Particle Filter notes

Mac Radigan

Particle Filter

The Kalman filter is limited by the need to be able to characterize the target under surveillance using a linearized model. Particle filters have more general application to targets with non-linear motion behavior, as the estimation can be produced adaptively by maintaining an internal set of probablility distribution weights[**?**].

The general idea is use an estimate x of the prior measurement z probability density function (PDF) to provide an estimate for updating the current measurement, shown in Equation ~(1) [**?**].

$$p\left(x_n|z_{1:n-1}\right) = \int p\left(x_n|x_{n-1}\right) p\left(x_{k-1}|z_{1:n-1}\right) \mathrm{d}x_{n-1} \tag{1}$$

The state space equation itself may be used to approximate the posterior PDF by sampling the state-space equation. Arulampalam shows the approximation in as weighting terms defined as conditional probability of a state given prior measurements $w_{n-1|n-1}^i \equiv Pr(\mathbf{x}_{n-1} = \mathbf{x_{n-1|z_{1:n-1}}^i})$. The prior PDF approximated from the state space equation is shown in Equation ~(2) [**?**].

$$p(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1}) \approx \sum_{i=1}^{\#Particles} \mathbf{w}_{n-1|n-1}^i \delta(\mathbf{x}_{n-1} - \mathbf{x}_{n-1}^i) \tag{2}$$

This approximation may then be used for prediction (Equation ~(3)) and update (Equation ~(4)) of the current measurements based on the prior PDF [**?**].

$$p(\mathbf{x}_n|\mathbf{z}_{1:n-1}) \approx \sum_{i=1}^{\#Particles} \mathbf{w}_{n|n-1}^i \delta(\mathbf{x}_n - \mathbf{x}_n^i) \tag{3}$$

$$p(\mathbf{x}_n|\mathbf{z}_{1:n}) \approx \sum_{i=1}^{\#Particles} \mathbf{w}_{n|n}^i \delta(\mathbf{x}_n - \mathbf{x}_n^i) \tag{4}$$

Operation

The particle filter algorithm given in (Equation ~(1)) may be used as a measurement correction [**?**] based on the grid PDF approximations given in Equations (2)-(4). The algorithm begins with assigning uniform particle weights. During each iteration, the weights of the prior PDF if updated based on the current measurement, conditioned on the sampled state space. These weights are then normalized, and used construct and estimate by integrating the prediction PDF over the current state. The state equation is then used to construct the prediction for the next iteration.

Algorithm 1 Particle Fitler Algorithm

---
for n=1:N **do**
  for i=1:#Particles **do**
    - Assign initial particles: $\mathbf{x}_0^{(n)}$
    - Assign initial normalized weights: $\mathbf{w}_0^{(n)} = \frac{1}{N}$
  end for
  for i=1:#Particles **do**
    - Assign weights: $\mathbf{w_n^{(i)}} = \frac{p(\mathbf{z_n^{(i)}}|\mathbf{x_n^{(i)}})}{p(\mathbf{z}_n^{(i)})}$
  end for
  for i=1:#Particles **do**
    - Normalize weights: $\mathbf{w}_n^{(i)} = \frac{\mathbf{w}_n^{(i)}}{\sum_{l=1}^{\#Particles} \mathbf{w_i^{(i)}}}$
  end for
  - Construct estimate: $\hat{\mathbf{x}}_n = \sum_{i=1}^{\#Particles} \mathbf{w}_n^{(i)} \mathbf{x}_n^{(i)}$
  for i=1:#Particles **do**
    - Advance prediction using state model: $\mathbf{x}_{n+1}^{(i)} = \mathbf{F}\mathbf{w}_n^{(i)} \mathbf{x}_n^{(i)}$
  end for
end for

Resampling

One of the drawbacks of this algorithm is known as "degeneracy", which is a tendency for the a single particle of the posterior PDF to accumulate the majority of the density, rendering all other particle weights negligible. Proper choice of importance densitiy and resampling are two techniques used to combat the degeneracy problem [**?**].

The degeneracy is commonly measured using a metric known as the effective sample size , given below in Equation ~(5). Small N_{eff} is an indicator of degeneracy [**?**].

$$\bar{N}_{eff} = \frac{1}{\sum_{i=1}^{\#Particles} \left(\mathbf{w}_n^i\right)^2} \tag{5}$$

A resampling method known as Sequence Importance Resampling (SIR) may be used to avoid the degeneration issue [**?**]. A typically employed resampling algorithm for the particle filter is given below in Algortihm 2 [**?**].

Algorithm 2 Particle Fitler Resampling Algorithm

- Build CDF:

$P_1 := 0$

for $i = 2 : \#Particles$ **do**

   $P_i := P_{i-1} + \mathbf{w_n^i}$

end for

- Create a uniform PDF: $u \sim \mathbf{U}\left(0, \frac{1}{\#Particles}\right)$

for $j = 1 : \#Particles$ **do**

   $u_j = u_{\frac{j}{\#Partilces}}$

   while $u_j > c_i$ **do**

      $i = i + 1$

   end while

   $\mathbf{x}_n^j = \mathbf{x}_n^i$

   $\mathbf{w}_n^j = \frac{1}{\#Particles}$

   $\mathbf{i}^j = i$

end for