Statistical Techniques in Robotics (16-831, F08) Lecture #4 (Sept 4, 2008)

# Particle Filters

*Lecturer: Drew Bagnell* *Scribe: Mark Desnoyer*

## Contents



## 1 Motivation

Last time we looked at discrete Bayes filters to probabilistically identify a state. However, this technique requires discretizing the entire state space and keeping track of each cell at all times. For many applications, like localizing a robot in a gridded room, this can be very computationally

expensive. So, instead of breaking up the state space, we will describe the current state as a sample of possible states.

## 2 Intuition

Describing the current state as a set of samples can be expressed as:

$$E_p[x] = \frac{1}{R}\sum_{i=1}^{R} f(x_i) \qquad \text{where} \qquad x_i \qquad \text{is sampled from} \qquad p(x_t|z_0, z_1, \ldots, z_t) \tag{1}$$

However, it is often not possible to sample from $p(x_t|z_0, z_1, \ldots, z_t)$ directly, so instead, we will sample from a different distribution and them make a correction. This process is called importance sampling.

## 3 Particle Filtering

### 3.1 Importance Sampling

Importance sampling is the process of sampling from a different distribution $q$, which is related to $p$ to compute a value we're interested in and then correcting the result. This works because

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

Often, the motion model is used for $q$:

$$q(x) = p(x_{t+1}|x_t) \tag{2}$$

and the observation model is used for $\frac{p}{q}$:

$$\frac{p_i}{q_i} = p(z|x_i) \tag{3}$$

$$= w_i \tag{4}$$

The weights also need to be normalized, which can be done by:

$$\hat{w_i} = \frac{w_i}{\sum w_i} \tag{5}$$

This leads to the following particle filtering algorithm:

**Algorithm 1** Basic Particle 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 $p(x_t|x_{t-1}^i)$
    $w_i \quad * = p(z_t|x_t^i)$
  **end for**
  $\hat{w}_i = \frac{w_i}{\sum w_i}$
**end for**
$E[f(x)] = \sum_i \hat{w}_i f(x_i)$

### 3.2 Resampling

However, because the observation model is only used to reweight the particles, as time increases, the particles will tend to diverge to different areas of the search space. In this case it is very easy to have one particle with a very high weight and many with very small weights, distributed over a very large search space. This is very inefficient.

So, to help solve this problem, we resample the data at every time step, thus pruning uninteresting solutions. This sampling is done so that the resulting particles are chosen from the previous set with a probability proportional to the previous weights (Figure 1.1):

$$p(x^i \leftarrow x^j) \quad \alpha \quad w^j \tag{6}$$

After resampling, all the weights are then reset to 1. The resulting modified algorithm is:

**Algorithm 2** Particle Filter Algorithm With Resampling

**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 $p(x_t|x_{t-1}^i)$
    $w_i \quad * = p(z_t|x_t^i)$
  **end for**
  $\hat{w}_i = \frac{w_i}{\sum w_i}$
  **for all** $i$ **do**
    draw $k$ with probability $\alpha \quad w_i$
    $y_t^i = x_t^k$
  **end for**
  **for all** $i$ **do**
    $x_t^i = y_t^i$
    $w_i = 1$
  **end for**
**end for**
$E[f(x)] = \sum_i \hat{w}_i f(x_i)$

### 3.3 Other Names

This technique has many names depending on the field. For example:

- Condensation - Computer Vision
- Particle Filtering - Robotics
- SIR Filter - Statistics
- Survival of the Fittest

## 4 Failure Modes

### 4.1 Bad Motion Model

If the motion model is not very good, it is possible to detect the same thing many times and not recognize that it is the same object. This leads to a large number of extra particles, which is inefficient.

### 4.2 Sensor Model is Too Good

If the sensor model is too good, particle filtering can fail. For example, say has a very sharp spike where the object is. Now, when the sensor model is sampled by the particles, a few things can happen, which are undesirable.

Only one particle might be in the vicinity of the spike. This will cause its weight to dwarf the other particles and so when resampling occurs, only one particle will exist. This means that when the situation changes, there might not be enough diversity of particles to be able to handle the disturbance.

Even worse, if all of the particles miss the spike in the observation model, then they will be reweighted based on the low-level differences. Thus, the weights are effectively noise and the particle filtering algorithm might not be able to converge.

### 4.3 Resampling Too Aggressively

If resampling occurs too often, it is possible to throw out a mode of solutions inadvertently.

## 5 Fixes to Failure Modes

### 5.1 Overestimate Noise

Overestimating the noise of the sensor model smooths it out, avoiding the possibility of having a model that is too good. However, this will add extra computation to the solution. Plus, it's a hack.

### 5.2 Use More Particles

Using more particles will often help with any of the above problems, however, the computational complexity is directly related to the number of particles used, so this approach can be expensive. Also, particle filters tend to be robust to bugs if you just use more particles, so it is possible that you have a bug if you need to increase your particles significantly to get a solution.

### 5.3 Improve the Motion Model

Improving the motion model will help if the root cause is a bad model.

### 5.4 Rao-Blackwellization

We'll discuss this in depth in a later lecture, but the basic idea is to make each particle smarter and thus carry more information. For example, each particle could contain a Kalman filter.

### 5.5 Change Q

When doing the importance sampling, we usually use the motion model. However, $Q$ only needs to be a different distribution, which we can later adjust. For example, we could sample from the observation model and weight by the motion model.

## 5.6 Low Variance Resampling

When resampling, instead of pulling from the distribution independent samples, we can choose a random number $r$ and the choose particles $u = r + \frac{m-1}{M}$. See Figure 4.7 from Probabilistic Robotics.

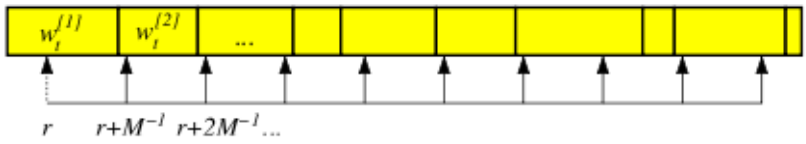


**Figure 4.7** Principle of the low variance resampling procedure. We choose a random number $r$ and then select those particles that correspond to $u = r + (m-1) \cdot M^{-1}$ where $m = 1, \ldots, M$.

From Probabilistic Robotics pg. 111

This technique tends to result in samples that have lower noise and is guaranteed not to drop particles as long as their probability is greater than $\frac{1}{M}$.

## 5.7 Only Resample When Necessary

If resampling is being done too aggresively, it is a good idea to resample less frequency, hopefully based on the current state of the particle filters. For example, one could resample only when the variance of the weights go over a threshold.