Introduction to Mobile Robotics

Bayes Filter – Particle Filter and Monte Carlo Localization

Wolfram Burgard
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

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

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?
Rejection Sampling

- 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 $f$”
  $$w = \frac{f}{g}$$
- $f$ is called target
- $g$ is called proposal
- Pre-condition:
  $$f(x) > 0 \rightarrow g(x) > 0$$
- Derivation: See webpage
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 $p(x|z_l)$ by adding noise to the detection parameters.
Importance Sampling

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

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

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

\[ Bel (x) \rightarrow p(x \mid u, x') \ Bel(x') \ dx' \]
Sensor Information: Importance Sampling

\[ Bel(x) \sim p(z \mid x) Bel(x) \]

\[ w \sim \frac{p(z \mid x) Bel(x)}{Bel(x)} = p(z \mid x) \]
Robot Motion

\[ Bel (x) \quad \neg \quad p(x \mid u, x') \quad Bel(x') \quad dx' \]
Particle Filter Algorithm

- Sample the next generation for particles using the proposal distribution

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

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

1. Algorithm `particle_filter(S_{t-1}, u_t, z_t)`:  
2. \( S_t = \emptyset, \quad \eta = 0 \)  
3. **For** \( i = 1, \ldots, n \) **Generate new samples**  
4. Sample index \( j(i) \) from the discrete distribution given by \( w_{t-1} \)  
5. Sample \( p(x_t \mid x_{t-1}, u_t) \) using \( u_t \) and \( x_{t-1}^{j(i)} \) **Compute importance weight**  
6. \[ w_t^i = p(z_t \mid x_t^i) \]  
7. \[ = + w_t^i \] **Update normalization factor**  
8. \( S_t = S_t \cup \{< x_t^i, w_t^i >\} \) **Add to new particle set**  
9. **For** \( i = 1, \ldots, n \)  
10. \( w_t^i = w_t^i / h \) **Normalize weights**
Particle Filter Algorithm

\[ Bel(x_t) = p(z_t \mid x_t) \ p(x_t \mid x_{t-1}, u_t) \ Bel(x_{t-1}) \ dx_{t-1} \]

- Draw \( x^i_{t-1} \) from \( Bel(x_{t-1}) \)
- Draw \( x^i_t \) from \( p(x_t \mid x^i_{t-1}, u_t) \)

Importance factor for \( x^i_t \):

\[
\mu \ p(z_t \mid x_t) \ p(x_t \mid x_{t-1}, u_t) \ Bel(x_{t-1})
\]

\[
= \frac{p(z_t \mid x^i_t) \ p(x^i_t \mid x_{t-1}, u_t) \ Bel(x_{t-1})}{p(x_t \mid x^i_{t-1}, u_t) \ Bel(x_{t-1})}
\]

\[
= w^i_t = \text{target distribution} \over \text{proposal distribution}
\]
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** $i = 2K \ n$

4. $c_i = c_{i-1} + w^i$

5. $u_1 \sim U \ [0, n^{-1}]$, $i = 1$

6. **For** $j = 1K \ n$

7. **While** ($u_j > c_i$)

8. $i = i + 1$

9. $S' = S' \cup \{< x^i, n^{-1} >\}$

10. $u_{j+1} = u_j + n^{-1}$

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

According to the estimated motion
- Decompose the motion into
  - Traveled distance
  - Start rotation
  - End rotation
Motion Model Reminder

- Uncertainty in the translation of the robot: Gaussian over the traveled distance
- Uncertainty in the rotation of the robot: Gaussians over start and end rotation
- For each particle, draw a new pose by sampling from these three individual normal distributions
Motion Model Reminder

Start

10 meters
Proximity Sensor Model Reminder

**Laser sensor**

**Sonar sensor**
Mobile Robot Localization Using Particle Filters (1)

- Each particle is a potential pose of the robot
- The set of weighted particles approximates the posterior belief about the robot’s pose (target distribution)
Mobile Robot Localization Using Particle Filters (2)

- Particles are drawn from the motion model (proposal distribution)
- Particles are weighted according to the observation model (sensor model)
- Particles are resampled according to the particle weights
Why is resampling needed?

- We only have a finite number of particles
- Without resampling: The filter is likely to lose track of the “good” hypotheses
- Resampling ensures that particles stay in the meaningful area of the state space
Sample-based Localization (sonar)
Initial Distribution
After Incorporating Ten Ultrasound Scans
After Incorporating 65 Ultrasound Scans
Estimated Path
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
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 a fixed number of samples
- This assumes that the robot can be teleported at any point in time
- Alternatively, insert random samples proportional to the average likelihood of the particles
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.