ECE 830 Fall 2011 Statistical Signal Processing

instructor: R. Nowak

Lecture 23: Density Propagation, Kalman and Particle Filtering

Density Propagation

The density propagation algorithm discussed in the last lecture involves the following ingredients and operates according to the block diagram below.

Observation:

$$L_n(z_n|x_n) := p(z_n|x_n)$$

Prior:

$$S_n(x_n|x_{n-1}) := p(x_n|x_{n-1}) \;\; \text{“}1st\ order\ Markov\ model\text{”}$$

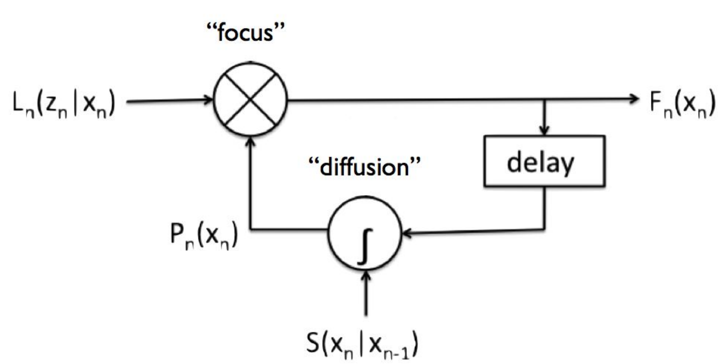
$$P_1(x_1) := p(x_1) \quad prior\ probability\ on\ initial\ x_1$$

Posterior:

$$F_n(x_n) := p(x_n|z_1, ..., z_n)$$

Prediction Distribution:

$$P_n(x_n) := p(x_n|z_1, ..., z_{n-1})$$



The Kalman Filter

The Kalman filter is the most famous instantiation of density propagation. It assumes linear dynamics and Gaussian distributions, which lead to very simple linear-algebraic operations.

Observation Model:

$$z_n \;=\; C\,x_n \;+\; D\,v_n \;,\; v_n \overset{iid}{\sim} \mathcal{N}(0, I_\ell)$$

where C is a $p \times d$ matrix, D is $p \times \ell$, $z_n \in \mathbb{R}^p$ and $x_n \in \mathbb{R}^d$.

Prior:

$$x_n = \underset{d\times d}{A} x_{n-1} + \underset{d\times k}{B} u_n \ , \quad \text{with } u_n \overset{iid}{\sim} \mathcal{N}(0, I_k) \text{ and } x_1 \sim \mathcal{N}(\mu_{P_1}, V_{P_1})$$

Now in this case $P_1(x_1)$, $S_n(x_n|x_{n-1})$ and $L_n(z_n|x_n)$ are all multivariate Gaussian, it follows that $F_n(x_n)$ and $P_n(x_n)$ are also Gaussian (since products and convolutions of Gaussians are also Gaussian). These distributions are easily computed using the Gauss-Markov Theorem. Therefore $F_n(x_n)$ is $\mathcal{N}(\mu_{F_n}, V_{F_n})$ and $P_n(x_n)$ is $\mathcal{N}(\mu_{P_n}, V_{P_n})$, and we only need to determine the means and covariances which have simple closed-form linear algebraic expressions.

Kalman Filter (special case of density propogation)

Start with prediction distribution $P_n(x_n)$ of the form $\mathcal{N}(\mu_{P_n}, V_{P_n})$.

Observe z_n and compute $F_n(x_n)$. Note that $z_n|x_n \sim \mathcal{N}(Cx_n, DD^T)$ and $P_n(x_n)$ is $\mathcal{N}(\mu_{P_n}, V_{P_n})$. A simple application of the Gauss-Markov Theorem shows

$$\begin{aligned} \mu_{F_n} &= \mu_{P_n} + V_{P_n} C^T (C V_{P_n} C^T + DD^T)^{-1} (z_n - C\mu_{P_n}) \\ V_{F_n} &= I_d - V_{P_n} C^T (C V_{P_n} C^T + DD^T)^{-1} C V_{P_n} \end{aligned}$$

Compute next prediction distribution $P_{n+1}(x_{n+1})$. This amounts to simply applying the dynamical model (a linear transformation) to F_n.

$$\begin{aligned} \mu_{P_{n+1}} &= A\mu_{F_n} \\ V_{P_{n+1}} &= AV_{F_n}A^T + BB^T \end{aligned}$$

Example 1 *It is clear that the convolution of two Gaussian densities is Gaussian (this follows from the fact that the sum of two independent Gaussians is Gaussian, and the density of the sum is the convolution of the two individual densities). The fact that products of Gaussian densities have a Gaussian form is less obvious. Here is a simple scalar example showing that the product of two Gaussian densities is also Gaussian in form (also see the Gauss-Markov Theorem). Suppose*

$$\begin{aligned} P_n(x_n) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{x_n^2}{2}} \\ L_n(z_n|x_n) &= \frac{1}{\sqrt{4\pi}} e^{\frac{-(x_n - z_n)^2}{4}} \end{aligned}$$

The posterior is then

$$\begin{aligned} F_n(x_n) &\propto P_n(x_n)\, L_n(z_n|x_n) \ \propto \ e^{-\frac{x_n^2}{2} - \frac{(x_n-z_n)^2}{4}} \\ &= e^{-\left(\frac{2x_n^2 + x_n^2 - 2x_n z_n + z_n^2}{4}\right)} \\ &= e^{-\left(\frac{3x_n^2 - 2x_n z_n + z_n^2}{4}\right)} \\ &= e^{-\left(\frac{\left(\sqrt{3}x_n - \frac{z_n}{\sqrt{3}}\right)^2 + const}{4}\right)} \\ &\propto e^{-\left(\frac{\left(\sqrt{3}x_n - \frac{z_n}{\sqrt{3}}\right)^2}{4}\right)} \\ &= e^{-\left(\frac{\left(x_n - \frac{z_n}{3}\right)^2}{\frac{4}{3}}\right)} \end{aligned}$$

Therefore, the posterior distribution $F_n(x_n)$ is $\mathcal{N}(\frac{z_n}{3}, \frac{2}{3})$.

Kalman Filter Output

Minimum Mean Squared Error (MMSE) estimator of x_n is

$$\begin{aligned} \widehat{x}_n &= \arg\min_{\widetilde{x}} \mathbb{E}\left[||x_n - \widetilde{x}||_2^2 \,|\, z_1, ..., z_n \right] \\ &= \text{mean of posterior } F_n \\ &= \int x\, F_n(x)\, dx \;=\; \mu_{F_n} \end{aligned}$$

The Extended Kalman Filter

The Extended Kalman Filter (EKF) can handle nonlinear observation models and dynamics by linearizing the current estimates at each step. The EKF is used routinely in GPS systems.
Prior:

$$x_n = \phi(x_{n-1}) + Bu_n$$

Likelihood:

$$z_n = \psi(x_n) + Dv_n$$

$$\textit{where } \phi, \psi \textit{ nonlinear}$$

Linearization:

$$z_n \approx \underbrace{\psi(\mu_{P_n}) + \psi'(\mu_{P_n})(x_n - \mu_{P_n})}_{\textit{Taylor series approximation of } \psi(x_n) \textit{ at } \mu_{P_n}} + Dv_n$$

$$x_n \approx \underbrace{\phi(\mu_{F_n}) + \phi'(\mu_{F_n})(x_n - \mu_{F_n})}_{\textit{Taylor series approximation of } \phi(x_n) \textit{ at point } \mu_{F_n}} + Bu_n$$

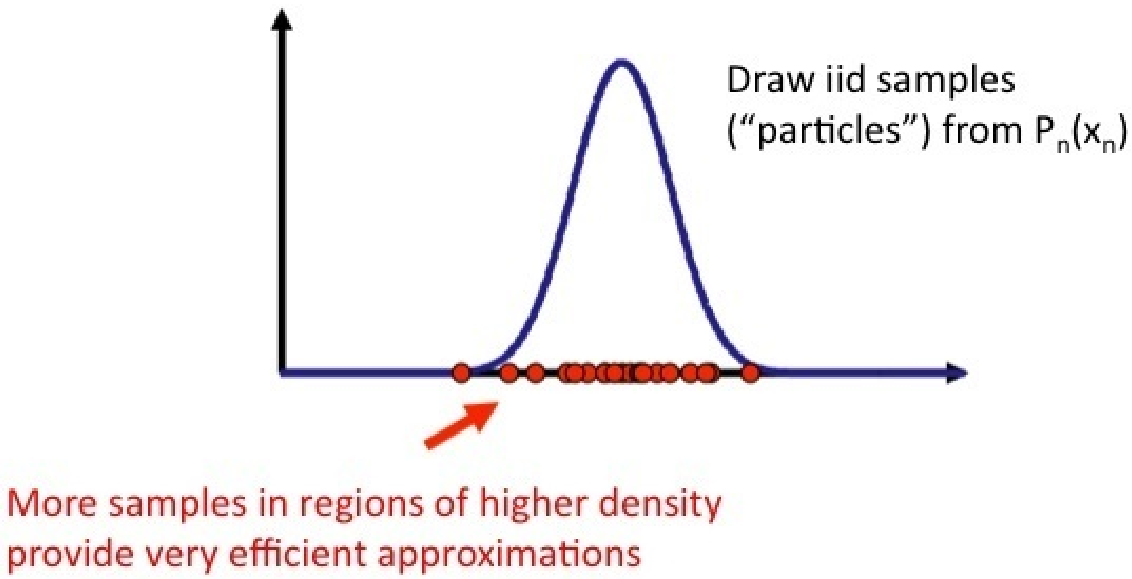
Alternatives to the Extended Kalman Filter (EKF)

1. The Unscented Kalman Filter is based on nonlinear transformations, rather than linearization
2. Point-mass filters replace continuous distributions with discrete point mass function approximations. All updates can be computed as sums. The problem is that this can be very computationally demanding in high dimensions.
3. Gaussian-Mixture Approximations can be used instead of ideal densities, and updates in terms of Gaussian mixtures are relatively easy to compute.
4. Particle Filters are a Monte Carlo version of the point-mass filtering idea.

Particle Filters

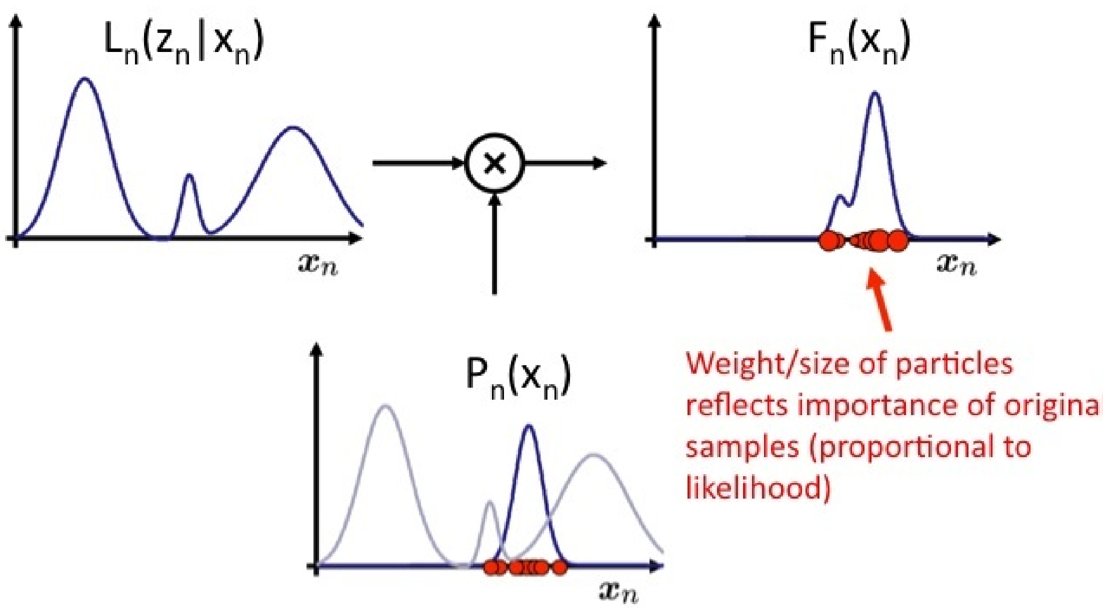
Particle Filters also employ discrete approximations of the underlying continuous distributions, but the discretization points are drawn randomly from the distributions (rather than on a deterministic grid). The key idea is that while the exact density propagation calculations can become intractable if the dynamics are nonlinear and/or the statistics are non-Gaussian, it is often still easy to 'simulate' the system. Particle filtering essentially runs many simulations of the dynamics in parallel, guided by the data as it is observed.

Example Particles for $P_n(x_n)$.

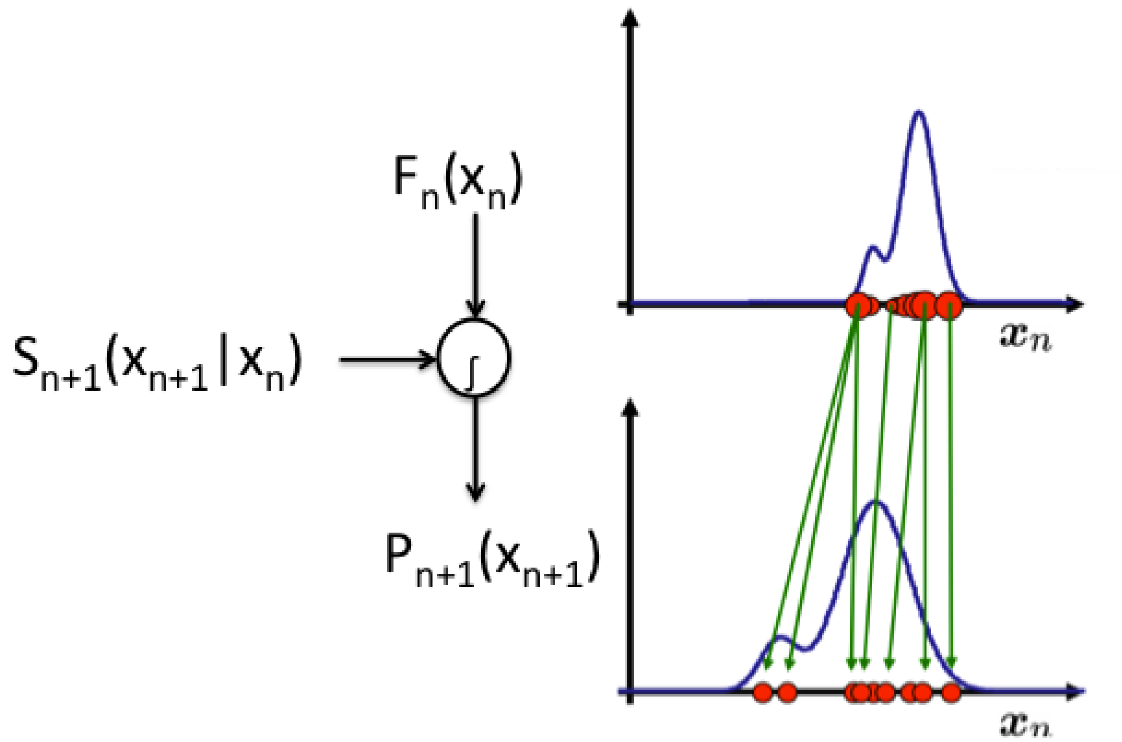


Condensation Algorithm (Isard & Blake 1998)

Start with samples from $P_n(x_n)$. Assume that the likelihood of a point x is easy to evaluate.

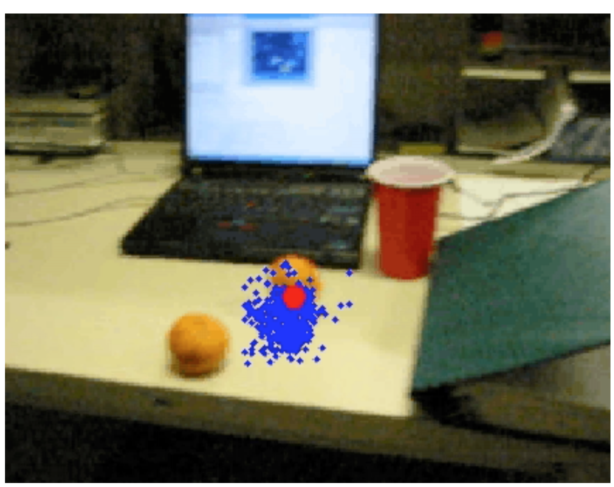


Resample according to importance weights (i.e., generate more new particles at points of larger likelihood) and simulate the dynamics for each. The key is that simulating the dynamics is usually very simple.



Essentially, the particles form a Monte Carlo sample that "tracks" the underlying densities. There are asymptotic (large numbers of particles) and non-asymptotic analyses that show particle filters can work very well. They are today's method of choice for nonlinear density propagation.

Example Orange Tracker As an example of particle filtering, consider the problem of tracking an orange as it rolls and moves in a video sequence. The orange can be tracked even though observation process is highly nonlinear (e.g., the orange may not be visible when it is occluded or moves out of the field of view).



<u>Prior:</u>

$$x_n = \begin{bmatrix} p_n \\ v_n \end{bmatrix} \begin{matrix} \textit{position} \\ \textit{velocity} \end{matrix} \; @ \; \textit{time } n \; (4x1 \; \textit{state})$$

$$\begin{aligned} x_n &= Ax_n + B\mu_n \\ &= \begin{bmatrix} I & \Delta \\ 0 & I \end{bmatrix} \begin{bmatrix} p_n \\ v_n \end{bmatrix} + \begin{bmatrix} 0 \\ \sigma^2 \end{bmatrix} \mu_n \;, \; \mu_n \overset{iid}{\sim} \mathcal{N}(0,1) \end{aligned}$$

<u>Observation:</u>

$$z_n = \psi(x_n)$$

where z_n is an $m \times m$ image of the "orange" color at every pixel and $\psi(x_n)$ is a highly nonlinear function of x_n since the orange can be occluded or move out of the scene.