Statistical Techniques in Robotics (16-831, F10) Lecture#04 (Thursday September 2)

Particle Filters: The Good, The Bad, The Ugly, And Some Fixes

Lecturer: Drew Bagnell *Scribe:Joydeep Biswas*

1 Particle Filters: The Good

1. Particle filters can answer most queries with samples of $Bel(x)$:
   This is good because we do not need (and indeed, in most cases would not have) a complete solution of $Bel(x)$
2. (Normalized) Importance Sampling [lecture #3]:
   Even if we cannot sample from
   $$p(x_{t+1}|x_t, z_{t+1}, u_t) \quad \propto \quad p(z_{t+1}|x_{t+1}) \cdot p(x_{t+1}|x_t, u_t) \tag{1}$$
   $$p(x_{t+1}|x_t, z_{t+1}, u_t) \quad = \quad \frac{obs.motion}{\mathbb{Z}} \tag{2}$$
   we can use some distribution q instead of p to sample from, and factor out importance weights [lecture #3]. However, unless q closely matches p, the weights would get increasingly smaller.
3. Works with a finite number of particles:
   Particle filters (should) use resampling to do "survival of the fittest", thus sampling densely around where the best estimates lie.

2 Particle Filters: The Bad

2.1 Loss of diversity

Repeated resampling (without low variance resampling) in the absence of any actual sensory observations could lead to loss of diversity. This problem is illustrated by the following example of a robot localizing itself in two dimensions on a map with two (identical) rooms.

Fig. 1 shows the state of a particle filter at a particular instant of time (say, $t = 0$). Suppose that the two clusters of particles have equal weights, and each particle has weight $1/N$, where $N =$ number of particles. After resampling in the next iteration, (if the particle filter does not use low variance resampling), it is unlikely that both the resampled clusters will still have equal weights. In the absence of any new observations, the particle filter would end up (after a few more time steps) in a state like that in Fig. 2, where one cluster has dwindled at the expense of the other, for no good reason.

2.2 Really good observation models result in bad particle filters

If the observation model is really good (both accurate and confident), then the updates of the belief of the particle filter are likely to result in confident (closely clustered), yet incorrect estimates. This

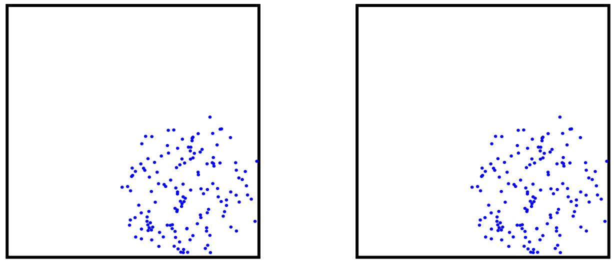

Figure 1: Loss of diversity example: State of particles at time step $t = 0$, with two equally weighted clusters and all particles with equal weights.

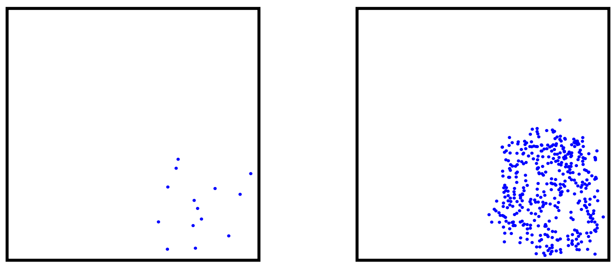

Figure 2: Loss of diversity example: State of particles at a later time step. The particles have essentially coalesced to a simgle hypotheses, located in room 2.

problem is best illustrated by the following example of a robot localizing itself in a single dimension x. Fig. 3 shows the observation likelihood $P(z|x)$ at an instant of time. Notably, the observation model has a very narrow but strong peak at location x_1, where the sensor believes that the robot is located. Also, there is a smaller peak due to some noisy readings (say) at location x_2. The observation likelihood is uniformly infinitesimal everywhere else. Given this observation likelihood, two problematic issues are likely to crop up:

1. Given the narrow width of the peak around x_1, it is likely that no particle would fall within the region covered by the peak. Thus, the weights of most particles would be driven to zero, and the accurate location estimate (as provided by the observation model) would be lost.
2. It is likely that a particle (or more) might lie under the region covered by the hump around x_2, and its (or their) weight(s) would increase with respect to the other particles, which (as described by the last point) would have infinitesimal weights. This would result in the particle filter providing a confident, albeit completetely wrong estimate of the robot's location.

2.3 Non-deterministic behaviour

Given the random sampling necessary for a particle filter, it is unlikely that two particle filters, even with identical initial states and observations of time, would follow the exact same history of estimates.

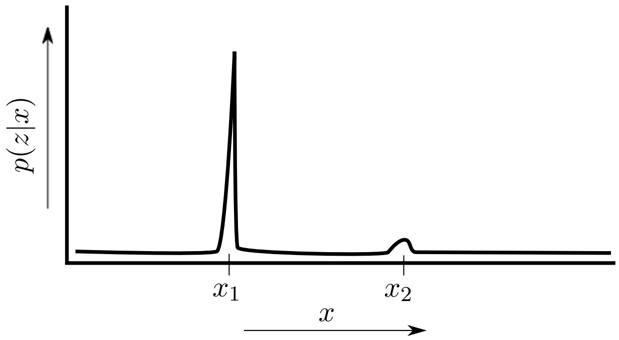

Figure 3: "Good" observation model: The observation likelihood has an accurate and confident (tall and narrow) peak around location x_1, and an erroneous low intensity peak around x_2.

2.4 Mismatch between p and q

The farther q is to p, the lesser it will be sampling the modes of p. Eventually, the particle filter could entirely loose track of the modes of p, and degenerate into overconfident, yet incorrect estimates.

3 Particle Filters: The Ugly (Misconceptions)

1. "*Particle filters sample from a motion model, weight by an observation model*"
   Particle filters do not *always* sample from a motion model and then weight by an observation model. They *could* sample from other distributions, if they were available, easy to sample from, and/or more accurate.

2. "*Particle filters are for localization*"
   Particle filters are not used *exclusively* for localization

4 Particle Filters: Fixes For The Bad

4.0 Use more particles

With infinite particles, even if p and q were dissimilar, one could at least ensure that q was such that all regions of p were being sampled. Also, resampling would not be as crucial.

4.1 Resample less

This would partially fix the issue raised in Section 2.1. One approach is to resample only when the variance in the weights of the particles is high.

4.2 Better resampling

This would again help with the issue raised in Section 2.1. Before resampling, (conceptually) a list of the particles is created (Fig. 4), with the length of the list equal to the sum of the weight of the particles, (say) W, such that each particle (conceptually) occupies a length in the list equal to its weight.
The naïve approach to resampling is to pick N random samples from the list, as depicted in Fig. 5. Intuitively, particles with weight greater than W/N should have been sampled at least once, particles with weight $2W/N$ at least twice, and so on. However, this is not ensured by the naïve algorithm.
An alternative algorithm is to find all particles with weights greater than W/N, draw one sample from them, decrease the length of their "segments" in the list by W/N, and draw the remaining samples from this new (shortened) list. An even better algorithm, called "low variance sampling" (similar in principle to "stochastic universal sampling"[1]) draws N samples using a single randomly generated number. Let the random number generated be r,such that $r \in (0, W)$. It is assumed that r is drown from a uniform distribution. Then, the first sample is drawn from location r in the list, the second from location $\left[\left(r + \frac{W}{N}\right) \bmod W\right]$, the i'th sample from location $\left[\left(r + \frac{i \cdot W}{N}\right) \bmod W\right]$ and so on, until N new samples have been generated, as illustrated by Fig. 6.

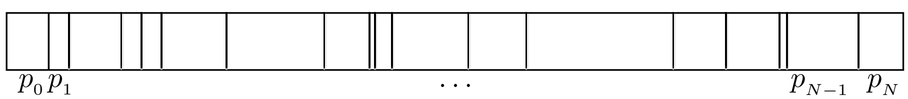

Figure 4: Resampling list

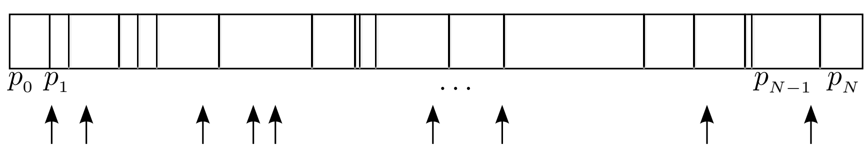

Figure 5: Naïve resampling: N samples are drawn from random locations, as indicated by arrows

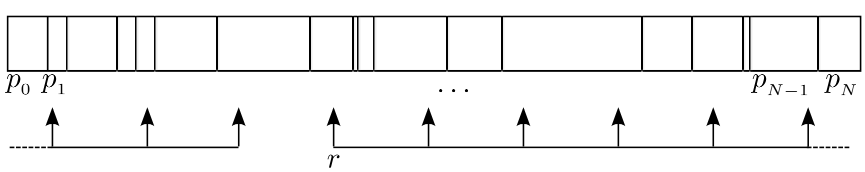

Figure 6: Low variance resampling: First sample drawn from random location r, rest picked at successive intevals of $\left[\frac{W}{N}\right]$

4.3 Make observation model less certain

Section 2.2 illustrated the ill-effects of a very good observation model. Two hacks to alleviate this problem are:

- Convolve the observation model with a Gaussian, thus smearing it out

[1] http://en.wikipedia.org/w/index.php?title=Stochastic_universal_sampling&oldid=356360569

- Raise the likelihood values returned by the observation model to a power c where $c \in (0, 1)$. A commonly chosen value of c is $1/M$, where M is the dimensionality of the observation, thus taking the geometric mean of the observation values.

4.4 Use a better sampling distribution q

The closer the distribution q to p, the more accurate the update of the belief is likely to be. One hack is to use a coarse grid estimate of p to sample from.

4.5 Rao-Blackwellization

Rao-Blackwellized particle filters (to be discussed later in the course) include most of the good properties and the "fixes" mentioned here, including sampling less, and being more exact.