24. Density Propagation
ECE 830, Spring 2014
Dynamic Filtering

In many applications we want to track a time-varying (dynamic) phenomenon.

Example: Tracking temperature or humidity in a museum room with an inaccurate device

Key: Temperature changes slowly with time so we should be able to average across time to obtain better estimates. How to do this? **Model dynamics of temperature changes and noise/uncertainties in measurement.**
Dynamical State Equation (Prior)

Let $x_1, x_2, \ldots$ denotes quantity ("state") of interest. The state is changing over time and we will model this variation stochastically as follows. The state at time $n$ depends causally on the past.

Let

$$p(x_n | x_{n-1}, x_{n-2}, \ldots, x_1)$$

denote the conditional distribution of the state at time $n$ given all the past states. This distribution is a $n$-variate function, and as $n$ grows it becomes more and more complex (to specify, to compute, etc).

A reasonable simplifying assumption is to assume that the probability distribution of the state at time $n$ depends only on value of the state at time $n - 1$. 
Definition: Markovian assumption

\[ p(x_n|x_{n-1}, \ldots, x_1) = p(x_n|x_{n-1}) . \]

Note that \( p(x_n|x_{n-1}) \) is bivariate and therefore much simpler than the general causal model.

To define the state process we must to specify

1. \( p(x_1) \), the “initial state” distribution
2. \( p(x_n|x_{n-1}) \), \( n = 2, 3, \ldots \), the state transition probability density functions

This is illustrated in the following example.
Example: Santa Tracker

On December 25th, legend has it that Santa Claus makes his way around the globe, delivering toys to all the good girls and boys. Tracking Santa’s delivery trip has attracted considerable interest by the signal processing research community in recent years, see http://www.noradsanta.org/. Here is a simple approach to the problem.

\[
\begin{align*}
x(t) &= \text{Santa’s position at time } t \text{ on Christmas Eve} \\
\frac{\partial x(t)}{\partial t} &= v(t), \text{ velocity} \\
\frac{\partial v(t)}{\partial t} &= u(t), \text{ acceleration}
\end{align*}
\]

We can sample Santa’s position once every second, producing a sequence of position values \( x_1, x_2, \ldots \). His velocity is also represented by a discrete-time process \( v_1, v_2, \ldots \).
We use the following model for Santa’s dynamics:

\[
\begin{bmatrix}
    x_{n+1} \\
    v_{n+1}
\end{bmatrix} =
\begin{bmatrix}
    1 & \Delta \\
    0 & 1
\end{bmatrix}
\begin{bmatrix}
    x_n \\
    v_n
\end{bmatrix} +
\begin{bmatrix}
    0 \\
    \sigma^2
\end{bmatrix} u_n,
\]

\[u_n \sim \mathcal{N}(0, 1), \quad \Delta \text{ small}\]

Also, Santa’s initial position is the North Pole, denoted by \(x_0\). So we take \(p(x_1) = \delta(x_1 - x_0 - \Delta v_0)\). In words, Santa’s position at time \(x_{n+1}\) is equal to his position at time \(n\) plus a small step proportional to his velocity. His velocity is modeled as a Gaussian white noise process, representing the fact that he randomly speeds up and slows down as he makes his stops around the world.
Observation Model (Likelihood)

Usually we cannot observe \( x_n \) directly. Instead we observe \( z_1, z_2, \ldots \), which are noisy and/or indirect measurements related to the states.

**Example: Observation processes**

\[
\begin{align*}
    z_n &= x_n + w_n, \quad w_n \sim \mathcal{N}(0, \sigma^2), \text{ simple signal+noise model} \\
    z_n &= A \begin{bmatrix} x_n \\ x_{n-1} \end{bmatrix} + w_n, \text{ where } A \text{ is a } 2 \times 2 \text{ matrix (e.g. blur)} \\
    z_n &= f(x_n) + w_n, \text{ } f \text{ is a non-linear function}
\end{align*}
\]
Let $p(z_n|x_n)$ denote the likelihood of $x_n$ based on observation $z_n$. We can combine the likelihoods and the priors $p(x_n|x_{n-1})$ to compute the posterior distribution of $x = (x_1, \ldots, x_n)$ given $z = (z_1, \ldots, z_n)$

$$p(x|z) \propto p(z|x)p(x) = \prod_{i=1}^{n} p(z_n|x_n)p(x_n|x_{n-1}).$$

The posterior can be computed efficiently in an incremental fashion by exploiting Markovian structure of state transitions (prior). This incremental procedure is called $Density Propagation$. 
Density Propagation

Density Propagation is an incremental procedure for efficiently computing $p(x_n|z_1, \ldots, z_n)$. First let’s establish some notation:

**Prior:** ($S$ is for State)

$$S_n(x_n|x_{n-1}) := p(x_n|x_{n-1}), \quad P_1(x_1) = p(x_1)$$

**Likelihood:** ($L$ is for Likelihood)

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

**Posterior:** ($F$ is for Filter)

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

**Prediction:** ($P$ is for Prediction or Prior)

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

$P_n(x_n)$ is the prediction of the value of $x_n$ using only observations up to time $n - 1$, and this will play a key role in the Density Propagation algorithm.
Density Propogation Algorithm

\( n = 1 \): predict \( x_1 \):

\[
x_1 \sim p_1(x_1)
\]

observe \( z_1 \) and compute posterior:

\[
F_1(x_1) = p(x_1 \mid z_1) = \frac{p(z_1 \mid x_1)p(x_1)}{p(z_1)} \propto L_1(z_1 \mid x_1)p_1(x_1)
\]

\( n = 2 \): predict \( x_2 \):

\[
p(x_1, x_2 \mid z_1) = \frac{p(x_1, x_2, z_1)}{p(z_1)} = \frac{p(x_2 \mid x_1, z_1)p(x_1 \mid z_1)p(z_1)}{p(z_1)} = p(x_2 \mid x_1)F_1(x_1) = S_2(x_2 \mid x_1)F_1(x_1)
\]

\[
p(x_2 \mid z_1) = \int S_2(x_2 \mid x_1)F_1(x_1)dx =: P_2(x_2)
\]
\( n = 2 \) (cont.): observe \( z_2 \) and compute posterior:

\[
F_2(x_2) = p(x_2 | z_1, z_2) \\
= \frac{p(x_2, z_1, z_2)}{p(z_1, z_2)} \\
= \frac{p(z_2 | x_2)p(x_2 | z_1)p(z_1)}{p(z_1, z_2)} \\
\propto L_2(z_2 | x_2)P_2(x_2)
\]
at time step $n$: predict $x_n$:

\[
P_n(x_n) = p(x_n | z_1, \ldots, z_{n-1})
\]

\[
= \int S_n(x_n | x_{n-1}) F_{n-1}(x_{n-1}) dx_{n-1}
\]

observe $z_n$ and compute posterior:

\[
F_n(x_n) = p(x_n | z_1, \ldots, z_n)
\]

\[
\propto L_n(z_n | x_n) P_n(x_n)
\]
Block Diagram

Block diagram of dynamic filtering.
Filtering

\[ F_n(x_n) \propto L_n(z_n|x_n)P_n(x_n) \]
Prediction

\[ P_{n+1}(x_{n+1}) = \int S_n(x_{n+1} | x_n) F_n(x_n) \, dx_n \]
Estimating \( x_n \)

We have many possibilities. Given

\[
F_n(x_n) = p(x_n | z_1, \ldots, z_n)
\]

we can minimize various risk functions based on a loss and the posterior distribution \( F_n \).

\( \ell_2 \):

\[
\hat{x}_n = \arg \min_{\tilde{x}} \mathbb{E}_{F_n}[(x_n - \tilde{x})^2]
\]

\[
= \int x_n F_n(x_n) \, dx_n
\]

\( \ell_1 \):

\[
\hat{x}_n = \arg \min_{\tilde{x}} \mathbb{E}_{F_n}[|x_n - \tilde{x}|]
\]

\( \ell_{0/1} \):

\[
\hat{x}_n = \arg \max_x F_n(x_n)
\]
Gauss-Markov Model

Definition: Gauss-Markov Model

A random process of the form

\[ x_n = Ax_{n-1} + Bu_n \]

where \( A \in \mathbb{R}^{p \times p} \) and \( B \in \mathbb{R}^{p \times q} \), \( u_n \) is a sequence of uncorrelated, independent jointly Gaussian vectors with \( \mathbb{E}[u_n] = 0 \), and the initial state \( x_0 \sim \mathcal{N}(\mu_x, C_x) \).

A simple model for \( x_n \) which allows us to specify the correlation between samples is the first-order Gauss-Markov process model:

\[ x_n = ax_{n-1} + u_n, \quad n = 1, 2, \cdots \]

\[ u_n \sim \mathcal{N}(0, \sigma_u^2) \] (White Gaussian noise process)
To initialize the process we take $x_0$ to be the realization of a Gaussian random variable:

$$x_0 \sim \mathcal{N}(0, \sigma_x^2)$$

$u_n$ is called the **driving** or **excitation** noise. The model

$$x_n = ax_{n-1} + u_n$$

is called the **dynamical** or **state** model. The current output $x_n$ depends only on the **state** of the system at the previous time, or $x_{n-1}$, and the current input $u_n$.

\[
\begin{align*}
  x_1 &= ax_0 + u_0 \\
  x_2 &= ax_1 + u_1 = a(ax_0 + u_0) + u_1 \\
  &= a^2x_0 + au_1 + u_2 \\
  &\vdots \\
  x_n &= a^{n+1}x_0 + \sum_{k=1}^{n} a^k u_{n-k}
\end{align*}
\]
What can we say about the statistics of $x_n$?

First note

$$\mathbb{E} [x_n] = a^{n+1} \mathbb{E} [x_0] + \sum_{k=1}^{n} a^k \mathbb{E} [u_{n-k}] =$$

The correlation can be computed by noting

$$\mathbb{E} [x_m x_n] = \mathbb{E} \left[ \left( a^{m+1} x_0 + \sum_{k=1}^{m} a^k u_{m-k} \right) \times \left( a^{n+1} x_0 + \sum_{l=1}^{n} a^l u_{m-l} \right) \right]$$

$$\mathbb{E} [x_m x_n] = \mathbb{E} [a^{m+n+2} x_0^2] + \mathbb{E} \left[ \sum_{k=1}^{m} \sum_{l=1}^{n} a^{k+l} u_{m-k} u_{n-l} \right]$$

$$\mathbb{E} [u_{m-k} u_{n-l}] =$$
If $m > n$, then

$$\mathbb{E}[x_m x_n] = a^{m+n+2} \sigma_x^2 + a^{m-n} \sigma_u^2 \sum_{k=1}^{n} a^{2k}$$

If $|a| > 1$, then it’s obvious that the process diverges ($\text{variance} \to \infty$). So, let’s assume $|a| < 1$ and hence a stable system. Thus as $m$ and $n$ get large

$$a^{m+n+2} \sigma_x^2 \to 0$$

Now let $m - n = \tau$. Then for $m$ and $n$ large we have

$$\mathbb{E}[x_m x_n] = a^\tau \sigma_u^2 \sum_{k=1}^{n} a^{2k} = \frac{a^{\tau+2} \sigma_u^2}{1 - a^2}$$

This shows us how correlated the process is.

$$|a| \to 1 \implies \text{heavily correlated (or anticorrelated)}$$

$$|a| \to 0 \implies \text{weakly correlated}$$
Let's look at a more general formulation of the problem at hand. Suppose that we have a vector-valued dynamical equation

\[ x_{n+1} = A x_n + B u_n \]

where

- \( x_n \) is \( p \times 1 \)
- \( x_0 \sim \mathcal{N}(\mu_0, C_0) \)
- \( A \) is \( p \times p \)
- \( B \) is \( p \times q \)
- \( u_n \) is \( q \times 1 \)
- \( u_n \sim \mathcal{N}(0, \sigma_u^2) \) iid (white Gaussian excitation)

This reduces to the case we just looked at when \( p = q = 1 \).
\(p\)-th order Gauss-Markov processes

\[ x_{n+1} = a_1x_n + a_2x_{n-1} + \cdots + a_p x_{n-p+1} + u_n \]

Define

\[ x_n = \begin{bmatrix} x_{n-p+1} \\ x_{n-p+2} \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} \]

Then

\[ x_{n+1} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ a_1 & a_2 & \cdots & a_{p-1} & a_p \end{bmatrix} x_n + \begin{bmatrix} x_{n-p+1} \\ x_{n-p+2} \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u_n \]

\(A = \text{state transition matrix}\)
Since \( x_n \) is a linear combination of Gaussian vectors:

\[
x_n = A^n x_0 + \sum_{k=1}^{n} A^{k-1} B u_{n-k}
\]

we know that \( x_n \) is also Gaussian distributed with mean \( \mu \) and covariance \( C_n = \mathbb{E}[x_n x_n^T] \):

\[
x_n \sim \mathcal{N}(\mu, C_n)
\]

The covariance can be recursively computed from the basic state equation:

\[
C_{n+1} =
\]