LT1 - Intro

The code is written using the ROS indigo framework using Python, the robots have a depth camera. We will be using SVN version control.

It's worth having reporty people in your team.

Exercises

Ex 1. Particle filter

- 30 Marks
- Due 11th Oct
- Viva. 12th Oct

Ex 2. Your own idea

- 70 Marks
- Demo 20% - 15th Nov
- Report 80% - 8th Dec

Learning outcomes

1. Program autonomous robots
2. Implement signal processing and control algorithms
3. Describe and analyze robot processes
4. Write technical reports
5. Use experimental methods

Exercise points

All of the coursework needs to be experimentally evaluated using suitable scientific methods - How it failed? - Why did it fail? - In what circumstances does it fail? - You need to justify any choices you make - Evidence based engineering - Statistical analysis

Moravec's Paradox

- *Easy* - Mathematics, Chess, Expert systems
- *Hard* - Seeing, Conversation, Walking

What's easy for humans is hard for robots and vice versa. # LT15? - Stuff

Given how much he's been talking about advanced planning (MDP, dynamic programing etc.) should we be worried about how our project doesn't really cover any of that.

Instead of creating a context map of the world we could integrate exploration and object mapping together, so the robot can find a table without knowing if one exists at the start of the plan. Rewards would be based on how similar the object is to a table e.g. a chair is similar to a table, we could use word2vec to calculate the similarity of objects, or by similarity of the colour profiles of the image?

Q learning will eventually converge on the reward values that value iteration would calculate. There is a theorem that proves this.

Bellmans curse of dimensionality as we have to iterate over all the states.

In reality both actions and states are a continuous space # LT16 - SLAM

- Simultaneous location and mapping
- Occupancy grid representation of space
- 2 sources of error: odometry and distance sensing

Inverse sensor models

We can process our sensor readings to remove erroneous values. We do this by learning an inverse sensor model.

With Bayes we come up with an observation model, so what is the probability of the hypothesis given the data.

The only hypothesis we can have about a grid cell, is it has something in it or not. So when we apply Bayes we can reason about each grid cell independently. The cells aren't independent but this assumption simplifies the amount of computation we have to do.

Train our neural network to know the probability of occupancy by a certain object, given images

OPTICS clustering then bayesian inference on the centroids? # LT2 - Localization ## Where am I? - Knowing where you are is a key problem in robotics. - Hard in mobile platforms because - No direct way of knowing where you are Indirect methods involve unreliable data - The problem is (mostly) addressed by probabilistic frameworks.

Combining evidence

- Start with a belief (all possible locations)
- Cut down belief by combining it with new data to form a new belief
- Repeat process. reducing overall belief and hence, number of possible places for agent
- This is non-probabilistic and relies on all candidates acting independently

Combining uncertain evidence

- Instead of yes/no, return a number between [0, 1]
- This is the certainty of matching the data set
- Data is now called the “data likelihood”, in contains the likelihood of the agent being in that space
- To find a new belief: multiply current belief cell value with data cell value
- Belief is a probability distribution, add values sum to 1
- Data is a likelihood as the values don’t sum to 1
- New belief is no longer a probability distribution so needs to be converted back
- To do this add up all cell values and divide each by the sum of the cell values
- As probability accumulates, more likely areas gain higher probabilities
- This allows possibility to recover from a failed sensor (given a good sensor model)
- This is a recursive bayesian filter

Bayes’ rule

- Recall the conditional probability rule

$$P(A\&B) = P(B|A)P(A) = P(A|B)P(B) \tag{1}$$

- Rearranged this gives

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2}$$

$$P(Hypothesis|Data) = \frac{P(Data|Hypothesis)P(Hypothesis)}{P(Data)} \tag{3}$$

Bayes’ rule recap

- Now we imagine a set of possible hypotheses which are

- Mutually exclusive (one and only one can be true)
- Exhaustive (one must be true)

- In this case

$$P(Data) = \sum_{i=1}^{N} P(Data|H_i)P(H_i) \tag{4}$$

- Hence

$$P(H_i|D) = \frac{P(D|H_i)P(H_i)}{\sum_{j=1}^{N} P(D|H_i)P(H_i)} \tag{5}$$