Particle Filter Notes

Snowballc

Dec. 25, 2019

1 System Model

The form of this model is very very crucial!!!

2 Bayes Filter

In: Thrun et al. (2005).
Target density: posterior (EQ 4.30)

$$bel(x_{0:t}) = p(x_{0:t}|u_{1:t}, z_{1:t})$$

Proposal density: (EQ 4.32)

$$p(x_t|x_{t-1}, u_t)\ bel(x_{0:t-1})$$

Weights: (EQ 4.33)

$$w_t^{[m]} = \frac{Target\ dist}{Proposal\ dist} = \eta\ p(z_t|x_t)$$

Do the filtering:

- 0. Init state x_0 and posterior prob $Bel(x_0)$.
- 1. Assign $Bel(x_{1:t-1}) = Bel(x_{1:t})$, or $Bel(x_{1:t-1}) = Bel(x_0)$ if at the start.
- 2. Forecast $\overline{Bel}(x_{1:t})$ using system equation $f(x_t|x_{t-1})$. (EQ.3)
- 3. Corect the forcast $\overline{Bel}(x_{1:t})$, using new observation y_t, and obtain $Bel(x_{1:t})$.(EQ.2)
- 4. Go to step 1 and do the iteration.

Eq.4.31 in Thrun et al. (2005)

$$Bel(x_{1:t}) = \eta\ l(y_t|x_t)\ p(x_t|x_{t-1}, u_t)\ Bel(x_{1:t-1}) \tag{1}$$

$$w(x_t) = \eta\ l(y_t|x_t) \tag{2}$$

$$q(x_{1:t}) = \ p(x_t|x_{t-1}, u_t)\ Bel(x_{1:t-1}) \tag{3}$$

In: Doucet and Johansen (2009).——————————

$\pi_n(x_{1:n})$ is the distribution that we want to know, but we don't know it, or we cant observe it directly, just like the posterior pdf of the tx signal x: $p(x|y)$ in Communication. Therefore, we do some equally transformation in mathematics, and reprent $p(x|y)$ with likelihood $l(y|x)$ and prior pdf $p(x)$.

$$p(x|y) = \frac{p(x,y)}{p(y)} = \frac{p(y|x)p(x)}{p(y)}$$

In Bayess Filtering, we use the same trick: (Represent π with γ)

$$\pi_n(x_{1:n}) = \frac{\gamma_n(x_{1:n})}{Z_n} \tag{4}$$

Forecest: Eq.12

$$p(x_n|y_{1:n-1}) = \int f(x_n|x_{n-1}) p(x_{n-1}|y_{1:n-1}) dx_{n-1}$$

Update Eq.11

$$p(x_n|y_{1:n}) = \frac{l(y_n|x_n)p(x_n|y_{1:n-1})}{p(y_n|y_{1:n-1})}$$

Where denominator:

$$p(y_n|y_{1:n-1}) = \int l(y_n|x_n)p(x_n|y_{1:n-1})dx_n$$

3 Particle Filter

Using mass particles to simulate the real distribution.

We generate a huge amount of particles in state x space, and use the statistics of x to represent distribuiton $p(x)$, or posterior $p(x|y)$, more precisely speaking.

In particle filter, we need to use particles(samples) to simulate the population, but how should we generate particles, uniformly or normally? to accurately and efficiently represent the population. The solution to this question is **Importance Sampling**.

Particle Filtering steps:

- 1. Sample: generating particles from the population.
- 2. Weight: if this particle is the right estimation of the state x, then compare it with the observation y, calc its weight.
- 3. Multiply: generate next generation using the system equation. Particles with higher weights generate more offsprings. Particles in the next generation all have equal weights now.
- 4. Go to step 2, recalc the weights, and loop.

3.1 Importance Sampling

We introduce **importance density** $q_n(x_{1:n})$. We generate particles using distribution $q_n(x_{1:n})$, then assign weights to such parcitles, to simulate the dist π. (Eq.21)

$$\pi(x_{1:n}) = \frac{w_n(x_{1:n})q_n(x_{1:n})}{Z_n} \tag{5}$$

Where Z_n is the factor doing normalization .

3.2 How to calc weights?

Theoretically, it shoud be:

$$w_n(x_{1:n}) \propto \frac{\pi(x_{1:n})}{q_n(x_{1:n})}$$

However, as mentioned before, we cant observe π directly. According to (4), we have:

$$w_n(x_{1:n}) \propto \frac{\pi_n(x_{1:n})}{q_n(x_{1:n})} \propto \frac{\gamma_n(x_{1:n})}{q_n(x_{1:n})}$$

This is so abstract to understand without concrete examples. However, for better understanding, you can refenence (EQ.2), where $Bel(x_{1:t})$ there is the γ here.

3.3 How to calc weights recursively?

We should select an importance distribution $q_n(x_{1:n})$ which has the following structure: (EQ.28)

$$q_n(x_{1:n}) = q_{n-1}(x_{1:n-1})q_n(x_n|x_{1:n-1}) \tag{6}$$

Then: (EQ.29)

$$\begin{aligned}
w_n(x_{1:n}) &= \frac{\gamma_n(x_{1:n})}{q_n(x_{1:n})} \\
&= \frac{\gamma_n(x_{1:n})}{q_n(x_{1:n})}\frac{\gamma_{n-1}(x_{1:n-1})}{\gamma_{n-1}(x_{1:n-1})} \\
&= \frac{\gamma_{n-1}(x_{1:n-1})}{q_{n-1}(x_{1:n-1})q_n(x_n|x_{1:n-1})}\frac{\gamma_n(x_{1:n})}{\gamma_{n-1}(x_{1:n-1})} \\
&= \frac{\gamma_{n-1}(x_{1:n-1})}{q_{n-1}(x_{1:n-1})}\frac{\gamma_n(x_{1:n})}{\gamma_{n-1}(x_{1:n-1})q_n(x_n|x_{1:n-1})} \\
&= w_{n-1}(x_{1:n-1})\frac{\gamma_n(x_{1:n})}{\gamma_{n-1}(x_{1:n-1})q_n(x_n|x_{1:n-1})}
\end{aligned} \tag{7}$$

Define $\alpha_n(x_{1:n})$ as **incremental importance weight**: (EQ.30)

$$\alpha_n(x_{1:n}) = \frac{\gamma_n(x_{1:n})}{\gamma_{n-1}(x_{1:n-1})q_n(x_n|x_{1:n-1})} \tag{8}$$

Then we get:

$$w_n(x_{1:n}) = w_{n-1}(x_{1:n-1})\alpha_n(x_{1:n}) \tag{9}$$

3.4 How to Multiply and Resample?

At the initial step, particles could be genreated uniformly in the prob space.
At the Multiply step.
Uniform or Gaussian???!!!

4 Doing Prediction

As noted in Kalman Filter notes, we can doing prediction of not only $t+1$, also $t+2$, $t+3$...

In Kalman Filter, we predict $x_{t+1|t}$ based on $x_{t|t}$, then, we think the state at time $t+1$ will have a Gaussian distribution with mean $x_{t+1|t}$ and variance σ^2.

However, in Particle Filter, after prediction, we have many particles of $x_{t+1|t}$. Those particles indeed give us the pdf of this prediction. The distribution is not limited to Gaussian. Indeed this posterior distribution could be any shape, without analytical form.

5 Example 1: logRet

Modeling log price or log return as

$$logR(t) = u(t) + W(t) \tag{10}$$

Where u(t) is a smoothy and slowly changing (not randomly) process, and W(t) is the obsearvation noise.

We assume that it is enought to use a second order polynomial to model u(t) when time interval Δt is small, meaning that:

$$u(t+\Delta t) = u(t) + u^{'}\Delta t + \frac{1}{2}u^{''}\Delta t^2 + O(\Delta t^2)) \tag{11}$$

Maybe this is not right to be used in Particle Filtering.

Instead, in PF, assume the process $u(t)$ is a 1-order Markov process, meaning that $u(t)$ only depends on $u(t-1)$ and get no information from $u(t-i), when\ i >= 2$.

First order Markov

$$f(x_n|x_{1:n-1}) = f(x_n|x_{n-1})$$

Starting point: Suppose we've got: $logR(0)$, $logR(1)$ and $logR(2)$, then we can get: $\sim$

6 Example 2

Referenced from Doucet and Johansen (2009).

In its section 4.1 : Assuming: $\gamma_n(x_{1:n}) = p(x_{1:n}, y_{1:n})$ and $Z_n = p(y_{1:n})$. Then we have $\pi_n(x_{1:n})$ as posterior probability:

$$
\begin{aligned}
\pi_n(x_{1:n}) &= \frac{\gamma_n(x_{1:n})}{Z_n} \\
&= \frac{p(x_{1:n}, y_{1:n})}{p(y_{1:n})} \\
&= p(x_{1:n}|y_{1:n})
\end{aligned}
\tag{12}
$$

$$
\begin{aligned}
\gamma_n(x_{1:n}) &= p(x_{1:n}, y_{1:n}) \\
&= l(y_{1:n}|x_{1:n})p(x_{1:n}) \\
&= l(y_{1:n-1}|x_{1:n-1})p(x_{1:n-1})\ l(y_n|x_n)p(x_n) \\
&= \gamma_{n-1}(x_{1:n-1})\ l(y_n|x_n)p(x_n)
\end{aligned}
\tag{13}
$$

$p(x_n)$ is calculated recursively. In the computation, $p(x_n) = p(x_n|y_n)$. It is the posterior prob at time t, then, it serves as the prior prob to calc $p(x_{t+1}|y_{t_1})$bat time $t+1$.

Next, we assign *importance density* $q_n(x_{1:n})$ as:

$$q_n(x_{1:n}) \sim N(0, 1)$$

Then, we compute weights w using (9)

$$w_n(x_{1:n}) = w_{n-1}(x_{1:n-1})\frac{\gamma_n(x_{1:n})}{\gamma_{n-1}(x_{1:n-1})q_n(x_n|x_{1:n-1})}$$

Near Eq. 39:

$$\alpha_n(x_{1:n}) = p(y_n|x_{n-1})$$

FORCAST:

$$p(x_n|x_{1:n-1}) = p(x_n|x_{n-1})$$

- Find an apropriote distribution: q.
- sample $X_1^i \sim q(x_1|y_1)$.
- compute the weights $w_1(X_1^i) =$
- resample $\{W_1^i, X_1^i\}$ to obtain N equally-weighted particles $\{\frac{1}{N}, \overline{X_1^i}\}$

References

Doucet, A. and Johansen, A. M. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic robotics*. MIT press.