E28 - Notes for Week of Nov. 4th

Noah Weinthal

— November 4th: Probability Continued, and the Bayes Filter. —

MATLAB Demo

Given a robot that has a door detector, we’re trying to get the robot to move through the world. That said, this is a probabilistic model, so we’re not guaranteed to sense a door even if we’re at a door. This model will be available online.

Probabilistic Filters

A Probabilistic Filter combines the probabilities of robot motion and sensing to produce a distribution over states, indicating probability of a robot being at a particular location. We formulate this based off of a **Motion Distribution** and a **Sensor Distribution** (sometimes called a measurement distribution). The motion distribution can be formally written as

\[ P(x'|x, u) \]

and the sensor distribution as

\[ P(z|x) \]

The motion distribution consists of the probability of reaching next state \( x' \) given that the robot with some applied constant control \( u \) is in state \( x \) and the sensor distribution is the probability of observing sensor reading \( z \) given the robot is in state \( x \).

The Bayes Filter

Let \( bel(x) \) be the belief that robot is in state \( x \). To construct a Bayes Filter, apply motion model by executing control \( u \) and set for each \( x' \):

\[ \bar{bel}(x') = \sum_x bel(x) \cdot P(x'|x, u) \] (1)

and apply sensor model by observing measurement \( z \) and set for each \( x \)

\[ bel(x) = \eta P(z|x) \bar{bel}(x) \] (2)
where

$$\eta = \frac{1}{\sum_{x'} P(z|x') \text{bel}(x')}$$

$\text{bel}(x)$ is intuitively just $P(x)$, but there is a subtle difference in formality. Actually, $\text{bel}(x)$ is $P(x_{t+1}|u_1...u_t, z_1...z_t)$, or in plainer terms the probability that we are in a particular state given all the controls and sensor observations up to that point in time. Note that we don’t know explicitly what $P(z)$ is from this. $\eta$ normalizes this to make sure we end up with a proper probability distribution (i.e. everything sums to one).

**Runtime considerations**

The spaces of all possible states can get rather large, depending on the resolution and size of the mapped space. For $n$ states and $m$ sensor readings, (1) is $O(n^2)$ and (2) is $O(n)$.

— November 6th: Bayes Filter Continued, and More Probability —

**Example motion and measurement models**

The motion model for the MATLAB example was

$$P(x'|x, u) = \begin{cases} 
12.5\% & : \text{if } \pm 1 \text{ off} \\
75\% & : \text{otherwise}
\end{cases}$$

while the measurement model was (less compactly):

$$P(z = 1|\text{door}) = 95\%$$

$$P(z = 0|\text{door}) = 5\%$$

$$P(z = 1|\text{no door}) = 10\%$$

$$P(z = 0|\text{no door}) = 90\%$$

**More on probability**

**Random sampling**

Suppose we’re given a function `rand()` which returns an integer from 0 to a large RAND_MAX with equal probability. We want to use `rand` to implement `frand()` which returns a floating point number uniformly distributed in [0,1].

```python
def frand():
    return float(rand())/RAND_MAX
```

2
which returns 1 of possible numbers between 0 and 1. Suppose we want to implement an unfair coin flip such that

\[
P(\text{heads}) = 0.7 \\
P(\text{tails}) = 0.3
\]

We can use \texttt{frand()} to implement:

```python
def unfair_flip():
    if frand() > 0.3:
        return 'HEADS'
    else:
        return 'TAILS'
```

we can do the same with an unfair die for the following distribution:

```
def unfair_dice_roll():
    p = frand()
    if p < 1.0/21:
        return 1
    elif p < 3.0/21:
        return 2
    elif p < 6.0/21:
    ```
Which is essentially searching the sorted list \([1/21, 3/21, \ldots, 21/21]\) for the smallest element greater than or equal to \(P\).

Discrete vs. Continuous Probability

Discrete has:

\[
x \in X \\
0 \leq P(x) \leq 1 \\
\sum P(x) = 1
\]

while Continuous has:

\[
x \in \mathbb{R} \\
P(x) \geq 0 \\
\int_{-\infty}^{\infty} P(x)dx = 1
\]

A continuous distribution is not the same as a discrete probability, despite similar notation. We’ve mostly been dealing with the discrete case up until here. To find the probability that \(x \in [a, b]\) we have to integrate the distribution along that interval.