BAYESIAN STATE ANALYSIS ON NONLINEAR DYNAMICAL SYSTEMS

Outline:

- **Particle filter**
- **Particle smoother**

Particle filter

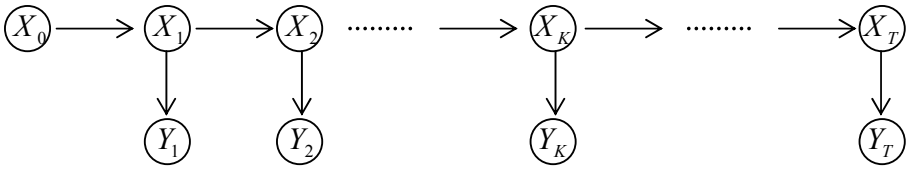
Introduction:

In the case that the state-space model class is nonlinear, Kalman filter and RTS smoother breaks down. Although it is always possible to linearize the nonlinear model so that Kalman filter and RTS smoother can still apply approximately, they can be not reliable. On the other hand, stochastic simulation approaches are not limited to linear model classes and can be adopted to draw samples of the state conditioning on the observation even if the model is nonlinear. One of such stochastic simulation approaches is called particle filter.

Let's now consider a more general version of state-space equations:

$$X_k = \phi_{k-1}\left(X_k, u_k, W_k\right) \qquad Y_k = \varphi_k\left(X_k, u_k, V_k\right)$$

where u_k = known system input at time k, W_k, V_k = modeling error, $W_i \perp W_j \left(i \neq j\right)$, $V_i \perp V_j \left(i \neq j\right)$, $W_i \perp V_j$, $X_0 \perp W_j$, $X_0 \perp V_j$. Note that this model class creates a hidden Markov chain.



For the sake of simplicity, let's consider a simpler model class in this lecture:

$$X_k = \phi_{k-1}\left(X_k, u_k\right) + W_k \qquad Y_k = \varphi_k\left(X_k, u_k\right) + V_k$$

where $W_k \sim N\left(0, \Sigma_W\right)$, $V_k \sim N\left(0, \Sigma_V\right)$. Similar to Kalman filter, the goal of particle filter is to draw samples from $f(x_k \mid \hat{Y}_{1:k})$, where $\hat{Y}_{1:k} = \{\hat{Y}_1, ..., \hat{Y}_k\}$ is the data up to time k. Different from Kalman filter, this posterior PDF is not Gaussian. In the

following, we adopt the following notation: $x_{1:k} = \{x_1, ..., x_k\}$ and $X_{1:k} = \{X_1, ..., X_k\}$. The particle filter algorithm provides a way of obtaining samples of $f(x_{1:k+1} \mid \hat{Y}_{1:k+1})$ given the samples of $f(x_{1:k} \mid \hat{Y}_{1:k})$ and the new observation $\hat{Y}_{k+1}$. One can see that once this algorithm is finished, we can obtain the samples of $f(x_{1:k} \mid \hat{Y}_{1:k})$ for all k recursively starting from the samples of $f(x_0 \mid \hat{Y}_{1:0}) \equiv f(x_0)$. Note that once we get the samples $\left\{\hat{X}_{1:k}^i : i = 1, ..., N\right\}$ from $f(x_{1:k} \mid \hat{Y}_{1:k})$, we just keep the $\left\{\hat{X}_k^i : i = 1, ..., N\right\}$ part of the samples; that will give us the samples that are distributed as $f(x_k \mid \hat{Y}_{1:k})$. Before we describe the algorithm, let's look at the Bayesian derivations. Given the samples $\left\{\hat{X}_{1:k}^i : i = 1, ..., N\right\}$, $f(x_{1:k} \mid \hat{Y}_{1:k})$ can be represented approximately by

$$f\left(x_{1:k} \mid \hat{Y}_{1:k}\right) \approx \frac{1}{N}\sum_{i=1}^{N} \delta\left(x_{1:k} - \hat{X}_{1:k}^i\right)$$

Note that this appromiation is in the sense of the Law of Large Number, i.e.

$$E\left(g\left(X_{1:k}\right) \mid \hat{Y}_{1:k}\right) \approx \frac{1}{N}\sum_{i=1}^{N} g\left(\hat{X}_{1:k}^i\right) = \int g\left(x_{1:k}\right) \cdot \frac{1}{N}\sum_{i=1}^{N} \delta\left(x_{1:k} - \hat{X}_{1:k}^i\right) dx_{1:k}$$

From the Bayes rule, we have

$$\begin{aligned}
f\left(x_{1:k+1} \mid \hat{Y}_{1:k+1}\right) &= \frac{f\left(x_{1:k+1}, \hat{Y}_{1:k+1}\right)}{f\left(\hat{Y}_{1:k+1}\right)} = \frac{f\left(x_{1:k}, x_{k+1}, \hat{Y}_{1:k}, \hat{Y}_{k+1}\right)}{f\left(\hat{Y}_{1:k}, \hat{Y}_{k+1}\right)} \\
&= \frac{f\left(x_{k+1}, \hat{Y}_{k+1} \mid x_{1:k}, \hat{Y}_{1:k}\right)}{f\left(\hat{Y}_{1:k}, \hat{Y}_{k+1}\right)} \cdot f\left(x_{1:k}, \hat{Y}_{1:k}\right) \\
&= \frac{f\left(x_{k+1}, \hat{Y}_{k+1} \mid x_{1:k}, \hat{Y}_{1:k}\right)}{f\left(\hat{Y}_{k+1} \mid \hat{Y}_{1:k}\right)} \cdot f\left(x_{1:k} \mid \hat{Y}_{1:k}\right) \\
&= \frac{f\left(\hat{Y}_{k+1} \mid x_{k+1}, x_{1:k}, \hat{Y}_{1:k}\right) f\left(x_{k+1} \mid x_{1:k}, \hat{Y}_{1:k}\right)}{f\left(\hat{Y}_{k+1} \mid \hat{Y}_{1:k}\right)} \cdot f\left(x_{1:k} \mid \hat{Y}_{1:k}\right) \\
&= \frac{f\left(\hat{Y}_{k+1} \mid x_{k+1}\right) f\left(x_{k+1} \mid x_k\right)}{f\left(\hat{Y}_{k+1} \mid \hat{Y}_{1:k}\right)} \cdot f\left(x_{1:k} \mid \hat{Y}_{1:k}\right) \\
&\approx \frac{f\left(\hat{Y}_{k+1} \mid x_{k+1}\right) f\left(x_{k+1} \mid x_k\right)}{f\left(\hat{Y}_{k+1} \mid \hat{Y}_{1:k}\right)} \cdot \frac{1}{N}\sum_{i=1}^{N} \delta\left(x_{1:k} - \hat{X}_{1:k}^i\right)
\end{aligned}$$

Note that the new data $\hat{Y}_{k+1}$ plays a major role in the equations. There is in fact a simple way of drawing samples from $f\left(x_{1:k+1} \mid \hat{Y}_{1:k+1}\right)$: (1) draw a sample $\hat{X}_{1:k}$ from $f\left(x_{1:k} \mid \hat{Y}_{1:k}\right) \approx \frac{1}{N}\sum_{i=1}^{N} \delta\left(x_{1:k} - \hat{X}_{1:k}^{i}\right)$ and (2) draw a sample $\hat{X}_{k+1}$ from $\frac{f\left(\hat{Y}_{k+1} \mid x_{k+1}\right) f\left(x_{k+1} \mid \hat{X}_k\right)}{f\left(\hat{Y}_{k+1} \mid \hat{Y}_{1:k}\right)}$ so that the combined sample $\hat{X}_{1:k+1} \equiv \left\{\hat{X}_{1:k}, \hat{X}_{k+1}\right\}$ should be approximately distributed as $f\left(x_{1:k+1} \mid \hat{Y}_{1:k+1}\right)$. However, this is not doable directly because the PDF $\frac{f\left(\hat{Y}_{k+1} \mid x_{k+1}\right) f\left(x_{k+1} \mid \hat{X}_k\right)}{f\left(\hat{Y}_{k+1} \mid \hat{Y}_{1:k}\right)}$ cannot be sampled directly.

On the other hand, one can employ the sample-importance resampling (SIR) technique to sample from $f\left(x_{1:k+1} \mid \hat{Y}_{1:k+1}\right)$. Let us choose the following importance sampling PDF:

$$q\left(x_{1:k+1}\right) = q_0\left(x_{k+1} \mid x_k, \hat{Y}_{k+1}\right) \cdot \frac{1}{N}\sum_{i=1}^{N} \delta\left(x_{1:k} - \hat{X}_{1:k}^{i}\right)$$

where $q_0\left(x_{k+1} \mid x_k, \hat{Y}_{k+1}\right)$ is some PDF of X_{k+1}. The sequential way of doing SIR through time is called the particle filter.

Algorithm: particle filter

1. Let $\left\{\hat{X}_{1:k}^{i} : i = 1, ..., N\right\}$ be the samples distributed as $f(x_{1:k} \mid \hat{Y}_{1:k})$.
2. Sample $\left\{\bar{X}_{1:k+1}^{i} : i = 1, ..., N\right\}$ from $q\left(x_{1:k+1}\right)$, i.e. draw $\bar{X}_{1:k}^{i}$ from $\frac{1}{N}\sum_{i=1}^{N} \delta\left(x_{1:k} - \hat{X}_{1:k}^{i}\right)$ and draw $\bar{X}_{k+1}^{i}$ from $q_0\left(x_{k+1} \mid \bar{X}_k^{i}, \hat{Y}_{k+1}\right)$, and do this for i = 1, …, N.
3. Compute the normalized importance weights:

$$w^{(i)} \equiv \frac{f\left(\hat{Y}_{k+1} \mid \bar{X}_{k+1}^{i}\right) f\left(\bar{X}_{k+1}^{i} \mid \bar{X}_k^{i}\right)}{q_0\left(\bar{X}_{k+1}^{i} \mid \bar{X}_k^{i}, \hat{Y}_{k+1}\right)} \Bigg/ \sum_{j=1}^{N} \frac{f\left(\hat{Y}_{k+1} \mid \bar{X}_{k+1}^{j}\right) f\left(\bar{X}_{k+1}^{j} \mid \bar{X}_k^{j}\right)}{q_0\left(\bar{X}_{k+1}^{j} \mid \bar{X}_k^{j}, \hat{Y}_{k+1}\right)}$$

4. Resampling: Let $\hat{X}_{1:k+1}^{j} = \bar{X}_{1:k+1}^{i}$ with probability $w^{(i)}$ for $j = 1, \cdots, N$, then

$\left\{\hat{X}^j_{1:k+1} : j = 1,\cdots,N\right\}$ will be approximately distributed as $f\left(x_{1:k+1} \mid \hat{Y}_{1:k+1}\right)$.

Remarks:

1. Since the particle filter algorithm is based on SIR, so it is not good for high dimensional X_k.
2. From the algorithm, it is clear that the samples before time k $\left\{\hat{X}^i_{1:k-1} : i = 1,...,N\right\}$ are not needed, so in the real application, the particle filter algorithm can be a full online algorithm, i.e. no need to store the samples older than the previous time step in the memory. So we can actually change the algorithm into the following:

   I. Let $\left\{\hat{X}^i_k : i = 1,...,N\right\}$ be the samples distributed as $f(x_k \mid \hat{Y}_{1:k})$.

   II. Draw $\bar{X}^i_k$ from $\frac{1}{N}\sum_{i=1}^{N}\delta\left(x_k - \hat{X}^i_k\right)$ and draw $\bar{X}^i_{k+1}$ from $q_0\left(x_{k+1} \mid \bar{X}^i_k, \hat{Y}_{k+1}\right)$, and do this for i = 1, …, N.

   III. Compute the normalized importance weights:

$$w^{(i)} \equiv \frac{f\left(\hat{Y}_{k+1} \mid \bar{X}^i_{k+1}\right) f\left(\bar{X}^i_{k+1} \mid \bar{X}^i_k\right)}{q_0\left(\bar{X}^i_{k+1} \mid \bar{X}^i_k, \hat{Y}_{k+1}\right)} \Bigg/ \sum_{j=1}^{N} \frac{f\left(\hat{Y}_{k+1} \mid \bar{X}^j_{k+1}\right) f\left(\bar{X}^j_{k+1} \mid \bar{X}^j_k\right)}{q_0\left(\bar{X}^j_{k+1} \mid \bar{X}^j_k, \hat{Y}_{k+1}\right)}$$

   IV. Resampling: Let $\hat{X}^j_{k+1} = \bar{X}^i_{k+1}$ with probability $w^{(i)}$ for $j = 1,\cdots,N$, then $\left\{\hat{X}^j_{k+1} : j = 1,\cdots,N\right\}$ will be approximately distributed as $f\left(x_{k+1} \mid \hat{Y}_{1:k+1}\right)$.

3. We can replace the random sampling of $\bar{X}^i_k$ in Step II with a deterministic one: just set $\bar{X}^i_k = \hat{X}^i_k$ for all i.
4. One can, in principle, skip the resampling step, but when doing so, the importance weights need to be accumulated to the next time step. However, this is not feasible because if we do so, the weights will ultimately become highly non-uniform so the effective number of samples will be very small. Therefore, the resampling step is essential. One way of interpreting the resampling step is to consider it as a splitting step according to the sample importance, i.e. the samples with large weights will be duplicated and propagated continuously, but the samples with small weights may be eliminated.
5. In fact, it can be shown that the particle filter algorithm can asymptotically draw

samples that are distributed as $f(x_k \mid \hat{Y}_{1:k})$. This is because SIR is asymptotically correct.

6. A critical question in the particle filter algorithm is how to select the importance PDF $q_0\left(x_{k+1} \mid \bar{X}_k^i, \hat{Y}_{k+1}\right)$. A popular choice is to let $q_0\left(x_{k+1} \mid \bar{X}_k^i, \hat{Y}_{k+1}\right)$ be equal to $f\left(x_{k+1} \mid \bar{X}_k^i\right)$ (a Gaussian PDF in our example) because we know how to directly sample and evaluate it. Although this is a convenient choice, but the resulting importance weights may be highly non-uniform. This undesirable situation may occur when the main support region of $f\left(x_{k+1} \mid \bar{X}_k^i\right)$ is very different from that of $f\left(\hat{Y}_{k+1} \mid x_{k+1}\right) f\left(x_{k+1} \mid \bar{X}_k^i\right)$.

   The optimal choice is to let $q_0\left(x_{k+1} \mid \bar{X}_k^i, \hat{Y}_{k+1}\right) = \dfrac{f\left(\hat{Y}_{k+1} \mid x_{k+1}\right) f\left(x_{k+1} \mid \hat{X}_k\right)}{f\left(\hat{Y}_{k+1} \mid \hat{Y}_{1:k}\right)}$ because the importance weights will be completely uniform. However, it is not feasible because we don't know how to directly sample and evaluate it.

   An interesting choice is to let $q_0\left(x_{k+1} \mid \bar{X}_k^i, \hat{Y}_{k+1}\right)$ be a Gaussian PDF that is similar to $\dfrac{f\left(\hat{Y}_{k+1} \mid x_{k+1}\right) f\left(x_{k+1} \mid \hat{X}_k\right)}{f\left(\hat{Y}_{k+1} \mid \hat{Y}_{1:k}\right)}$. This important sampling PDF can be obtained using Extend Kalman filter (EKF): Note that the prior PDF $f\left(x_{k+1} \mid \hat{X}_k\right)$ is Gaussian. If we employ the linearization technique in EKF to linearize the likelihood function $f\left(\hat{Y}_{k+1} \mid x_{k+1}\right)$ so that it's linear in x_{k+1}, we can obtain a Gaussian approximation of the posterior PDF $\dfrac{f\left(\hat{Y}_{k+1} \mid x_{k+1}\right) f\left(x_{k+1} \mid \hat{X}_k\right)}{f\left(\hat{Y}_{k+1} \mid \hat{Y}_{1:k}\right)}$ and use that as our importance sampling PDF $q_0\left(x_{k+1} \mid \bar{X}_k^i, \hat{Y}_{k+1}\right)$.

7. After we obtain samples $\left\{\hat{X}_k^i : i = 1, \cdots, N\right\}$, $E\left[g(X_k) \mid \hat{Y}_{1:k}\right]$ can be estimated as

$$E\left[g(X_k) \mid \hat{Y}_{1:k}\right] \approx \frac{1}{N} \sum_{i=1}^{N} g(\hat{X}_k^i)$$

Particle smoother

Introduction:

The goal of particle smoother is to sample from $f\left(x_{1:T} \mid \hat{Y}_{1:T}\right)$, i.e. draw the complete time history of the state conditioning on the data. It role is very similar to the backward sampler for linear Gaussian model classes, but particle smoother is applicable to nonlinear model classes. Note that if we know how to sample from $f\left(x_{1:T} \mid \hat{Y}_{1:T}\right)$, we know how to sample from $f\left(x_k \mid \hat{Y}_{1:T}\right)$ for every *k*, i.e. discard all samples except the state sample at time *k*. Note that the particle smoother requires the results from particle filter: $\left\{\hat{X}_k^i : i=1,\cdots,N, k=1,\cdots,T\right\}$, where $\hat{X}_k^i$ is distributed as $f\left(x_k \mid \hat{Y}_{1:k}\right)$.

Consider the following equation:

$$
\begin{aligned}
& f\left(x_{1:T} \mid \hat{Y}_{1:T}\right) \\
&= f\left(x_T \mid \hat{Y}_{1:T}\right) f\left(x_{T-1} \mid x_T, \hat{Y}_{1:T}\right) \cdots f\left(x_k \mid x_{k+1}, x_{k+2}, \cdots, x_T, \hat{Y}_{1:T}\right) \cdots f\left(x_0 \mid x_1, \cdots, x_T, \hat{Y}_{1:T}\right) \\
&= f\left(x_T \mid \hat{Y}_{1:T}\right) f\left(x_{T-1} \mid x_T, \hat{Y}_{1:T-1}\right) \cdots f\left(x_k \mid x_{k+1}, \hat{Y}_{1:k}\right) \cdots f\left(x_0 \mid x_1\right)
\end{aligned}
$$

One way of sampling from $f\left(x_{1:T} \mid \hat{Y}_{1:T}\right)$ is to first draw $\hat{X}_T^S$ (the 'S' superscript denotes smoother samples to differentiate them from the samples from the particle filter) from $f\left(x_T \mid \hat{Y}_{1:T}\right)$ and then draw $\hat{X}_{T-1}^S$ from $f\left(x_{T-1} \mid \hat{X}_T^S, \hat{Y}_{1:T-1}\right)$, then draw $\hat{X}_{T-2}^S$ from $f\left(x_{T-2} \mid \hat{X}_{T-1}^S, \hat{Y}_{1:T-2}\right)$, and so on to obtain $\hat{X}_{0:T}^S$, distributed as $f\left(x_{1:T} \mid \hat{Y}_{1:T}\right)$. Drawing samples from $f\left(x_{1:T} \mid \hat{Y}_{1:T}\right)$ is easy with the results from the particle filter; however, $f\left(x_{T-1} \mid \hat{X}_T^S, \hat{Y}_{1:T-1}\right)$ cannot be directly sampled.

Consider the following equation:

$$
\begin{aligned}
f\left(x_{T-1} \mid \hat{X}_T^S, \hat{Y}_{1:T-1}\right) &= \frac{f\left(x_{T-1}, \hat{X}_T^S, \hat{Y}_{1:T-1}\right)}{f\left(\hat{X}_T^S, \hat{Y}_{1:T-1}\right)} = \frac{f\left(\hat{X}_T^S \mid x_{T-1}, \hat{Y}_{1:T-1}\right) f\left(x_{T-1}, \hat{Y}_{1:T-1}\right)}{f\left(\hat{X}_T^S, \hat{Y}_{1:T-1}\right)} \\
&= \frac{f\left(\hat{X}_T^S \mid x_{T-1}\right)}{f\left(\hat{X}_T^S \mid \hat{Y}_{1:T-1}\right)} \cdot f\left(x_{T-1} \mid \hat{Y}_{1:T-1}\right) \approx \frac{f\left(\hat{X}_T^S \mid x_{T-1}\right)}{f\left(\hat{X}_T^S \mid \hat{Y}_{1:T-1}\right)} \cdot \frac{1}{N} \sum_{i=1}^{N} \delta\left(x_{T-1} - \hat{X}_{T-1}^i\right)
\end{aligned}
$$

$$\approx \sum_{i=1}^{N} \frac{1}{N} \frac{f\left(\hat{X}_T^S \mid \hat{X}_{T-1}^i\right)}{f\left(\hat{X}_T^S \mid \hat{Y}_{1:T-1}\right)} \delta\left(x_{T-1} - \hat{X}_{T-1}^i\right)$$

$$\approx \sum_{i=1}^{N} \frac{f\left(\hat{X}_T^S \mid \hat{X}_{T-1}^i\right)}{\sum_{j=1}^{N} f\left(\hat{X}_T^S \mid \hat{X}_{T-1}^j\right)} \delta\left(x_{T-1} - \hat{X}_{T-1}^i\right)$$

So $\hat{X}_{T-1}^S$ has probability of $\dfrac{f\left(\hat{X}_T^S \mid \hat{X}_{T-1}^i\right)}{\sum_{j=1}^{N} f\left(\hat{X}_T^S \mid \hat{X}_{T-1}^j\right)}$ to be equal to $\hat{X}_{T-1}^i$. The same concept applies when sampling from $\hat{X}_{T-2}^S$, and so on.

Algorithm: particle smoother

1. Do particle filter first to get the filtering samples $\left\{\hat{X}_k^i : i = 1,\cdots,N, k = 1,\cdots,T\right\}$.
2. Resample $\hat{X}_T^S = \hat{X}_T^j$ wit probability $\dfrac{1}{N}$
3. Compute $w_{T-1}^j = f\left(\hat{X}_T^S \mid \hat{X}_{T-1}^j\right)$ for $j = 1,\cdots,N$
4. Resample $\hat{X}_{T-1}^S = \hat{X}_{T-1}^j$ with probability $\dfrac{w_{T-1}^j}{\sum_{i=1}^{N} w_{T-1}^i}$
5. Repeat Step 3-4 for $k = T-1, T-2,\cdots,0$ to get $\left\{\hat{X}_T^S, \hat{X}_{T-1}^S,\cdots,\hat{X}_0^S\right\} \sim f\left(x_{1:T} \mid \hat{Y}_{1:T}\right)$
6. Repeat Step 2-5 M times to get M time-history samples of $f\left(x_{1:T} \mid \hat{Y}_{1:T}\right)$.

Remarks:

1. Since the particle smoother is based on the particle filter results, so it is also asymptotically correct and not good for high dimensional X_k.
2. M is better not to be large than N since otherwise it seems like getting something from nothing.
3. After we obtain samples $\left\{\hat{X}_{1:T}^{S,i} : i = 1,\cdots,M\right\}$, $E\left[g(X_{1:T}) \mid \hat{Y}_{1:T}\right]$ can be estimated as

   $$E\left[g(X_{1:T}) \mid \hat{Y}_{1:T}\right] \approx \frac{1}{M} \sum_{i=1}^{M} g(\hat{X}_{1:T}^{S,i})$$