Quantifying Uncertainty

Sai Ravela

M. I. T

Last Updated: Spring 2013
Particle Filters

- Applied to Sequential filtering problems
- Can also be applied to smoothing problems
- Solution via Recursive Bayesian Estimation
- Approximate Solution
- Can work with non-Gaussian distributions/non-linear dynamics
- Applicable to many other problems e.g. Spatial Inference
Notation

\( x_t, X_k \): Models states in continuous and discrete space-time respectively.
\( x_t^T \): True system state
\( y_t, Y_k \): Continuous and discrete measurements, respectively.
\( X_k^n \): \( n^{th} \) sample of discrete vector at step \( k \).
\( M \): model, \( P \): probability mass function.
\( Q \): Proposal Distribution, \( \delta \): kronecker or dirac delta function.

We follow Arulampalam et al.'s paper.

Non-Gaussianity

- Sampling
  - SIS
  - SIR
  - Kernel
  - RPF
Sequential Filtering

Recall: Ensemble Kalman filter & Smoother

We are interested in studying the evolution of $y_t \in f(x_t^T)$, observed system, using a model with state $x_t$. 
This means (in discrete time, discretized space):

\[ P(X_k | Y_{1:k}) \]

Can be solved recursively

\[ P(X_k | Y_{1:k}) = \frac{P(X_k, Y_{1:k})}{P(Y_{1:k})} \]
$Y_{1:k}$ is a collection of variables $Y_1 \ldots Y_k$

So:

$$P(X_k | Y_{1:k}) = \frac{P(X_k, Y_{1:k})}{P(Y_{1:k})}$$

$$= \frac{P(Y_k | X_k) P(X_k | Y_k) P(Y_{1:k-1})}{P(Y_k | Y_{1:k-1}) P(Y_{1:k-1})}$$

$$= \frac{P(Y_k | X_k) P(X_k | Y_{1:k-1})}{P(Y_k | Y_{1:k-1})}$$
Contd.

\[ P(X_k \mid Y_{1:k}) = \frac{P(Y_k \mid X_k) \sum_{X_{k-1}} P(X_k \mid X_{k-1}) P(X_{k-1} \mid Y_{1:k-1})}{\sum_{X_k} \sum_{X_{k-1}} P(Y_k \mid X_k) P(X_k \mid X_{k-1}) P(X_{k-1} \mid Y_{k-1})} \]

1. From the Chapman-Kolmogorov equation
2. The measurement model/observation equation
3. Normalization Constant

When can this recursive master equation be solved?
Let’s say

\[ X_k = F_k X_{k-1} + V_k \]
\[ Z_k = H_k X_k + \eta_k \]
\[ v_k = N(\cdot, P_{k|k}) \]
\[ \eta_k = N(0, R) \]

Linear Gaussian \(\rightarrow\) Kalman Filter
For non linear problems

Extended Kalman Filter, via linearization
Ensemble Kalman filter
  ▶ No linearization
  ▶ Gaussian assumption
  ▶ Ensemble members are “particles” that moved around in state space
  ▶ They represent the moments of uncertainty
How may we relax the Gaussian assumption?

If $P(X_k|X_{k-1})$ and $P(Y_k|X_k)$ are non-gaussian;

How do we represent them, let alone perform these integrations in (2) & (3)?
Particle Representation

Generically

\[ P(X) = \sum_{i=1}^{N} w^i \delta(X - X^i) \]

pmf/pdf defined as a weighted sum
→ Recall from Sampling lecture
→ Response Surface Modeling lecture
Contd.

Even so, Whilst $P(X)$ can be evaluated sampling from it may be difficult.
Importance Sampling

Suppose we wish to evaluate

$$\int f(x) P(x) dx \quad \text{(e.g. moment calculation)}$$

$$\int f(x) \frac{P(x)}{Q(x)} Q(x) dx, \quad X^i \sim Q(x)$$

$$= \frac{1}{N} \sum_{i=1}^{N} f(x = X^i) w^i, \quad w^i = \frac{P(x = X^i)}{Q(x = X^i)}$$
So:

Sample from $Q \equiv$ Proposal distribution
Evaluate from $P \equiv$ the density
Apply importance weight $= w^i = \frac{P(X^i)}{Q(X^i)}$

Now let’s consider

\[
P(x) = \frac{\hat{P}(x)}{\int \hat{P}(x) dx} = \frac{\hat{P}(x)}{\mathcal{Z}_p}
\]

\[
Q(x) = \frac{\hat{Q}(x)}{\int \hat{Q}(x) dx} = \frac{\hat{Q}(x)}{\mathcal{Z}_q}
\]
So:

\[
\frac{1}{N} \frac{Z_q}{Z_p} \sum_{i=1}^{N} f(x = X^i) \hat{w}^i
\]

where

\[
\hat{w}^i = \frac{\hat{P}(x = X^i)}{\hat{Q}(x = X^i)} \text{ These are un-normalized “mere potentials”}
\]

It turns out:

\[
\frac{NZ_p}{Z_q} = \sum_i \hat{w}^i
\]

\[
\therefore f(x) P(x) dx \approx \frac{\sum_{i=1}^{N} f(x = X^i) \hat{w}^i}{\sum_j \hat{w}^j}
\]
Particle Filters

is just a “weighted sum”

\[
\frac{\sum_i f(X^i) \hat{w}^i}{\sum_j \hat{w}^j}
\]

Where a proposal distribution was used to get around sampling difficulties and the importance weights manage all the normalization.

⇒ It is important to select a good proposal distribution. Not one that focus on a small part of the state space and perhaps better than an uninformative prior.
Application of Importance Sampling to Bayesian Recursive Estimation

Particle Filter

$$P(X) \approx \frac{\sum_i \hat{w}^i \delta(X - X^i)}{\sum_j \hat{w}^j} = \sum_i w^i \delta(X - X^i)$$

$w^i$ is normalized.
Let’s consider again:

\[ X_k = f(X_{k-1}) + V_k \]
\[ Y_k = h(X_k) + \eta_k \]

A relationship between the observation and the state (measurement)

⇒ Additive noise, but can be generalized
Let's consider the joint distribution

\[ P(X_{0:k} \mid Y_{1:k}) \]

We may factor this distribution using particles
Chain Rule with Weights

\[
P(X_{0:k} | Y_{1:k}) = \sum_{i=1}^{N} w^i \delta(X_{0:k} - X^i_{0:k})
\]

\[
w^i \equiv \frac{P(X^i_{0:k} | Y_{1:k})}{Q(X^i_{0:k} | Y_{1:k})}
\]

And let’s factor \( P(X_{0:k} | Y_{1:k}) \) as

\[
P(X_{0:k} | Y_{1:k}) = \frac{P(Y_k | X_{0:k}, Y_{1:k-1}) P(X_{0:k} | Y_{1:k-1})}{P(Y_k | Y_{1:k-1})}
\]

\[
= \frac{P(Y_k | X_k) P(X_k | X_{k-1}) P(X_{k-1} | Y_{1:k-1})}{P(Y_k | Y_{1:k})}
\]
Suppose we pick

\[ Q(X_{0:k} \mid Y_{1:k}) = Q(X_k \mid X_{0:k-1}, Y_{1:k})Q(X_{0:k-1} \mid Y_{1:k-1}) \]

i.e. there is some kind of recursion on the proposal distribution. Further, if we approximate

\[ Q(X_k \mid X_{0:k-1}, Y_{1:k}) = Q(X_k \mid X_{k-1}, Y_k) \]

i.e. there is a Markov property.
Recursive Weight Updates

Then we may have found an update equation for the weights.

\[
P(X_{0:k} | Y_{1:k}) = \frac{P(Y_k | X_k) P(X_k | X_{k-1}) P(X_{0:k-1}, Y_{1:k-1})}{Q(Y_{1:k} | X_{0:k})}
\]

\[
w^i_k = \frac{P(Y_k | X^i_k) P(X^i_k | X^i_{k-1})}{Q(X^i_k | X^i_{k-1}, Y_k) P(Y_k | Y_{1:k-1})} \frac{P(X^i_{0:k-1}, Y_{1:k-1})}{Q(X^i_{0:k-1}, Y_{1:k-1})}
\]

\[= \frac{P(Y_k | X^i_k) P(X^i_k | X^i_{k-1})}{Q(X^i_k | X^i_{k-1}, Y_k) P(Y_k | Y_{1:k-1})} w^i_{k-1}
\]

\[\propto \frac{P(Y_k | X^i_k) P(X^i_k | X^i_{k-1})}{Q(X^i_k | X^i_{k-1}, Y_k)} w^i_{k-1}
\]
The Particle Filter

In the filtering problem

\[ P(X_k | Y_{1:k}) \]

\[ w_k^i \propto w_{k-1}^i \frac{P(Y_k | X_k^i) P(X_k^i | X_{k-1}^i)}{Q(X_k^i | X_{k-1}^i, Y_k)} \]

(\text{So}) \ P(X_k | Y_{1:k}) \approx \sum_{i=1}^{N} w_k^i \delta(X_k - X_k^i) \]

Where the \( x_k^i \sim Q(X_k | X_{k-1}^i, Y_k) \)

The method essentially draws particles from a proposal distribution and recursively update its weights.

⇒ No gaussian assumption

⇒ Neat
Algorithm Sequential Importance Sampling

Input: \( \{X^{i}_{k-1}, w^{i}_{k-1}\}, \ Y_{k} \ \ i = 1 \ldots N \)
for: \( i = 1 \ldots N \)
Draw: \( X^{i}_{k} \sim Q(X_{k} | X^{i}_{k-1}, Y_{k}) \)
\[
    w^{i}_{k} \propto w^{i}_{k-1} \frac{P(Y_{k} | X^{i}_{k})P(X^{i}_{k} | X^{i}_{k-1})}{Q(X^{i}_{k} | X^{i}_{k-1}, Y_{k})}
\]
end
BUT The Problem

In a few intervals one particle will have a non negligible weight; all but one will have negligible weights!

\[ \hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N}(w_k^i)^2} \]
\( \hat{N}_{\text{eff}} \equiv \text{Effective Sample size} \)

When \( \hat{N}_{\text{eff}} \ll N \rightarrow \text{Degeneracy sets in} \)

**Resampling is a way to address this problem**

**Main idea**

You can sample uniformly and set weights to obtain a representation. You can sample pdf to get particles and reset their weights.
Resampling algorithm

Uniform weights

Many points

Cdf(w)

Sample more probable states more

Particle Filters
Algorithm

Input \( \{X^i_k, w^i_k\} \)

1. Construct cdf
   \[
   \text{for } i = 2 : N \quad C_i \leftarrow C_{i-1} + w^i_k \text{ (sorted)}
   \]

2. Seed \( u_1 \sim U[0, N^{-1}] \)

3. for \( j = 1 : N \)
   \[
   u_j = u_1 + \frac{1}{N} (j - 1)
   \]
   \[
   i \leftarrow \text{find}(C_i \geq u_j)
   \]
   \[
   \hat{X}^j_i = X^i_k \quad w^j_i = \frac{1}{N}
   \]
   Set Parent of \( i^j \rightarrow i \)
So the resampling method can avoid degeneracy because it produces more samples for higher probability points

But Sample impoverishment may result; Too many samples too close $\rightarrow$ impoverishment or loss of diversity

$\Rightarrow$ MCMC may be a way out
Generic Particle filter

Input: \( \{X_{k-1}^i, w_{k-1}^i\}, Y_k \)

for \( i = 1 : N \)

\[
X_k^i \sim Q(X_k | X_{k-1}^i, Y_k)
\]

\[
w_k^i \leftarrow w_{k-1}^i \frac{P(Y_k | X_k^i) P(X_k^i | X_{k-1}^i)}{Q(X_k^i | X_{k-1}^i, Y_k)}
\]

end

\[
\eta = \sum_i w_k^i \\
w_k^i \leftarrow w_k^i / \eta
\]

\[
\hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^N (w_k^i)^2}
\]

If \( \hat{N}_{\text{eff}} < N_T \)

\( \{X_k, w_k^i\} \leftarrow \text{Resample} \ \{X_k^i, w_k^i\} \)
What is the optimal Q function? we try to minimize \( \sum_{i=1}^{N} (w_k^*)^2 \)

Then:

\[
Q^*(X_k | X_{k-1}^i, Y_k) = P(X_k | X_{k-1}^i, Y_k)
\]

\[
= \frac{P(Y_k | X_k, X_{k-1}^i)P(X_k | X_{k-1}^i)}{P(Y_k | X_{k-1}^i)}
\]

\[
w_k^i \propto w_{k-1}^i \frac{P(Y_k | X_k)P(X_k | X_{k-1}^i)}{P(Y_k | X_k)P(X_k | X_{k-1}^i)}
\]

\[
\propto w_{k-1}^i \int_{X_k} P(Y_k | X_k)P(X_k | X_{k-1}^i) dX_k
\]

Not easy to do!
Asymptotically:
\[ \hat{Q} \sim P(X_k|X_{k-1}^i) \leftarrow \text{Common choice } Q \equiv P(X_k|X_{k-1}^i) \]

Sometimes feasible to use proposal from process noise

Then

\[ w_k^i \propto w_{k-1}^i P(Y_k|X_k^i) \]

If resampling is done at every step:

\[ w_k^i \propto p(Y_k|X_k^i) \]

\((w_{k-1}^i \propto \frac{1}{N})\)
**SIR - Sampling Importance Resampling**

**Input** \( \{X^i_{k-1}, w^i_{k-1}\}, Y_k \)

for \( i = 1 : N \)

\( X^i_k \sim P(X_k | X^i_{k-1}) \)

\( w^i_k = P(Y_k | X^i_k) \)

end

\( \eta = \sum_i w^i_k \)

\( w^i_k = w^i_k / \eta \)

\( \{x^i_k, w^i_k\} \leftarrow \text{Resample} \ [\{X^i_k, w^i_k\}] \)
Example

\[ X_k = \frac{X_{k-1}}{2} + \frac{25X_{k-1}}{1 + X_{k-1}^2} + 8 \cos(1.2k) + \nu_{k-1} \]

\[ Y_k = \frac{X_k^2}{W} + \eta_k \]

\( \eta_k \sim N(0, R) \)

\( \nu_{k-1} \sim N(0, Q_{k-1}) \)