

Sampling Based Filters

*Lecturer: Drew Bagnell**Scribe:Nate Wood*¹

1 Monte Carlo Method

As discussed in previous lectures, the belief of a robot, for example, is defined as:

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

If we wish to use the belief of x_t in any useful manner we are going to use the expected value. The expected value of any function f is given by:

$$E[f] = \sum_x f(x)p(x), \quad (2)$$

where $p(x)$ is the distribution of x . Rarely, if ever, is the distribution known in closed form. However, if you can generate samples of x_i from the distribution $p(x)$ the expectation can be estimated as:

$$E[f] \approx \frac{1}{N} \sum_{i=1}^N f(x_i). \quad (3)$$

This sampling based approach falls into a general class of algorithms called Monte Carlo Methods. This method of using random sampling to compute results have been reinvented/rediscovered many times, with the name being given by physicists at Los Alamos Scientific Laboratory in reference to the Monte Carlo Casino in Monaco.

1.1 Example 1: Area Estimation

Suppose we have a shape we wish to calculate the area of, as depicted by the amoeba shape in Fig. 1.1. Conceptually we can find the desired area by randomly sampling the area and taking the proportion of samples that fell inside the area to the total number of samples and multiplying by the area of the box.

More formally:

- Let $X = [x \ y]^T$ be our uncertain quantity (Random Variable).
- Assume we have an indicator function $Y = g(X)$ which takes the value 1 if it lies in the area and 0 otherwise.

¹Some content adapted from previous scribes: Matt Aasted, Elliot Cuzzillo

- We can sample uniformly from X
- The percentage of the total area the desired area covers is the expected value of the indicator function.

$$E[Y] \approx \frac{1}{N} \sum_{i=1}^N g(X_i) \quad (4)$$

- The area of the shape then

$$A_{amoeba} \approx E[Y]A_{total} \quad (5)$$

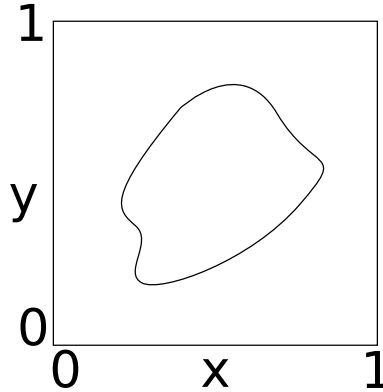


Figure 1: Area we wish to calculate

1.2 An aside on π

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.

1.3 Example 2: Which room am I in?

We have a map with rooms numbered 1, 2, 3 as shown in Fig. 1.3.

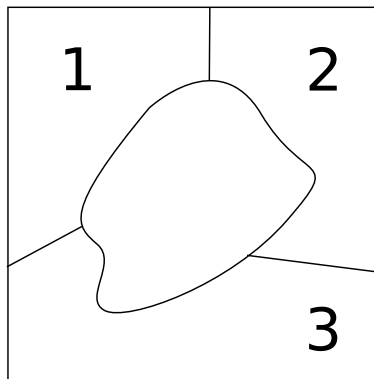


Figure 2: Rooms in which robot may be located

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

We define rooms as subsets of continuous space, and invent a function $f()$ which is 1 if $x_t \in \text{room}(1)$, 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: $Pr(\text{robot is in room}(1)) = E[1_{\text{room}(x)}]$. 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

$$\frac{1}{n} \sum x_i \rightarrow_{n \rightarrow \infty} E[x] \quad (6)$$

or more strongly

$$\frac{1}{n} \sum f(x_i) \rightarrow_{n \rightarrow \infty} E[f(x)] \quad (7)$$

1.4 Return to Robots

In the case of robots we would like to compute the expected value of a function f knowing both actions, u , and measurements, z .

$$E_{p(x_t|z_t, x_{t-1}, u_{t-1})} [f(x_t)] \quad (8)$$

Knowing that we will never know a closed form solution for this distribution leaves the previously described sample based method. In order to use the Monte Carlo Method would require samples drawn from the distribution

$$p(x_t|z_t, x_{t-1}, u_{t-1}), \quad (9)$$

which can be rewritten as

$$\eta p(z_t|x_t)p(x_t|x_{t-1}, u_{t-1}). \quad (10)$$

The distribution is the product of the measurement model, $p(z_t|x_t)$, and the robot motion model, $p(x_t|x_{t-1}, u_{t-1})$. It is not obvious how to sample from the product of these distributions. It is, however, straightforward to sample from the robot motion model, a classic example of which is shown in Fig. 1.4.

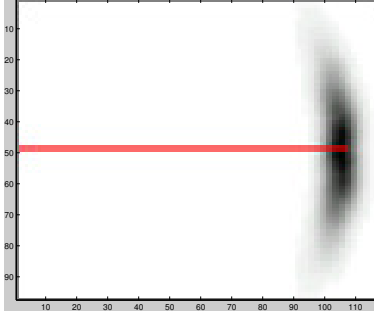


Figure 3: The motion model: Posterior distributions of the robot’s pose upon executing the motion command illustrated by the solid line. The darker a location, the more likely it is. This plot has been projected into 2-D. The original density is three-dimensional, taking the robot’s heading direction θ into account.

2 Importance sampling (IS)

Although we are unable to sample from the required distribution, $p(x)$, we can use a trick that will allow us to sample from the known distribution, $q(x)$. The trick: Sample from the available distribution and reweight samples to fix it. Manipulating the expectation over $p(x)$ yields the following approximation:

$$\mathbb{E}_{p(x)}[f(x)] = \sum p(x)f(x) \tag{11a}$$

$$= \sum p(x)f(x)\frac{q(x)}{q(x)} \tag{11b}$$

$$= \sum q(x)\frac{p(x)}{q(x)}f(x) \tag{11c}$$

$$= \mathbb{E}_{q(x)}\left[\frac{p(x)}{q(x)}f(x)\right] \tag{11d}$$

$$\approx \frac{1}{N} \sum_{i=1}^N \frac{p(x_i)}{q(x_i)}f(x_i). \tag{11e}$$

$\frac{p(x)}{q(x)}$ 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:

$$\frac{1}{N} \sum_{i=1}^N \frac{p(x_i)}{q(x_i)}f(x_i) \rightarrow_{N \rightarrow \infty} E_p[f(x)] \tag{12}$$

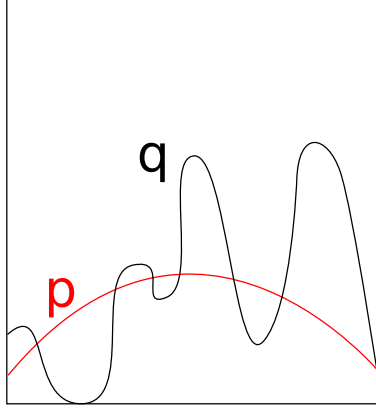


Figure 4: Comparison of distribution we wish to sample from, p , and distribution we can sample from, q .

2.1 Example

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

$$p = \eta p(z_t|x_t)p(x_t|x_{t-1}) \quad (13)$$

$$X^i \sim p(x_t|x_{t-1}) \quad (14)$$

$$w^i = \frac{p}{q} = p(z_t|x_t) \quad (15)$$

$$\bar{X} = \mathbb{E}[X] = \frac{1}{N} \sum_{i=1}^N w^i X^i \quad (16)$$

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

3 Normalized IS

We can normalize estimate using:

$$\frac{1}{N} \sum w^i \xrightarrow{N \rightarrow \infty} \sum q(x) = \sum p(x_t|x_{t-1})p(z_t|x_t) = Z \quad (17)$$

$$\tilde{w}^i = \frac{w^i}{\sum w^i} \quad (18)$$

$$\bar{X} = \sum \tilde{w}^i X^i \quad (19)$$

Algorithm 1 NIS Filter Algorithm

```
for all  $i$  do
  sample  $x_0^i$  from  $p(x_0)$ 
end for
for all  $t$  do
  for all  $i$  do
    sample  $x_t^i$  from  $p(x_t|x_{t-1}^i)$ 
     $w_i^* = p(z_t|x_t^i)$ 
  end for
   $\tilde{w}_i = \frac{w_i}{\sum w_i}$ 
end for
 $E[f(x)] = \sum_i \tilde{w}_i f(x_i)$ 
```

4 SIR Filter (Particle filter, Condensation)

There is a bug in the NIS filter. 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 a Particle filter.