficiently represent non-Gaussian distribution

- Basic principle
  - Set of state hypotheses (“particles”)
  - Survival-of-the-fittest
Sample-based Localization (sonar)
Mathematical Description

- Set of weighted samples

\[ S = \left\{ \langle s[i], w[i] \rangle \mid i = 1, \ldots, N \right\} \]

State hypothesis  Importance weight

- The samples represent the posterior

\[ p(x) = \sum_{i=1}^{N} w_i \cdot \delta_{s[i]}(x) \]
Function Approximation

- Particle sets can be used to approximate functions

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

- How to draw samples from a function/distribution?
Let us assume that $f(x) < 1$ 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
  - Pre-condition: $f(x) > 0 \Rightarrow g(x) > 0$
Importance Sampling with Resampling: Landmark Detection Example
Distributions
Distributions

Wanted: samples distributed according to $p(x| z_1, z_2, z_3)$
This is Easy!

We can draw samples from $p(x|z_i)$ by adding noise to the detection parameters.
Importance Sampling

Target distribution $f : p(x \mid z_1, z_2, \ldots, z_n) = \prod_{k} p(z_k \mid x) \frac{p(x)}{p(z_1, z_2, \ldots, z_n)}$

Sampling distribution $g : p(x \mid z_l) = \frac{p(z_l \mid x) p(x)}{p(z_l)}$

Importance weights $w : \frac{f}{g} = \frac{p(x \mid z_1, z_2, \ldots, z_n)}{p(x \mid z_l)} = \frac{p(z_l) \prod_{k \neq l} p(z_k \mid x)}{p(z_1, z_2, \ldots, z_n)}$
Importance Sampling with Resampling

Weighted samples

After resampling
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

$$Bel^{-}(x) \leftarrow \int p(x | u, x') Bel(x') \, dx'$$
Particle Filter Algorithm

- Sample the next generation for particles using the proposal distribution

- Compute the importance weights:
  
  \[ \text{weight} = \frac{\text{target distribution}}{\text{proposal distribution}} \]

- Resampling: “Replace unlikely samples by more likely ones”

- [Derivation of the MCL equations on the blackboard]
Particle Filter Algorithm

1. Algorithm `particle_filter`( $S_{t-1}$, $u_{t-1}$, $z_t$):
2. $S_t = \emptyset$, $\eta = 0$
3. For $i = 1$K $n$  
   \textit{Generate new samples}
4. Sample index $j(i)$ from the discrete distribution given by $w_{t-1}$
5. Sample $x_i^t$ from $p(x_t | x_{t-1}, u_{t-1})$ using $x_{t-1}^{j(i)}$ and $u_{t-1}$
6. $w_i^t = p(z_t | x_i^t)$  
   \textit{Compute importance weight}
7. $\eta = \eta + w_i^t$  
   \textit{Update normalization factor}
8. $S_t = S_t \cup \{< x_i^t, w_i^t >\}$  
   \textit{Insert}
9. For $i = 1$K $n$
10. $w_i^t = w_i^t / \eta$  
    \textit{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 \( x^i_{t-1} \) from \( Bel(x_{t-1}) \)
- draw \( x^i_t \) from \( p(x_t | x^i_{t-1}, u_{t-1}) \)
- Importance factor for \( x^i_t \):

\[
w^i_t = \frac{\text{target distribution}}{\text{proposal distribution}}
= \frac{\eta \, p(z_t | x_t) \, p(x_t | x_{t-1}, u_{t-1}) \, Bel \left( x_{t-1} \right)}{p(x_t | x_{t-1}, u_{t-1}) \, Bel \left( x_{t-1} \right)}
\propto p(z_t | x_t)
\]
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
Resampling Algorithm

1. Algorithm systematic_resampling(S,n):

2. \( S' = \emptyset, c_1 = w^1 \)
3. For \( i = 2K \ n \) \ Generate cdf \n4. \( c_i = c_{i-1} + w^i \)
5. \( u_1 \sim U[0,n^{-1}], i = 1 \) \ Initialize threshold \n6. For \( j = 1K \ n \) \ Draw samples ... \n7. While ( \( u_j > c_i \) ) \ Skip until next threshold reached \n8. \( i = i + 1 \)
9. \( S' = S' \cup \{ < x^i, n^{-1} > \} \) \ Insert \n10. \( u_{j+1} = u_j + n^{-1} \) \ Increment threshold

11. Return \( S' \)

Also called stochastic universal sampling
Mobile Robot Localization

- Each particle is a potential pose of the robot

- Proposal distribution is the motion model of the robot (prediction step)

- The observation model is used to compute the importance weight (correction step)

[For details, see PDF file on the lecture web page]
Motion Model Reminder
Proximity Sensor Model Reminder

Laser sensor

Sonar sensor
Sample-based Localization (sonar)
Initial Distribution
After Incorporating Ten Ultrasound Scans
After Incorporating 65 Ultrasound Scans
Estimated Path
Localization for AIBO robots
Using Ceiling Maps for Localization

[Dellaert et al. 99]
Vision-based Localization

\[ h(x) \]

\[ P(z|x) \]
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
Limitations

- The approach described so far is able to
  - track the pose of a mobile robot and to
  - globally localize the robot.

- How can we deal with localization errors (i.e., the kidnapped robot problem)?
Approaches

- Randomly insert samples (the robot can be teleported at any point in time).
- Insert random samples proportional to the average likelihood of the particles (the robot has been teleported with higher probability when the likelihood of its observations drops).
Summary – Particle Filters

- Particle filters are an implementation of recursive Bayesian filtering
- They represent the posterior by a set of weighted samples
- They can model non-Gaussian distributions
- Proposal to draw new samples
- Weight to account for the differences between the proposal and the target
- Monte Carlo filter, Survival of the fittest, Condensation, Bootstrap filter
Summary – PF Localization

- In the context of localization, the particles are propagated according to the motion model.
- They are then weighted according to the likelihood of the observations.
- In a re-sampling step, new particles are drawn with a probability proportional to the likelihood of the observation.