Probabilistic Robotics

Discrete Filters and Particle Filters Models

Some slides adopted from: Wolfram Burgard, Cyrill Stachniss, Maren Bennewitz, Kai Arras and Probabilistic Robotics Book

\[ \text{Bel}(x \mid z, u) = \alpha p(z \mid x) \int_{x'} p(x \mid u, x') \text{Bel}(x')dx' \]
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. \( Bel'(x) = \frac{1}{\eta}Bel'(x) \)
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 \( Bel'(x) \)
Implementation (1)

- To update the belief upon sensory input and to carry out the normalization one has to iterate over all cells of the grid.
- Especially when the belief is peaked (which is generally the case during position tracking), one wants to avoid updating irrelevant aspects of the state space.
- One approach is not to update entire sub-spaces of the state space.
- This, however, requires to monitor whether the robot is de-localized or not.
- To achieve this, one can consider the likelihood of the observations given the active components of the state space.
Implementation (2)

- To efficiently update the belief upon robot motions, one typically assumes a bounded Gaussian model for the motion uncertainty.
- This reduces the update cost from $O(n^2)$ to $O(n)$, where $n$ is the number of states.
- The update can also be realized by shifting the data in the grid according to the measured motion.
- In a second step, the grid is then convolved using a separable Gaussian Kernel.
- Two-dimensional example:

  \[
  \begin{array}{ccc}
  1/16 & 1/8 & 1/16 \\
  1/8 & 1/4 & 1/8 \\
  1/16 & 1/8 & 1/16 \\
  \end{array}
  \quad \Rightarrow \quad 
  \begin{array}{ccc}
  1/4 \\
  1/2 \\
  1/4 \\
  \end{array}
  + 
  \begin{array}{ccc}
  1/4 & 1/2 & 1/4 \\
  \end{array}
  \]

- Fewer arithmetic operations
- Easier to implement

Grid-based Localization
Sonars and Occupancy Grid Map

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

\[ S = \{ (s[i], w[i]) \mid i = 1, \ldots, N \} \]

- 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 form 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
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 often called the target

\( g \) is often called the proposal

Pre-condition:

\[
f(x) > 0 \Rightarrow g(x) > 0
\]
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 $f : p(x | z_1, z_2, ..., z_n) = \frac{\prod_{i} p(z_i | x) \cdot p(x)}{p(z_1, z_2, ..., z_n)}$

Sampling distribution $g : p(x | z_i) = \frac{p(z_i | x) \cdot p(x)}{p(z_i)}$

Importance weights $w : \frac{f}{g} = \frac{p(x | z_1, z_2, ..., z_n)}{p(x | z_i)} = \frac{p(z_i) \cdot \prod_{i} p(z_i | x)}{p(z_1, z_2, ..., z_n)}$

The more is the sample consistent with all the measurements the higher the weight will be.

---

Importance Sampling with Resampling

Weighted samples

After resampling

Given: set of samples
Wanted: single sample where probability of sample is given by $w_i$
Repeat $n$ times
Particle Filters

Sensor Information: Importance Sampling

\[
\begin{align*}
\text{Bel}(x) & \leftarrow \alpha p(z | x) \text{Bel}'(x) \\
w & \leftarrow \frac{\alpha p(z | x) \text{Bel}'(x)}{\text{Bel}'(x)} = \alpha p(z | x)
\end{align*}
\]
Robot Motion

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

Sensor Information: Importance Sampling

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

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

\[ Bel^\sim(x) \leftarrow \int p(x \mid u, x') Bel(x') \, dx' \]

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”
Particle Filter Algorithm

1. Algorithm particle_filter( S(t−1), u(t−1), z(t)):
2. \( S_t = \emptyset , \eta = 0 \)
3. For \( i = 1 \ldots n \) Generate new samples
4. Sample index \( j(i) \) from the discrete distribution given by \( w_{t−1} \)
5. Sample \( x'_i \) from \( p(x_i | x_{t−1}, u_{t−1}) \) using \( S_{t−1}^{j(i)} \) and \( u_{t−1} \)
6. \( w'_i = p(z_t | x'_i) \) Compute importance weight
7. \( \eta = \eta + w'_i \) Update normalization factor
8. \( S_t = S_t \cup \{ x'_i, w'_i \} \) Insert
9. For \( i = 1 \ldots n \) Normalize weights
10. \( w'_i = w'_i / \eta \) Normalize weights

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

- Draw \( x'_{t−1} \) from \( Bel(x_{t−1}) \)
- Draw \( x'_t \) from \( p(x_t | x'_{t−1}, u_{t−1}) \)

Importance factor for \( x'_t \):

\[ w'_i = \frac{\text{target distribution}}{\text{proposal distribution}} = \frac{\eta \, p(z_t | x_t) \, p(x_t | x_{t−1}, u_{t−1}) \, Bel(x_{t−1})}{p(x_t | x_{t−1}, u_{t−1}) \, Bel(x_{t−1})} \propto p(z_t | x_t) \]
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 Algorithm

1. Algorithm \texttt{systematic_resampling}(S,n):
2. \( S' = \emptyset, c_1 = w^j \)
3. \textbf{For} \( i = 2 \ldots n \)
4. \( c_i = c_{i-1} + w^j \)
5. \( u_i \sim U[0,n^{-1}], i = 1 \)
6. \textbf{For} \( j = 1 \ldots n \)
7. \textbf{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. \textbf{Return} \( S' \)

Also called \textit{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

Vision-based Localization

\[ P(z|x) \]

\[ h(x) \]
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.