Statistical Techniques in Robotics (16-831, F08) Lecture #3 (September 2, 2008)

Filtering Theory

Lecturer: Drew Bagnell *Scribe: Alex Styler*

1 Occupancy Mapping

1.1 Log-odds of the Belief

From last time:

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

The next step is to take the log of (1) to get the log odds of the belief $bel_t(x)$, denoted as $l_t(x)$:

$$l_t(x) = l_{t-1}(x) + log\frac{p(x|z_t)}{p(\bar{x}|z_t)} + log\frac{p(\bar{x})}{p(x)} \tag{2}$$

The $\bar{x}$ in the numerator of (2) seems contrary to intuition, but if you rearrange the terms:

$$l_t(x) = l_{t-1}(x) + log\frac{p(x|z_t)}{p(x)} + log\frac{p(\bar{x})}{p(\bar{x}|z_t)} \tag{3}$$

it makes more sense. If the probability of $\bar{x}$ decreases with the observation z_t: $p(\bar{x}) > p(\bar{x}|z_t)$, it causes the log odds of the belief to increase.

1.2 Inverse Beam Sensor Model

Figure 1 (c) shows the typical result from a ray trace in the inverse beam sensor model. All cells start gray: let's say $p(m_i) = 0.5$. After the observation, cells penetrated by the beam decrease their probability to contain an obstacle, cells that reflect the beam increase their probability, and cells beyond the reflection do not update as no information was gained about their state.

1.3 Problems With Occupancy Mapping

Occupancy mapping has a few problems:

- Non-independence of individual readings
- Cells might be 'partially' filled, implying the districtized state model is bogus
- Cells aren't naturally independent, especially when dealing with walls and large obstacles

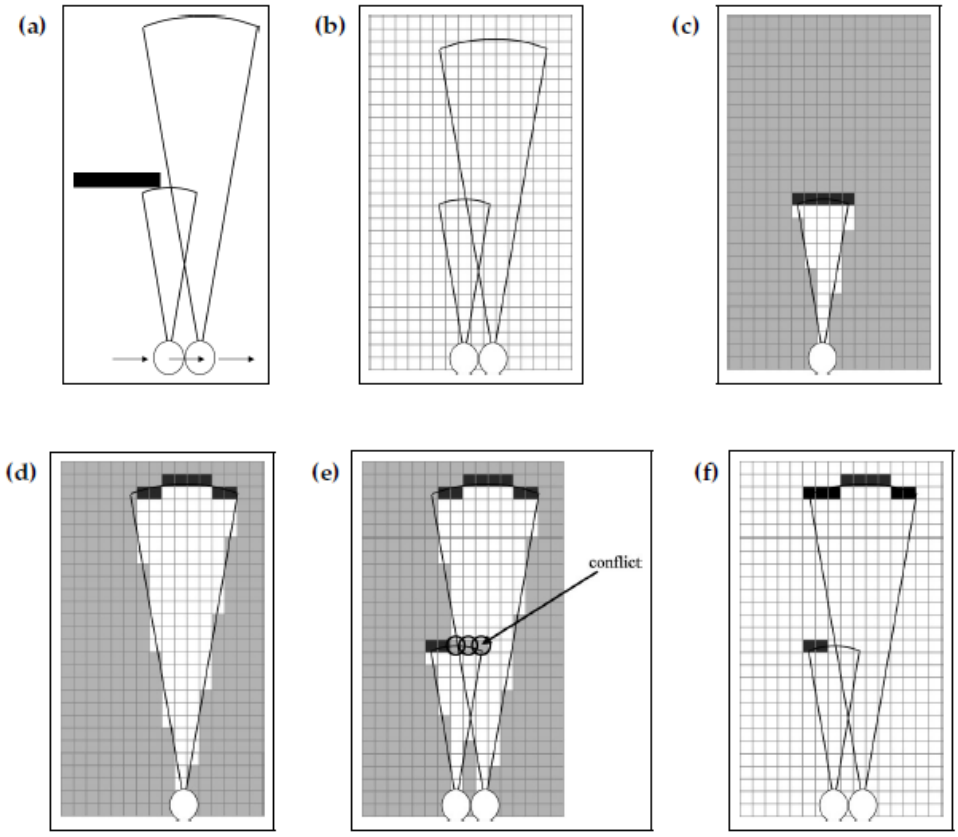

Figure 1: Examples of inverse sensor model

- Likelihood doesn't factorize over cells

The problem of partially filled cells can be handled using Density Mapping ...

2 Density Mapping

2.1 Why Density Mapping?

Consider a districtized world with a robot and a tree. Let's consider one cell to be 10% filled by a twig from the tree. In binary occupancy mapping, the cell might start with a $p(x_i) = 0.5$. After 10 measurements, in which 9 penetrate the cell and only 1 reflects, the binary occupancy mapping will quickly reach the conclusion that the cell is empty. Assuming the cell is empty could be catastrophic for some robots. Density mapping is different, and tries to predict "'how full"' a cell is using a real-number density instead of a binary full/empty state.

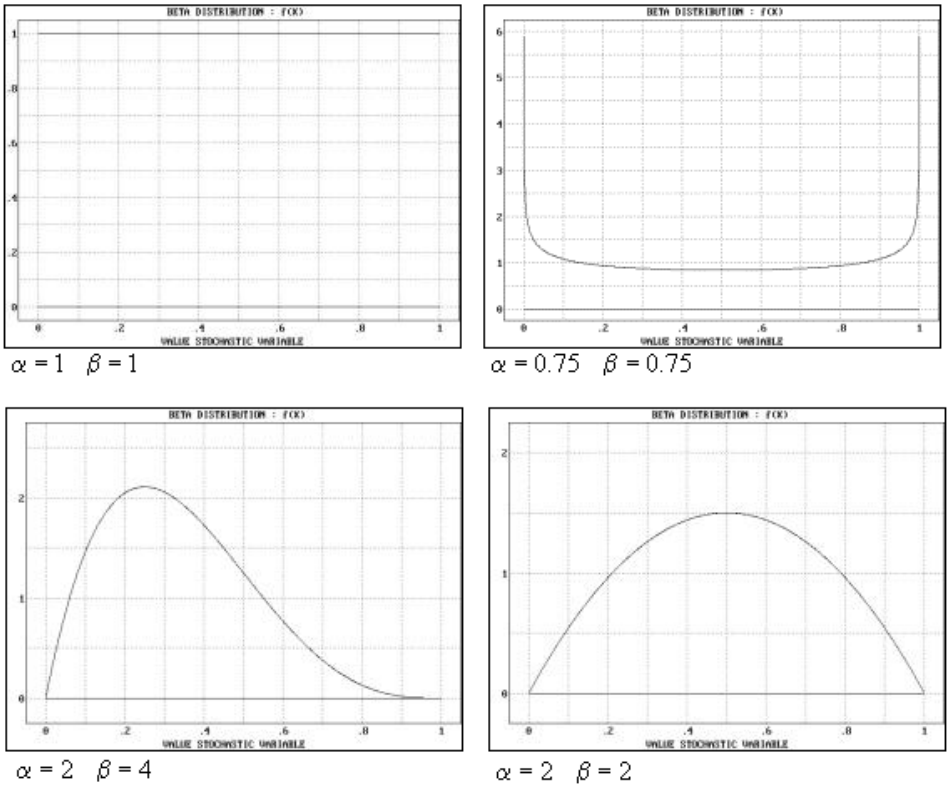

Figure 2: Examples of Beta Distributions

2.2 The Beta Distribution

Density mapping typically uses a beta distribution to keep track of a cell's density. Some example beta distributions are shown in Figure 2. Initial parameters are tuned to the expected environment: maybe having every cell weighted to be more likely empty or low density if obstacles are sparse. The beta-distribution is represented as:

$$p(x|\alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{z} \tag{4}$$

Where z is an arbitrary normalizing constant to make the integral evaluate to 1.

2.3 The Bayes Update

We represent the probability of a beam hitting/reflecting off an obstacle in a cell of a world with a simple Bernoulli distribution:

$$p(O_t^i = hit) = bernoulli(x^i) = x^i \tag{5}$$

$$p(O_t^i = passthru) = 1 - x^i \tag{6}$$

Given a hit, what happens when we use the Bayes filter to update the prior with the likelihood?

$$\begin{aligned} Belief &= \frac{Prior * Likelihood}{Normalizer} \\ Belief &= \left[\frac{x^{\alpha-1}(1-x)^{\beta-1}}{z}\right][x] \\ Belief &= \left[\frac{x^{(\alpha+1)-1}(1-x)^{\beta-1}}{z}\right] \end{aligned}$$

So, given a hit, all we need to do to update the prior is increment α. The same can be shown that for a passthrough, β is incremented. As one is repeatedly incremented, the distribution skews in the appropriate direction and sharpens.

2.4 Problems

A problem with density mapping is that sensor noise is not accounted for. This can be mitigated by throwing away extremely unlikely measurements, but is a little bit of a hack.

The problem with all districtized mapping is that the computation required increases exponentially with respect to the number of dimensions. A few might be reasonable but as dimensionality increases the computation becomes intractable.

3 Sampling

Instead of using districtized mapping, another method is to sample points from the continuous space. So far, we've been intersted in estimating $p(x_t|...)$. Usually we're interested in the distribution to calculate some meaningful values like the mean state (7), the state variance (8), or some marginal probability of a state variable (9).

$$E_p[x] = \sum_x p(x)x \tag{7}$$

$$Var(x) = E_p[(x - E_p[x])^2] \tag{8}$$

$$E_p[1_{X_i=x_i}] = p(X_i = x_i) \tag{9}$$

One method to approximate these values is using a sampling approach:

$$x_{i=1...R} \sim p(x) \tag{10}$$

To approximate the expectation of some function $f(x)$, $E_{p(x)}[f(x)]$, we calculate:

$$\hat{\mu}_f = \frac{1}{R}\sum_{i=1}^{R} f(x^i) \tag{11}$$

By the Strong Law of Large Numbers, as R increases, $\hat{\mu}_f$ will approach $E_{p(x)}[f(x)]$

$$Var(\hat{\mu}_f) = \frac{Var_p[f]}{R} \tag{12}$$

More samples yeild a better, more accurate estimate, but are limited by computational feasibility and time.

3.1 The Derivation of the Normalized Importance Filter

Given some starting state, x_0, we're interested in sampling the following distribution.

$$p(x_1|z_1) = \frac{p(z_1|x_1)p(x_1|x_0)}{p(z_1)} \tag{13}$$

But there is no easy way to sample from the product of our motion model distribution, $p(x_1|x_0)$, and our measurement model distribution, $p(z_1|x_1)$ without massive computation. To achieve this, we will instead use Importance Sampling, which involves sampling from the wrong distribution and then correcting it:

$$\begin{aligned}
E_p[f(x)] &= \sum_x p(x)f(x) \\
&= \sum_x p(x)f(x)\frac{q(x)}{q(x)} \qquad \forall q(x) \neq 0 \\
&= \sum_x q(x)\frac{p(x)}{q(x)}f(x) \\
&= E_q\left[\frac{p(x)}{q(x)}f(x)\right]
\end{aligned}$$

So instead, let's draw our samples from the distribution Q and correct later to p:

$$\begin{aligned}
x_{i=1\ldots R} &\sim Q(x) \\
\hat{\mu}_f^p &= \frac{1}{R}\sum_{i=1}^{R}\left[\frac{p(x^i)}{q(x^i)}\right]f(x^i)
\end{aligned}$$

The quantity in brackets, $\left[\frac{p(x^i)}{q(x^i)}\right]$, is known as the importance weight, w^i and corrects the samples from the Q distribution to the p distribution. This lets us rewrite the equation as:

$$\begin{aligned}
\hat{\mu}_f^p &= \frac{1}{R} * \sum_{i=1}^{R} w^i * f(x^i) \\
w^i &= \left[\frac{p(x^i)}{q(x^i)}\right]
\end{aligned}$$

Again, we want to sample from the distribution in (13). Instead, lets use our motion model, $p(x_{t+1}|x_t)$ as Q and do important sampling. Forgetting about the normalizer $p(z_{t+1})$ for the moment,

$$\begin{aligned}
\hat{\mu}_x^p &= \frac{1}{R}\sum_{i=1}^{R}\left[\frac{p(x^i)}{q(x^i)}\right]x^i \\
\hat{\mu}_x^p &= \frac{1}{R}\sum_{i=1}^{R}\left[\frac{p(z_{t+1}|x_{t+1})p(x_{t+1}|x_t)}{p(x_{t+1}|x_t)}\right]x^i \\
\hat{\mu}_x^p &= \frac{1}{R}\sum_{i=1}^{R}p(z_{t+1}|x_{t+1})x^i
\end{aligned}$$

By sampling from the motion model as Q, which is easily computed, we can correct using the measurement model. The last problem to solve is normalizing the result. Using the same samples from Q, the motion model:

$$p(z_{t+1}) = \frac{1}{R}\sum_{i=1}^{R}p(z_{t+1}|x_{t+1}) \tag{14}$$

Combining the two,

$$\begin{aligned}
\hat{\mu}_x^p &= \frac{\frac{1}{R}\sum p(z_{t+1}|x_{t+1})x^i}{\frac{1}{R}\sum_{i=1}^{R}p(z_{t+1}|x_{t+1})} \\
\hat{\mu}_x^p &= \frac{\sum p(z_{t+1}|x_{t+1})x^i}{\sum_{i=1}^{R}p(z_{t+1}|x_{t+1})}
\end{aligned}$$

3.2 The Normalized Importance Sampling Algorithm

- Sample from the motion model $x_{i=1\ldots R} \sim p(x_{t+1}|x_t)$
- Compute the weights $w^i = p(z_{t+1}|x^i)$ from the measurement model
- Compute $\hat{w}^i = \frac{w^i}{\sum w^i}$
- Compute $\hat{\mu}_{f(x_{t+1})} = \sum \hat{w}^i f(x_i)$

Next time: adding multiple measurement iterations and the particle filter...