Statistical Techniques in Robotics (16-831, F08) Lecture #2 (Thursday August 28)

Filtering Theory

Lecturer: Drew Bagnell *Scribe: Felix Duvallet*

1 Filtering Problems

1.1 Givens

Here is what we have:

- Stream of data: $d = \{u_1, z_2, u_2, z_2, \dots, u_t, z_t\}$
- Inititial distribution over states: $p(x_0)$
- Motion model
$$p(x_t|x_{t-1}, u_t) \tag{1}$$
- Observation model
$$p(z_t|x_t) \tag{2}$$

1.2 Compute

And this is what we want:

- Posterior state distribution
$$Bel(x_t) = p(x_t|d) \tag{3}$$

NOTE: We never actually get states (x_i) given to us, but since this is a recursive process having $p(x_0)$ is enough.

1.3 Markov Assumptions

Both of these assume that past and future data are independent if one knows the current state.

- Observation model Markov assumption
$$p\left(z_t|x_{0:t}, u_{1:t}, z_{1:t-1}\right) = p\left(z_t|x_t\right) \tag{4}$$
- Transition/Action model Markov assumption
$$p\left(x_t|x_{0:t-1}, u_{1:t}, z_{1:t-1}\right) = p\left(x_t|x_{t-1}, u_t\right) \tag{5}$$

Note that these assumption are not always true (for example if we have unmodeled dynamics not included in the state), but they are practical and make computation tractable.

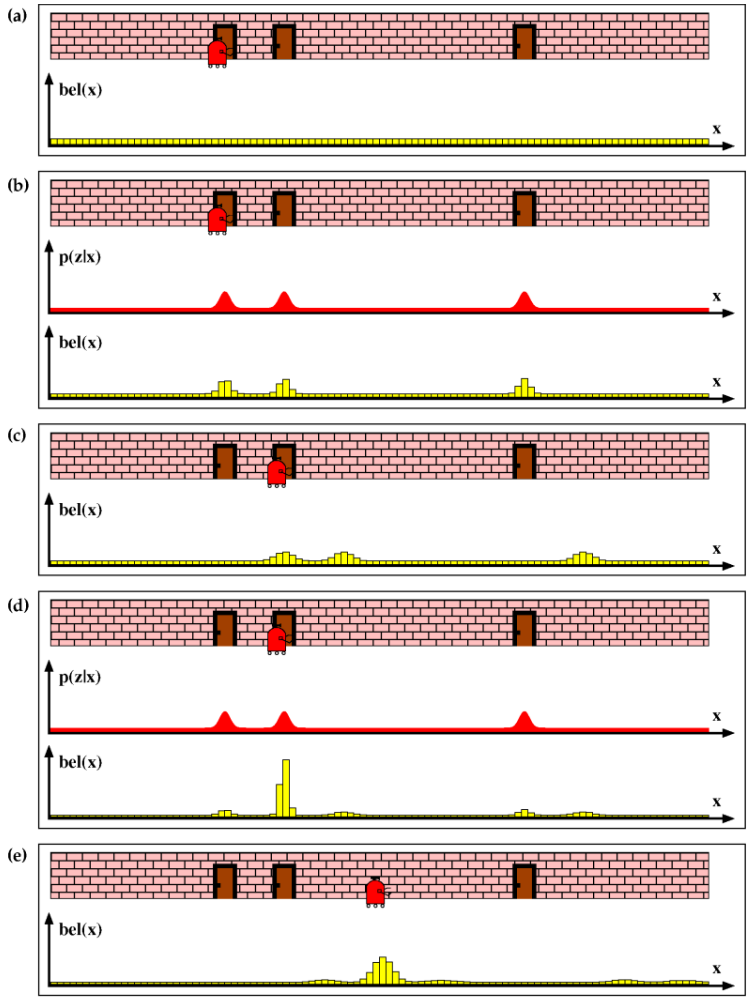

Figure 8.1 Grid localization using a fine-grained metric decomposition. Each picture depicts the position of the robot in the hallway along with its belief $bel(x_t)$, represented by a histogram over a grid.

Figure 1: Grid localization

2 Derivation of the Bayes Filter

In this derivation, we assume that we correctly initialized the prior belief ($Bel(x_0)$). It also requires that controls be chosen at random.

$$
\begin{aligned}
Bel(x_t) &= p(x_t|z_{1:t}) && \\
&= \eta\, p(z_t|x_t, z_{1:t-1})\; p(x_t|z_{1:t-1}) && \text{Bayes' Rule} \\
&= \eta\, p(z_t|x_t)\; p(x_t|z_{1:t-1}) && \text{Markov Assumption} \\
&= \eta\, p(z_t|x_t) \int [p(x_t|z_{1:t-1}, x_{t-1})\; p(x_{t-1}|z_{1:t-1})]\; dx_{t-1} && \text{Law of Total Probability} \\
&= \eta\, p(z_t|x_t) \int [p(x_t|x_{t-1})\; Bel(x_{t-1})]\; dx_{t-1} && \text{Markov Assumption}
\end{aligned}
$$

3 Bayes Filter as Algorithm

Here is the Bayes Filter implemented as an algorithm:

Algorithm 1 filter($Bel(x)$, data):

if data is a perception (z) **then**
  for all x **do**
    $Bel'(x) = p(z|x)Bel(x)$
    $\eta = \eta + Bel'(x)$
  end for
  for all x **do**
    $Bel'(x) = \frac{1}{\eta}Bel'(x)$
  end for
else if data is an action (u) **then**
  $Bel'(x) = \sum_{\hat{x}} Bel(\hat{x})\, p\,(x|action, \hat{x})$
end if
return $Bel'(x)$

4 Example: Piecewise Constant Filter

Figure 1 shows a simple 1D example. A sensor tells the robot the state of the door. Note that the belief begins as a uniform distribution, and then changes as observations and actions are processed.

5 Markov Localization (Grid-based Localization)

In this simplistic implementation shown in Figure 2, one keeps track of the likelihood for all possible states. As the robot moves, the uncertainty in position is reduced.

NOTE: This quickly becomes intractable as the number of dimensions increases.

6 Forward Sensor Models

- Beam-based sensors (lasers, sonars, stereo vision)
- Observations: A scan z consists of K measurements:

$$z = \{z_1, z_2, \ldots, z_K\} \tag{6}$$

- Likelihood of observation (given map m):

$$p(z|x, m) = \prod_{k=1}^{K} p\,(z_k|x, m) \tag{7}$$

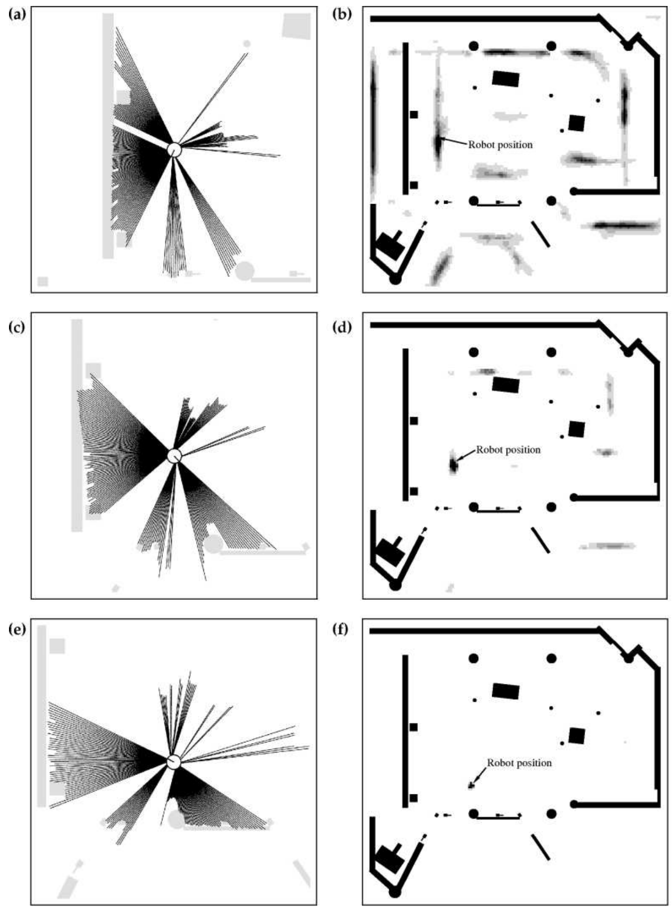

Figure 8.6 Global localization in a map using laser range-finder data. (a) Scan of the laser range-finders taken at the start position of the robot (max range readings are omitted). Figure (b) shows the situation after incorporating this laser scan, starting with the uniform distribution. (c) Second scan and (d) resulting belief. After integrating the final scan shown in (e), the robot's belief is centered at its actual location (see (f)).

Figure 2: Localization in a Museum

- NOTE: The individual measurements are independent given the position. This assumption is not generally true, as this can lead to overconfident likelihoods, but it is (once again) practical for now.
- Sources of measurement error (See Figure 3):
  - Beams reflected by obstacles
  - People
  - Random measurements
  - Max range measurements

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)

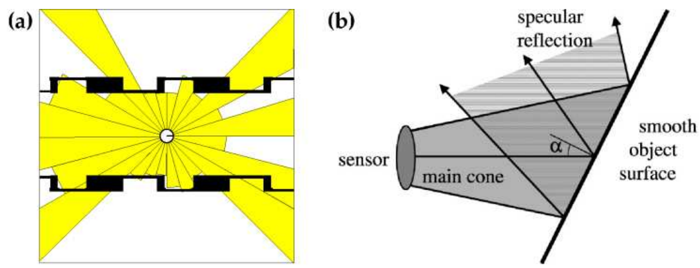

Figure 6.1 (a) Typical ultrasound scan of a robot in its environment. (b) A misreading in ultrasonic sensing. This effect occurs when firing a sonar signal towards a reflective surface at an angle α that exceeds half the opening angle of the sensor.

Figure 3: Typical ultrasound errors

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)

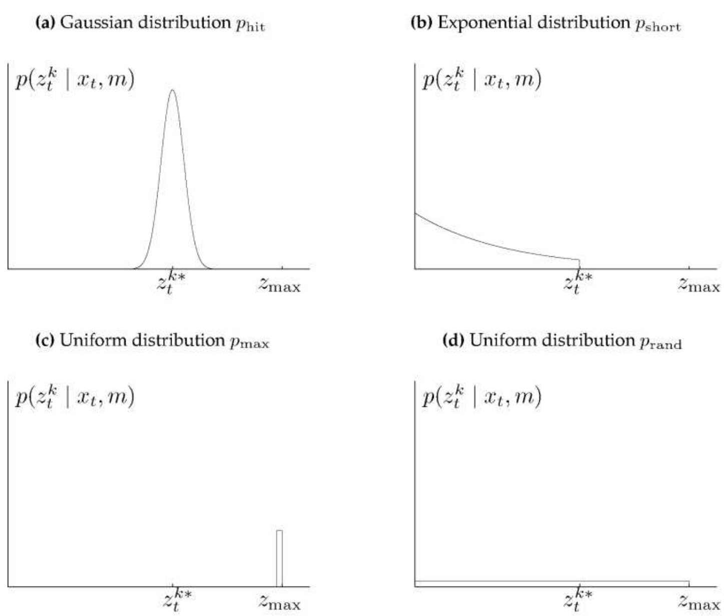

Figure 6.3 Components of the range finder sensor model. In each diagram the horizontal axis corresponds to the measurement z_t^k, the vertical to the likelihood.

Figure 4: Components of error model

We can do a linear combination of the distributions to get a "Pseudo-density" distribution (Figure 5).

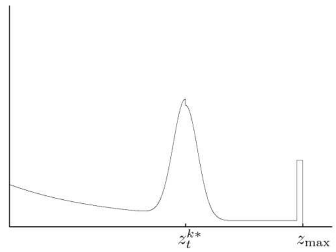

Figure 6.4 "Pseudo-density" of a typical mixture distribution $p(z_t^k \mid x_t, m)$.

Figure 5: Combined error model

7 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.

7.1 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.

7.2 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 $\bar{x}$

$$p(\bar{x}|z_{1:t}) = \frac{p(\bar{x}|z_t)p(z_t)}{p(\bar{x})} \frac{p(\bar{x}|z_{1:t-1})}{p(z_t|z_{1:t-1})}, \tag{8}$$

and then divide the two:

$$\frac{p(x|z_{1:t})}{p(\bar{x}|z_{1:t})} = \left[\frac{p(x|z_t)}{p(\bar{x}|z_t)}\right] \left[\frac{p(\bar{x})}{p(x)}\right] \left[\frac{p(x|z_{1:t-1})}{p(\bar{x}|z_{1:t-1})}\right] \tag{9}$$

Taking the log of (9) 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).