UNIVERSITÄT BONN

Rheinische Friedrich-Wilhelms-Universität Bonn

Institut für Informatik
Abteilung VI
Humanoid Robots Lab

Prof. Dr. Maren Bennewitz

Adresse::
Friedrich-Hirzebruch-Allee 8
53115 Bonn

Cognitive Robotics

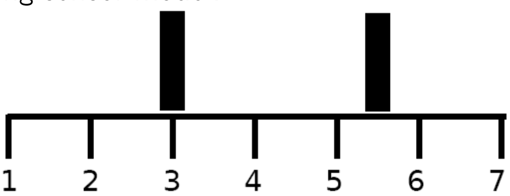
Assignment 5

Due Tuesday, November 21st, before class.

Note: If you submit your solutions via e-mail, then please submit a single PDF file containing all solutions for the exercise sheet, write your name on the first page, and name the file after your name.

5.1) Suppose your robot is equipped with a simple but noisy sensor that can detect a pole and returns the relative position of the pole. The possible measurements are: the pole is at the location to your left ($z = -1$), to your right ($z = 1$) or at your current position ($z = 0$). The maximum reading range is 1 meter, so sometimes you will not detect a pole at all ($z = \mathrm{n}$). Your robot lives in a one-dimensional world and can be at one of seven possible locations ($x \in \{1,2,\dots,7\}$), each 1 meter apart. There a two poles at positions $x = 3$ and $x = 5.5$.

You are given the following sensor model:



$p(z \mid x)$	$x = 1$	$x = 2$	$x = 3$	$x = 4$	$x = 5$	$x = 6$	$x = 7$
$z = -1$	0.0	0.00	0.25	0.50	0.0	0.5	0.0
$z = 0$	0.0	0.25	0.50	0.25	0.5	0.5	0.0
$z = 1$	0.0	0.50	0.25	0.00	0.5	0.0	0.0
$z = n$	1.0	0.25	0.00	0.25	0.0	0.0	1.0

Assume the following prior probabilities:
$p(x = 1) = 0.1$; $p(x = 2) = 0.3$; $p(x = 3) = 0.2$; $p(x = 4) = 0.2$;
$p(x = 5) = 0.2$; $p(x = 6) = 0.0$; $p(x = 7) = 0.0$

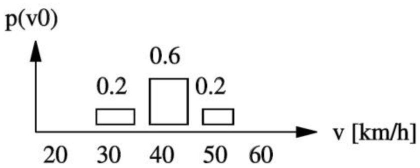
a) The robot makes a measurement $z = 1$. Update the belief distribution of the robot over the robot locations by incorporating the likelihood of the measurement.

5 points

b) The robot tries to move one meter to the right, but the drive is imprecise. It might be that it stays where it is ($p = 0.2$), that it really moves one meter to the right ($p = 0.6$) or that it actually moves two meters to the right ($p = 0.2$). Update the belief distribution that you got after part a) by applying this motion model.

5 points

5.2) Consider the following every-day situation: You help your grandma to buy some groceries. Unfortunately, her car is rather old and the speed indicator is not working any more. Since you cannot afford another speeding ticket, you have to reason about your speed using just the public speed indicators on the side of the street (see the picture below). You guess that your current speed v_0 is distributed as follows:



Of course, the acceleration is not perfect for such an old car. For each possible action, $a = -10$ (slowing down 10 km/h), $a = +10$ (accelerating by 10 km/h), $a = 0$ (keeping the speed), the transition probabilities for the speed v of your car are given in the following table.

$p(v_{i+1} \mid v_i, a_i)$	$v_{i+1} = v_i - 10$	$v_{i+1} = v_i$	$v_{i+1} = v_i + 10$
$a_i = -10$	0.7	0.3	0.0
$a_i = 0$	0.0	1.0	0.0
$a_i = +10$	0.0	0.2	0.8

The public speed indicators that provide you with speed measurements m_i have the following measurement accuracy:

	$m_i < v_i - 10$	$m_i = v_i - 10$	$m_i = v_i$	$m_i = v_i + 10$	$m_i > v_i + 10$
$p(m_i \mid v_i)$	0	0.1	0.7	0.2	0

On the ride to the supermarket, you perform the following actions and obtain the following measurements. Each measurement m_i is obtained after the according action a_i has had its effect on the speed.

time i	1	2	3
action a_i	-10	0	$+10$
measurement m_i	40	50	50

Please use the Bayesian Filtering technique to calculate your belief about the car speed after each time step i.

6 points

5.3) Particle filters use a set of weighted state hypotheses, which are called particles, to approximate the true state x_t of the robot at every time step t. Think of three different techniques to obtain a single state estimate $\overline{x_t}$ given a set of N weighted samples:

$$S_t = \left\{ x_t^{[i]}, w_t^{[i]} \;\middle|\; i = 1, \ldots, N \right\}$$

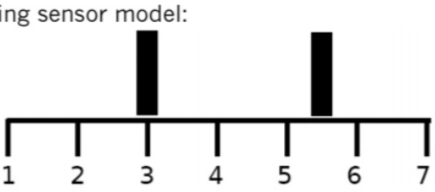
2 points

5.4) The samples are weighted according to the observation model. Provide a derivation that explains that.

2 points

5.1) Suppose your robot is equipped with a simple but noisy sensor that can detect a pole and returns the relative position of the pole. The possible measurements are: the pole is at the location to your left ($z = -1$), to your right ($z = 1$) or at your current position ($z = 0$). The maximum reading range is 1 meter, so sometimes you will not detect a pole at all ($z = \mathrm{n}$). Your robot lives in a one-dimensional world and can be at one of seven possible locations ($x \in \{1,2,\dots,7\}$), each 1 meter apart. There a two poles at positions $x = 3$ and $x = 5.5$.

You are given the following sensor model:



$p(z \mid x)$	$x = 1$	$x = 2$	$x = 3$	$x = 4$	$x = 5$	$x = 6$	$x = 7$
$z = -1$	0.0	0.00	0.25	0.50	0.0	0.5	0.0
$z = 0$	0.0	0.25	0.50	0.25	0.5	0.5	0.0
$z = 1$	0.0	0.50	0.25	0.00	0.5	0.0	0.0
$z = n$	1.0	0.25	0.00	0.25	0.0	0.0	1.0

Assume the following prior probabilities:
$p(x = 1) = 0.1$; $p(x = 2) = 0.3$; $p(x = 3) = 0.2$; $p(x = 4) = 0.2$;
$p(x = 5) = 0.2$; $p(x = 6) = 0.0$; $p(x = 7) = 0.0$

a) The robot makes a measurement $z = 1$. Update the belief distribution of the robot over the robot locations by incorporating the likelihood of the measurement.

5 points

Sensor model: $P(z|x)$

Prior: $P(x)$

$z = 1$

$$\underbrace{bel(x)}_{\text{New belief}} = P(z|x) \cdot \underbrace{bel(x)}_{\text{Prior}} = P(z=1|x) \cdot \underbrace{P(x)}_{\text{Prior}} = \underbrace{Q(x)}_{\text{New belief}}$$

→ Actual values calculated are in .ipynb file

b) The robot tries to move one meter to the right, but the drive is imprecise. It might be that it stays where it is ($p = 0.2$), that it really moves one meter to the right ($p = 0.6$) or that it actually moves two meters to the right ($p = 0.2$). Update the belief distribution that you got after part a) by applying this motion model.

5 points

Motion Model : $P(x_t | u_t, x_{t-1})$

Prior : $bel(x_{t-1})$

$$P(x_t | u_t, x_{t-1}) = [0.2, 0.6, 0.2]$$

2m right

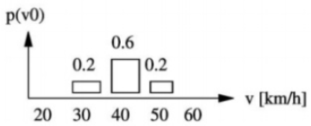
1m right

stays

$$\overline{bel}(x_t) = \sum_{x_{t-1}} P(x_t | u_t, x_{t-1}) \cdot bel(x_{t-1})$$

→ Actual values calculated are in .ipynb file

5.2) Consider the following every-day situation: You help your grandma to buy some groceries. Unfortunately, her car is rather old and the speed indicator is not working any more. Since you cannot afford another speeding ticket, you have to reason about your speed using just the public speed indicators on the side of the street (see the picture below). You guess that your current speed v_0 is distributed as follows:



Of course, the acceleration is not perfect for such an old car. For each possible action, $a = -10$ (slowing down 10 km/h), $a = +10$ (accelerating by 10 km/h), $a = 0$ (keeping the speed), the transition probabilities for the speed v of your car are given in the following table.

$p(v_{i+1} \mid v_i, a_i)$	$v_{i+1} = v_i - 10$	$v_{i+1} = v_i$	$v_{i+1} = v_i + 10$
$a_i = -10$	0.7	0.3	0.0
$a_i = 0$	0.0	1.0	0.0
$a_i = +10$	0.0	0.2	0.8

The public speed indicators that provide you with speed measurements m_i have the following measurement accuracy:

	$m_i < v_i - 10$	$m_i = v_i - 10$	$m_i = v_i$	$m_i = v_i + 10$	$m_i > v_i + 10$
$p(m_i \mid v_i)$	0	0.1	0.7	0.2	0

On the ride to the supermarket, you perform the following actions and obtain the following measurements. Each measurement m_i is obtained after the according action a_i has had its effect on the speed.

time i	1	2	3
action a_i	-10	0	$+10$
measurement m_i	40	50	50

Please use the Bayesian Filtering technique to calculate your belief about the car speed after each time step i.

6 points

u_{t-1} → control executed at $(t-1)$ timestamp from V_{t-1}

Given: Prior: $bel(V_0)$

Motion Model: $P(V_t \mid u_{t-1}, V_{t-1})$

Observation Model: $P(m_t \mid V_t)$

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

$$bel(x_t) = \eta\, p(z_t \mid x_t)\, \overline{bel}(x_t)$$

→ Instead of $\int$ we use Σ for Discrete Bayes Filter

→ Implementation and calculations are in .ipynb file

5.3) Particle filters use a set of weighted state hypotheses, which are called particles, to approximate the true state x_t of the robot at every time step t. Think of three different techniques to obtain a single state estimate $\overline{x_t}$ given a set of N weighted samples:

$$S_t = \left\{ x_t^{[i]}, w_t^{[i]} \,\middle|\, i = 1, \dots, N \right\}$$

2 points

1 Weighted average

$$X_t = \frac{1}{\sum W_j} \cdot \sum X_j \cdot W_j$$

2 Argmax of W_j

$$X_t = \underset{X_j}{\operatorname{argmax}} \{W_j\}$$

3 Resampling + Averaging

1) Resample X_j based on W_j

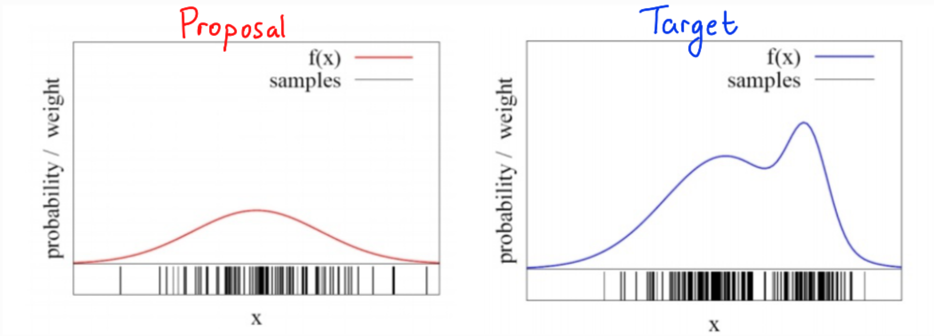
2) Take average state of that new list of states

$$X_t = \frac{1}{N} \sum_{1}^{N} X_j$$

5.4) The samples are weighted according to the observation model. Provide a derivation that explains that.

2 points

Let's start from two distributions, as shown on the image below:



Proposal distribution is Gaussian distribution

Target distribution is non-Gaussian distribution

Sampling state "x" from Gaussian distribution is possible, and it is done in following way:

$$\boxed{x \leftarrow \frac{1}{2}\sum_{i=1}^{12} \text{rand}(-\sigma, \sigma)}$$

→ This sampled "x" is called particle

→ Particles approximate function in the way that:
The more particles fall into a region, the higher the probability of the region

→ Since target is not Gaussian, we cannot sample from it using some closed-form solution

→ This means, in order to approximate target distribution with set of particles, we need a way to sample from it

→ This is possible, in a following way:

1) Sample from Gaussian (i.e. proposal) ⇒ X_j

2) Weight sample using target distribution and proposal distribution:

$$W_j = \frac{\text{target}(x_j)}{\text{proposal}(x_j)}$$

3) Create list of (X_j, W_j) pairs

4) Resample X_j with probability / importance / weight W_j of being drawn

→ This can be used for robot localization

→ We will sample from motion model (which we suppose that it is Gaussian distribution) X_j

→ Add weight to X_j: $W_j \propto P(Z_j \mid x_j)$

→ This will ensure that, during resampling process, particles X_j who are closer to true state have higher probability of being drawn, and therefore we will have more particles around true state