Statistical Techniques in Robotics (16-831, F09) Lecture #2 (Thursday August 27)

Filtering Theory

Lecturer: Drew Bagnell *Scribe: Supreeth Achar*

1 Filtering Problems

1.1 Overview

The goal of filtering is to maintain an online estimate of the state (x_t) of a system. In a localization problem this state would be the pose of the robot, in mapping the state would be a map of the environment and in SLAM the state would be both the pose and map. A general filter would have the following inputs

- A timestep indexed stream of data (d) containing observations ($z_{1:t}$) and actions ($u_{1:t}$) : $d = \{u_{1:t}, z_{1:t}\}$
- A probability distribution modeling the initial state : $p(x_0)$
- Action Model or Motion model
$$p(x_t|x_{t-1}, u_t) \tag{1}$$
- Measurement model or Observation model
$$p(z_t|x_t) \tag{2}$$

At each timestep we want the filter to output a probability distribution of the current state x_t given all available information $d = \{u_{1:t}, z_{1:t}\}, p(x_0)$. This is the distribution $p(x_t|u_{1:t}, z_{1:t}, x_0)$ and is denoted by the shorthand notation $Bel(x_t)$.

Filters are defined recursively, with the output at time instant t being an input to the filter at $t+1$.

1.2 Other Notations

Filtering of the form described here has applications in many different domains including robotics, electrical engineering and computer science. As a result, notations and terminologies used can vary. State (x) may be denoted by s or if the state being estimated is a map, m might be used. Action u is also called the control input (u) or the decision (a). Measurements (z) are also called observations and may be represented by y, o or x. The motion model is also called the state transition function or the plant dynamics.

2 Markov Assumptions

The amount of measurement and motion data ($z_{1:t}$ and $u_{1:t}$ grows as time progresses. Using all of this data to calculate x_t at each filter timestep is not feasible. To overcome this difficultly, it is assumed that the observation model and action model are Markovian which means that measurements made and state updates are independent of past states ($x_{1:t-1}$) if the current state (x_t) is known

- Observation model Markov assumption

$$p\left(z_t|x_{0:t}, u_{1:t}, z_{1:t-1}\right) = p\left(z_t|x_t\right) \tag{3}$$

- Action model Markov assumption

$$p\left(x_{t+1}|x_{0:t}, u_{1:t}, z_{1:t-1}\right) = p\left(x_{t+1}|x_t, u_t\right) \tag{4}$$

For a robot, the observations are dependent on the map (m), for simplicity this dependence is not shown explicitly. These assumptions seem justified but may not always be true in practice. For example the Markovian motion model may be innaccurate if the dynamics of the state are not included. If the actual environment differs from the map used to generate expected observations then the Markov assumption for the observation model does not hold. Even so, these Markovian approximations are important as they are practical and make computation tractable.

3 Derivation of the Bayes Filter

This derivation gives an expression for the state distribution at time t, $Bel(t)$ in terms of the distribution at time $t-1$ $Bel(t-1)$

$$\begin{aligned}
Bel(x_t) &= p(x_t|z_{1:t}, u_{1:t}) && \\
&= \eta\, p(z_t|x_t, z_{1:t-1}, u_{1:t})\, p(x_t|u_{1:t}, z_{1:t-1}) && \text{Bayes' Rule} \\
&= \eta\, p(z_t|x_t) p(x_t|u_{1:t}, z_{1:t-1}) && \text{Markov Assumption} \\
&= \eta\, p(z_t|x_t) \int \left[p(x_t|x_{t-1}, u_{1:t}, z_{1:t-1})\, p(x_{t-1}|u_{1:t}, z_{1:t-1})\right]\, dx_{t-1} && P(A) = \textstyle\int P(A|B)P(B)dB \\
&= \eta\, p(z_t|x_t) \int \left[p(x_t|x_{t-1}, u_t)\, p(x_{t-1}|u_{1:t-1}, z_{1:t-1})\right]\, dx_{t-1} && \text{Markov Assumption} \\
&= \eta\, p(z_t|x_t) \int \left[p(x_t|x_{t-1}, u_t)\, Bel(x_{t-1})\right]\, dx_{t-1} && \text{Definition of } Bel(x_t)
\end{aligned}$$

4 Example: Piecewise Constant Filter

Figure 1 shows how a Bayes Filter can be used for robot localization. Consider an imaginary robot that moves in one dimension along the length of corridor and has a sensor that tells it whether it is in front of a door or not. The robot also has a 'map', a list of positions at which doors are located along the corridor. The robot starts of with no clue to where it is on the corridor, so the initial pose belief is a uniform distribution as shown in 1a. As door observations are made, the belief distribution forms peaks which tend to flatten out as the robot moves.

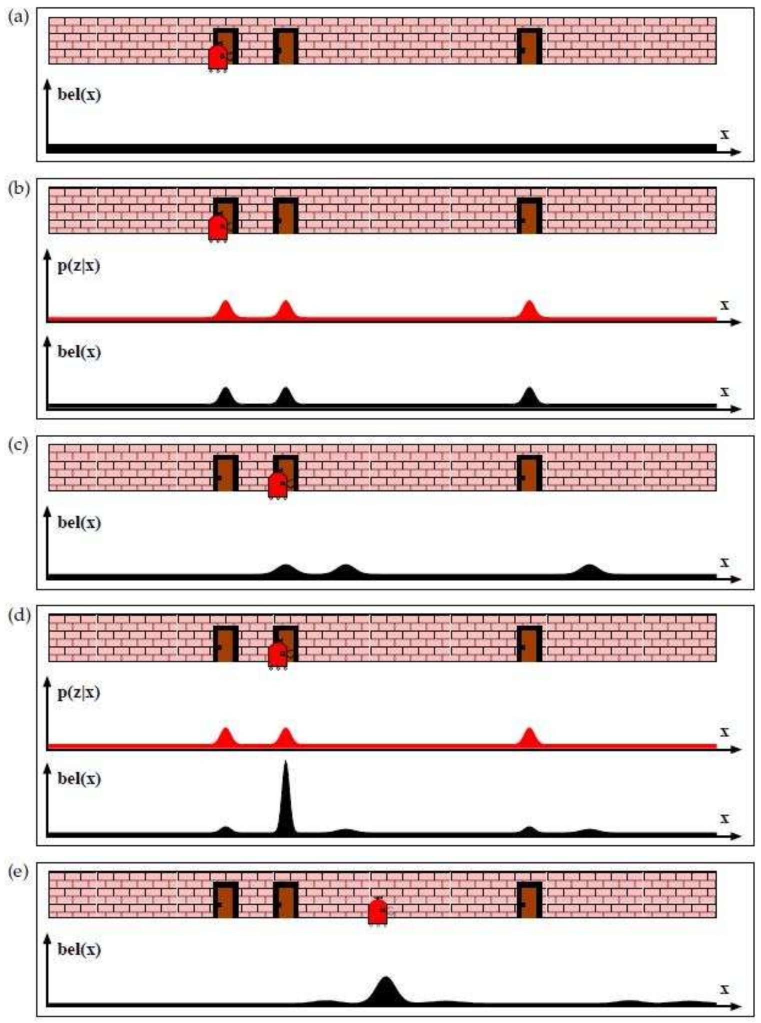

Figure 1: Grid localization

5 Implementing a Bayes Filter

Implementing a Bayesian recursive filter for continuous state variables as shown above is usually not possible computationally. The probability distributions desribed can in general take on any form and so closed form solutions do not exist to the integrals. A number of different assumptions and approximations can be made to design a filter based on Bayesian recursive estimation. Some examples of practical techniques based on the Bayes Filter are

1. Kalman Filters (KF, EKF,UKF etc)
2. Histogram Filters
3. Particle Filters
4. Hidden Markov Models (HMMs) - These are just discrete Bayes Filters

6 Markov Localization (Grid-based Localization)

Markov localization uses a histogram filter. The basic idea is to discretize the (continuous) state space into a number of bins. Each of these bins has a probability mass associated with it. If the number of bins is sufficient, the probability mass distribution over these bins is an accurate approximation of the underlying distribution in the state space. The integrals in the Bayes Filter are replaced with summations which are easier to compute and which need to be calculated over a finite number of cells.

For Markov localization, the state space is the pose of the robot. This space is quantized into a rectangular grid and each cell (k) in the grid has a probability mass ($p_{k,t}$) associated with it that equals the probability of the robot being somewhere within that cell in the pose space. Motion causes the pose distribution to shift and flatten out due to the increased uncertainity in position while measurements tend to create peaks in the distribution. Figure 2 shows an example of Markov Localization, a likelihood estimate is maintained for all the cells. The number of cells grows exponentially in the number of dimensions, so for high dimensional states this method is not practical.

Algorithm 1 grid_based_localizer($\{p_{k,t-1}\}, u_t, z_t, m$)

```
1: for all $k$ do
2:     $p'_{k,t} = \sum_i p_{i,t-1}$ motion_model($x_k, u_t, x_i$)
3:     $p_{k,t} = \eta p'_{k,t}$ measurement_model($z_t, x_k, m$)
4: end for
5: return $\{p_{k,t}\}$
```

7 Forward Sensor Models

Bayesian Filtering techniques for robot state estimation require some probability density function that gives the likelihood of a measurement from a certain pose. This is $p(z_t|x_t, m)$ and is called the forward sensor model. A typical range finding sensor like a laser will return a number of measurements in a single scan $z_t = \{z_t^1, z_t^2, \ldots, z_t^K\}$. In reality measurements in a scan will depend on each other, but modeling this dependency is complicated so it will be assumed that scan readings are independent although this can lead to overconfident likelihoods.

$$p(z|x,m) = \prod_{k=1}^{K} p\left(z^k|x,m\right) \tag{5}$$

There are a number of sources of error in measurements (See Figure 3). These are

- Readings from objects not present in the map (people, furniture etc)
- Noise around a 'true' measurement
- 'Random' measurements
- Sensor returning maximum range

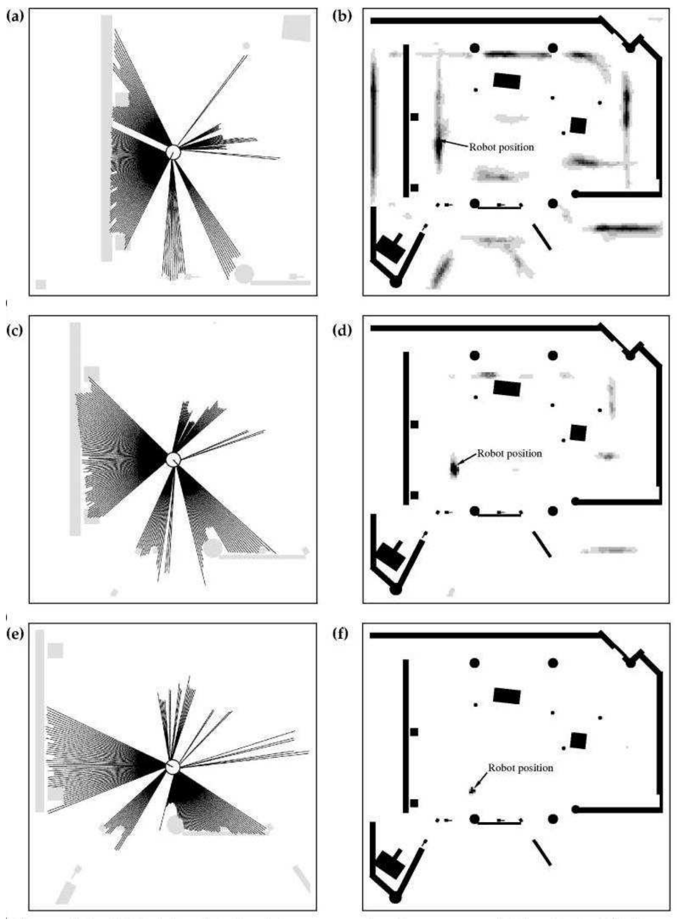

Figure 2: Localization in a Museum

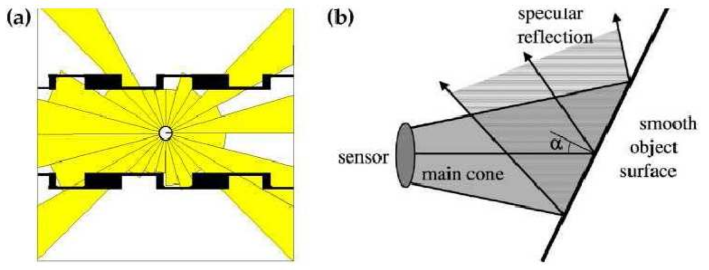

Figure 6.1 (a) Typical ultrasound scan of a robot in its environment. (b) A misreading in ultrasonic sensing. This effect occurs when firing a sonar signal towards a reflective surface at an angle α that exceeds half the opening angle of the sensor.

Figure 3: Typical ultrasound errors

We would like to develop a model for the proximity sensors used that accounts for the various sources of noise. It will incorporate four types of measurement errors shown in Figure 4:

1. Local measurement noise: Accounts for non-idealities in the measurement process (a)

2. Unexpected obstacles: Accounts for dynamic objects which are not represented in the static map (b)

3. Max-range measurement: In the event that the sensor 'misses' the target and returns a max-range measurement (c)

4. Random measurements: When the sensor is *really* having a bad day... (d)

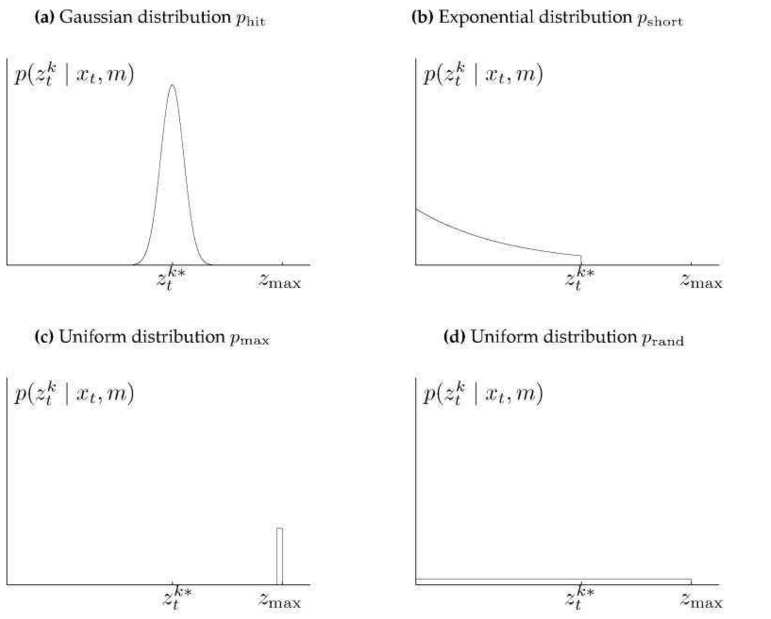

Figure 6.3 Components of the range finder sensor model. In each diagram the horizontal axis corresponds to the measurement z_t^k, the vertical to the likelihood.

Figure 4: Components of error model

We can use a linear combination of the distributions to get a "Pseudo-density" distribution (Figure 5). The mixing coefficients should sum to one and can be determined experimentally by collecting ground truthed range data and finding the coefficients that provide the best fit.

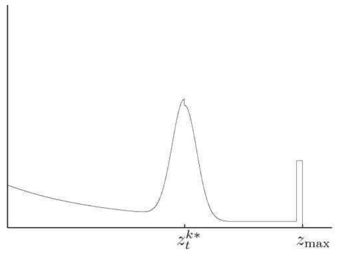

Figure 6.4 "Pseudo-density" of a typical mixture distribution $p(z_t^k \mid x_t, m)$.

Figure 5: Combined error model

8 From Localization to Mapping

Instead of doing localization, we will try to solve the opposite problem: Given the position of the robot, find the map of the world.

8.1 Occupancy mapping

We will partition the world into cells, with each being one of two states: filled or not. The individual grid cells are m_i, and $\vec{m}$ is the vector of all grid cells. Unfortunately, the curse of dimensionality prevents us from filtering $\vec{m}$. Instead, we will filter each cell independently, assuming that they are in fact independent. In truth, this is a very bad assumption to make, but it is practical.

8.2 Derivation

Let x represent the state of grid cell m_i (essentially $x \leftarrow m_i$).

$$\begin{aligned} p(x|z_{1:t}) &= \frac{p(z_t|x)p(x|z_{1:t-1})}{p(z_t|z_{1:t-1})} \\ &= \frac{p(x|z_t)p(z_t)}{p(x)} \frac{p(x|z_{1:t-1})}{p(z_t|z_{1:t-1})} \end{aligned}$$

Here we need to use a trick because we have many nasty terms that are hard to calculate. We will start by computing (by analogy) the likelihood for the opposite event $\bar{x}$

$$p(\bar{x}|z_{1:t}) = \frac{p(\bar{x}|z_t)p(z_t)}{p(\bar{x})} \frac{p(\bar{x}|z_{1:t-1})}{p(z_t|z_{1:t-1})}, \tag{6}$$

and then divide the two:

$$\frac{p(x|z_{1:t})}{p(\bar{x}|z_{1:t})} = \left[\frac{p(x|z_t)}{p(\bar{x}|z_t)}\right] \left[\frac{p(x)}{p(\bar{x})}\right] \left[\frac{p(x|z_{1:t-1})}{p(\bar{x}|z_{1:t-1})}\right] \tag{7}$$

Taking the log of (7) gives us the log odds ratio of the belief. We can use this to turn a series of multiplication and divisions into additions and subtractions (see page 96).