Statistical Techniques in Robotics (16-831, F10) Lecture#03 (Tuesday August 31)

Sample-based Filtering

Lecturer: Drew Bagnell *Scribe: Matt Aasted* [1]

1 Sampling

The belief of the robot is defined as:

$$Bel(x_t) := p(x_t|u_{1:t}, z_{1:t})$$

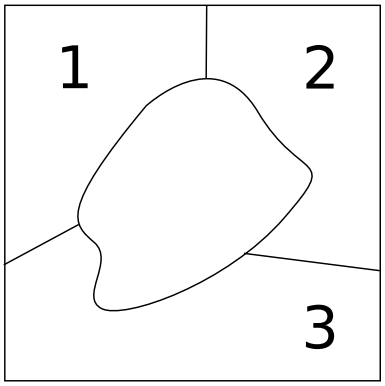
Our claim is that in general, this is difficult to compute in closed form. It is difficult or impossible to directly calculate the direct table of probabilities. But what are we going to use the belief of x_t for? We're not just vaguely interested – we want to do things with it. What we really want most of the time is the expectation, or various statistics such as the variance or other moments.

$$E[\vec{x}_t]$$
$$Var[\vec{x}_t] = E[(\vec{x} - E[\vec{x}])^2]$$

We can use sampling based methods to estimate these statistics without explicitly calculating the distributions involved.

1.1 An Example

We have a map with rooms numbered 1, 2, 3 as shown below.



We want probability that robot is in room $\{1, 2, 3\}$. But how would we write this an expectation?

[1] Some content adapted from previous scribes: Elliot Cuzzillo

We define rooms as subsets of continuous space, and invent a function $f()$ which is 1 if $x_t \in room(1)$, 0 otherwise. This is essentially the indicator function of the room: 1_{room}

Then the probability the robot is in the room is the also the expectation of the indicator function: $Pr(robot\ is\ in\ room(1)) = E[1_{room(x)}]$. In our case, this is the sum of the probabilities of being in a particular room. In general, the expectation of an indicator function is the probability of the probability of the indicated expression.

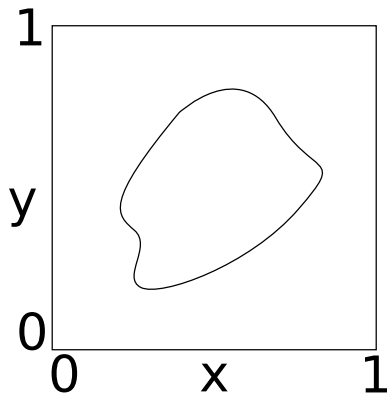
In many cases, we don't know $p(x)$ in closed form, but we can generate samples (X) with $pr(X = x)$. Supposing this is true, we can do

$$\frac{1}{n}\sum x_i \rightarrow_{n\rightarrow\infty} E[x]$$

or more strongly

$$\frac{1}{n}\sum f(x_i) \rightarrow_{n\rightarrow\infty} E[f(x)]$$

Now, suppose that we have another room:



The room is the amoeba in the center. Let X be our uncertain quantity (Random Variable) and $y = 1_{room(X)}$ be the indicator function for the room. $X \sim$ uniform on square - it is a vector where both elements are uniformly distributed on the interval $[0, 1]$ We have oracle access to whether X room or not, but we don't have direct access to $P(y)$.

To estimate it, we can generate random samples for X and calculate y. We then have samples of y, and we can do:

$$E[y] \approx \frac{1}{N}\sum_{1_{room}}(x)$$

where X_is are generated inependently for uniform ([0-1],[0-1])

1.2 An aside on π

Given that you can sample using this method, how can you compute the area of the room? The estimated area of the room is the total area times $E[y]$

The classical example is a Monte Carlo estimate of π. To do this, we make the room a circle, estimate random X, and count the samples of X inside the circle.

This was originally used for the Manhattan project and in nuclear physics. It has been reinvented several times.

1.3 Return to Robots

Suppose we are in some state x_t and we execute the command u_t, producing a banana shaped distribution for x_t. What we'd like to calculate is $p(x_{t+t}|x_t, u_t)$, but more reasonably we'd like to compute

$$E_{p(x_{t+1}|x_t,u_t)}[f(x_{t+1})]$$

where f() is arbitrary (example, the mean, or the variance)

However, we're missing a key component: measurement probability.

What we really want to do is something that looks like:

$$p(x_{t+1}|z_{t+1}, x_t) \propto p(z_{t+1}|x_{t+1})p(x_{t+1}|x_t)$$

It is not obvious how to sample from the product of these – it is clear how to sample each individually, though less often for measurements given state than next state given state.

2 Importance sampling (IS)

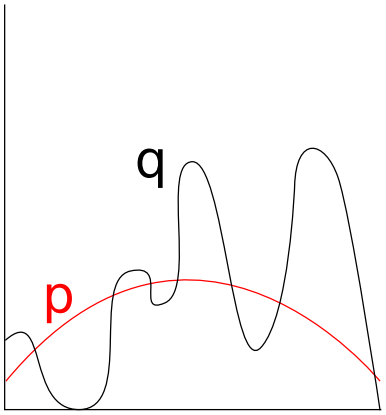
Note: from here on we drop the dependence on controls for the sake of notational convenience.

The trick: Sample from the incorrect distribution, reweight samples to fix it. We cannot sample from p, but it's what we want. We can sample from q.

$$\begin{aligned} E_p(x)[f(x)] &= \sum p(x)f(x)\frac{q(x)}{q(x)} \\ &= \sum q(x)\frac{p(x)}{q(x)}f(x) \\ &= E_{q(x)}[\frac{p(x)}{q(x)}f(x)] \end{aligned}$$

$\frac{p(x)}{q(x)}$ is called the importance weight. When we are undersampling an area, it weights it stronger; likewise, oversampled areas are weighted more weakly.

Consider the following (somewhat contrived) pair of functions:



Notice that there are places where p is nonzero and q is zero, which means we have regions to sample that we aren't sampling at all.

Our estimate of the expectation of p becomes:

$$\frac{1}{N}\sum_{i=1}^{N}\frac{p(x_i)}{q(x_i)}f(x_i) \rightarrow_{N\infty} E_p[f(x)]$$

Note that the samples x_i are drawn from q.

For any of these to work, we have to be able to sample $p(x_t + 1|x_t)$ and evaluate $p(z_{t+1}|x_{t+1})$

2.1 Example

Set q as the motion model. Note that in reality we never use this – we use a better approximation of $p(x_t + 1|x_t)$ because the motion model takes too many samples to be reliable.

$$p = \frac{1}{Z}p(z_{t+1}|x_{t+1})p(x_{t+1}|x_t)$$

$$X^i \sim p(x_{t+1}|x_t)$$

$$w^i = \frac{p}{q} = p(z_{t+1}|x_{t+1})$$

$$\bar{X} = E[X] = \frac{1}{N}\sum_{i=1}^{N} w^i X^i$$

But notice that we dropped the normalizer from bayes rule! (if we don't know Z)

3 Normalized IS

Proposed fix:

$$\bar{X} = \sum \frac{w^i}{\sum w^i} X^i$$

$\gamma = \sum_i w^i$ is an estimate of how much we've sampled over how much we should've sampled which is an estimate of $p(z_t| \dots)$

If it's low, we haven't drawn enough samples or the probability of the observation is low.

forall i **do**
  Sample x_0^i from $p(x_0)$
end
forall $t \in [1, T]$**do**
  forall i **do**
    sample x_t^i *from* $p(x_t|x_{t-1}^i)$
    $w^i* = p(z_t|x_t^i)$
  end
  forall i **do**
    Normalize weights
    $w_i = \frac{w_i}{\sum w_i}$
  end
end

At any point in the algorithm, we can extract the expectation of the distribution:

$$E[f(x)] = \sum w_i f(x^i)$$

4 SIR Filter (Particle filter, Condensation)

There is a bug here: The true path will be a sample with vanishing probability, so we have to throw out low probability samples and resample in high probability regions. This is also known as survival of the fittest.

Essentially, we insert a step where we line up all the samples with width determined by their weights normalized by the sum of the weights. We then draw from the uniform distribution from [0,1] and use those to select from the lined up samples. Since we sample by the weights, the new samples are weighted as either 1 or $\frac{1}{N}$ where N is the number of samples. See Figure 1 for an illustration of the sampling (though this is the low variance version from the next lecture – we are just selecting from the yellow strip at random).

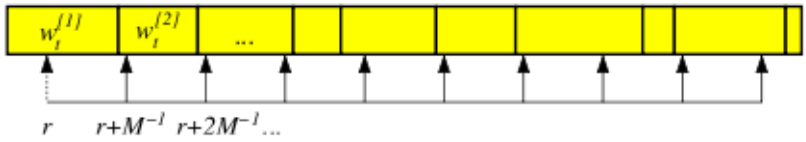


Figure 4.7 Principle of the low variance resampling procedure. We choose a random number r and then select those particles that correspond to $u = r + (m-1) \cdot M^{-1}$ where $m = 1, \ldots, M$.