Stochastic Filters ECE 830, Spring 2017

Rebecca Willett

Dynamic Filtering

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



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 denote the 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.

Defintion: 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.** $p(x_n|x_{n-1}), n = 2, 3, ...,$ 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.

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 .

Example: (cont.)

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

$$\begin{array}{lll} z_n &=& x_n + w_n \ , \ w_n \sim \mathcal{N}(0,\sigma^2) \ , \ \text{simple signal+noise model} \\ z_n &=& A \left[\begin{array}{c} x_n \\ x_{n-1} \end{array} \right] + w_n \ , \ \text{where} \ A \ \text{is a} \ 2 \times 2 \ \text{matrix} \ \text{(e.g. blur)} \\ z_n &=& f(x_n) + w_n \ , \ f \ \text{is a non-linear function} \end{array}$$

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 $z = (z_1, \ldots, z_n)$

$$p(x|z) \propto p(z|x)p(x) = \prod_{i=1}^{n} p(z_i|x_i)p(x_i|x_{i-1})$$

The posterior can be computed efficiently in an incremental fasion 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)

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

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

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

 $P_n(x_n)$ 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:

$$F_1(x_1) = p(x_1|z_1) = \frac{p(z_1|x_1)p(x_1)}{p(z_1)} \propto L_1(z_1|x_1)p_1(x_1)$$

n = 2: predict x_2 :

$$p(x_1, x_2|z_1) = \frac{p(x_1, x_2, z_1)}{p(z_1)}$$

$$= \frac{p(x_2|x_1, z_1)p(x_1|z_1)p(z_1)}{p(z_1)}$$

$$= p(x_2|x_1)F_1(x_1)$$

$$= S_2(x_2|x_1)F_1(x_1)$$

$$p(x_2|z_1) = \int S_2(x_2|x_1)F_1(x_1)dx_1$$

$$=: P_2(x_2)$$

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 :

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

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

observe z_n and compute posterior:

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

$$\propto L_n(z_n|x_n)P_n(x_n)$$

Block Diagram



Figure: Block diagram of dynamic filtering.

Filtering

$F_n(x_n) \propto L_n(z_n|x_n)P_n(x_n)$



14 / 44

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

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

we can minimize various risk functions based on a loss and the posterior distribution ${\cal F}_n.$

 ℓ_2 :

$$\widehat{x}_n = \arg \min_{\widetilde{x}} \mathbb{E}_{F_n} [(x_n - \widetilde{x})^2]$$
$$= \int x_n F_n(x_n) dx_n$$

 ℓ_1 :

$$\widehat{x}_n = \arg\min_{\widetilde{x}} \mathbb{E}_{F_n}[|x_n - \widetilde{x}|]$$

 $\ell_{0/1}$:

$$\widehat{x}_n = \arg\max_x F_n(x_n)$$

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:

$$x_n = ax_{n-1} + u_n, \ n = 1, 2, \cdots$$

 $u_n \sim \mathcal{N}(0, \sigma_u^2)$ (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 .

$$x_{1} = ax_{0} + u_{0}$$

$$x_{2} = ax_{1} + u_{1} = a(ax_{0} + u_{0}) + u_{1}$$

$$= a^{2}x_{0} + au_{0} + u_{1}$$

$$\vdots$$

$$x_{n} = a^{n}x_{0} + \sum_{k=0}^{n-1} a^{k}u_{n-k}$$

What can we say about the statistics of x_n ?

First note

$$\mathbb{E}[x_n] = a^n \mathbb{E}[x_0] + \sum_{k=0}^{n-1} a^k \mathbb{E}[u_{n-k}] = 0.$$

The correlation can be computed by noting

$$\begin{split} \mathbb{E}\left[x_m x_n\right] &= \mathbb{E}\left[\left(a^m x_0 + \sum_{k=0}^{m-1} a^k u_{m-k}\right) \times \left(a^n x_0 + \sum_{l=0}^{n-1} a^l u_{m-l}\right)\right] \\ \mathbb{E}\left[x_m x_n\right] &= \mathbb{E}\left[a^{m+n} x_0^2\right] + \mathbb{E}\left[\sum_{k=0}^{m-1} \sum_{l=0}^{n-1} a^{k+l} u_{m-k} u_{n-l}\right] \\ \mathbb{E}\left[u_{m-k} u_{n-l}\right] &= \begin{cases} \sigma_u^2, & \text{if } m-k=n-l\\ 0, & \text{otherwise} \end{cases} \end{split}$$

If m > n, then

$$\mathbb{E}[x_m x_n] = a^{m+n} \sigma_x^2 + a^{m-n} \sigma_u^2 \sum_{k=0}^{n-1} a^{2k}$$

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

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

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

$$\mathbb{E}[x_m x_n] = a^{\tau} \sigma_u^2 \sum_{\substack{k=0\\\frac{1-a^{2n}}{1-a^2}}}^{n-1} a^{2k} = \frac{a^{\tau} (1-a^{2n}) \sigma_u^2}{1-a^2}$$

This shows us how correlated the process is.

$$|a| \rightarrow 1 \implies$$
 heavily correlated (or anticorrelated)
 $|a| \rightarrow 0 \implies$ weakly correlated

Vector case

Let's look at a more general formulation of the problem at hand. Suppose that we have a vector-valued dynamical equation

$$\boldsymbol{x}_{n+1} = A\boldsymbol{x}_n + B\boldsymbol{u}_n$$

where

This reduces to the case we just looked at when p = q = 1.

p-th order Gauss-Markov processes

$$x_{n+1} = a_1 x_n + a_2 x_{n-1} + \dots + a_p x_{n-p+1} + u_n$$

Define

$$\boldsymbol{x}_{n} = \begin{bmatrix} x_{n-p+1} \\ x_{n-p+2} \\ \vdots \\ x_{n-1} \\ x_{n} \end{bmatrix}$$

Then

$$\boldsymbol{x}_{n+1} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ a_1 & a_2 & \cdots & a_{p-1} & a_p \end{bmatrix}}_{A = \text{state transition matrix}} \underbrace{\begin{bmatrix} x_{n-p+1} \\ x_{n-p+2} \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}}_{\boldsymbol{x}_n} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}}_{B} u_n$$

Since x_n is a linear combination of Gaussian vectors:

$$\boldsymbol{x}_n = A^n \boldsymbol{x}_0 + \sum_{k=1}^n A^{k-1} B \boldsymbol{u}_{n-k}$$

we know that x_n is also Gaussian distributed with mean **0** and covariance $C_n = \mathbb{E}[x_n x_n^T]$:

$$\boldsymbol{x}_n \sim \mathcal{N}(0, C_n)$$

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

$$C_{n+1} = AC_nA^T + BB^T$$

Density Propagation

The density propagation algorithm discussed earlier 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:

$$P_1(x_1) := p(x_1)$$

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

prior probability on initial x_1 1st order Markov model

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 distirbutions, which lead to very simple linear-algebraic operations.

Kalman filter setup

Observation Model:

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

where $C \in \mathbb{R}^{p \times d}$, $D \in \mathbb{R}^{p \times \ell}$ $z_n \in \mathbb{R}^p$, and $x_n \in \mathbb{R}^d$. **Prior:**

$$x_n = Ax_{n-1} + Bu_n$$

with

$$u_n \stackrel{iid}{\sim} \mathcal{N}(0, I_k) \text{ and } x_1 \sim \mathcal{N}(\mu_{P_1}, V_{P_1})$$

where $A \in \mathbb{R}^{d \times d}$ and $B \in \mathbb{R}^{p \times k}$

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, so it follows that $F_n(x_n)$ and $P_n(x_n)$ are also Gaussian (since products and convolutions of Gaussians are also Gaussian). 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.

Example:

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.

Example: (cont.)

Suppose

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

The posterior is then

$$F_n(x_n) \propto P_n(x_n) L_n(z_n|x_n) \propto e^{-\frac{x_n^2}{2} - \frac{(x_n - z_n)^2}{4}}$$

$$= e^{-\left(\frac{2x_n^2 + x_n^2 - 2x_n z_n + z_n^2}{4}\right)} = e^{-\left(\frac{3x_n^2 - 2x_n z_n + z_n^2}{4}\right)}$$

$$= e^{-\left(\frac{\left(\sqrt{3}x_n - \frac{z_n}{\sqrt{3}}\right)^2 + const}{4}\right)} \propto e^{-\left(\frac{\left(\sqrt{3}x_n - \frac{z_n}{\sqrt{3}}\right)^2}{4}\right)}$$

$$= e^{-\left(\frac{(x_n - \frac{z_n}{3})^2}{4}\right)}$$

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

Kalman Filter (special case of density propogation)

Start with prediction density $P_n(x_n)$.

1. Correction/Update: After observing z_n compute $F_n(x_n)$:

$$K_n := V_{P_n} C^T (C V_{P_n} C^T + D^T D)^{-1}$$
$$\mu_{F_n} = \mu_{P_n} + K_n (z_n - C \mu_{P_n})$$
$$V_{F_n} = (I - K_n C) V_{P_n}$$

2. Estimation: Estimate x_n using BMMSE estimation:

$$\widehat{x}_n = \mathbb{E}[x_n | z^n] = \mu_{F_n}.$$

3. Prediction: Compute the updated prediction $P_{n+1}(x_{n+1})$:

$$\mu_{P_{n+1}} = A\mu_{F_n}$$
$$V_{P_{n+1}} = AV_{F_n}A^T + B^TB$$

A word on notation

The notation used in these notes is a little different from that used in many texts. The key differences are described here:

	These notes	"Standard"
Variables:	μ_{F_n}	$\widehat{s}_{n n}$
	V_{F_n}	$M_{n n}$
	μ_{Pn}	$\widehat{s}_{n n-1}$
	V_{P_n}	$M_{n\mid n-1}$
Prediction:	$\mu_{P_{n+1}} = A\mu_{F_n}$	$\hat{s}_{n\mid n-1} = A\hat{s}_{n-1\mid n-1}$
	$V_{P_{n+1}} = A V_{F_n} A^T + B^T B$	$M_{n n-1} = AM_{n-1 n-1} + B^T B$
Kalman gain:	$K_n := V_{P_n} C^T (C V_{P_n} C^T + D^T D)^{-1}$	$K_n = M_{n n-1} H^T (D^T D + C M_{n n-1} C^T)^{-1}$
Filtering:	$\mu_{F_n} = \mu_{P_n} + K_n (z_n - C \mu_{P_n})$	$\hat{s}_{n n} = \hat{s}_{n n-1} + K_n(\boldsymbol{x}_n - C\hat{s}_{n n-1})$
	$V_{F_n} = (I - K_n C) V_{P_n}$	$M_{n n} = (I - K_n H)M_{n n-1}$

Example: First-order Gauss-Markov process

Setup:

$$\begin{split} s_0 &\sim \mathcal{N}(0, \sigma_s^2) \\ u_n &\sim \mathcal{N}(0, \sigma_u^2) \\ s_{n+1} &= as_n + u_n \\ \sigma_s &= \sigma_u = 0.1 \\ a &= 0.99 \Rightarrow \text{ highly correlated process} \end{split}$$

$$(A = a , B = \sigma_u)$$

Measurements/observation model:

$$x_n = s_n + w_n$$
$$w_n \sim \mathcal{N}(0, \sigma_w^2)$$
$$\sigma_w = 0.5$$

 $(C=1, D=\sigma_w^2)$

Example: (cont.)

- 1. Initialize: Start with initial prediction density $P_0(s_0) \sim \mathcal{N}(0, 0.1)$. I.e. $\mu_{P_0} = 0$, $V_{P_0} = 0.1$
- **2.** Correction/Update: Observe x_n , then compute

$$K_n = \frac{V_{P_{n-1}}}{V_{P_{n-1}} + \sigma_w^2}$$
$$\mu_{F_n} = \mu_{P_{n-1}} + K_n(z_n - \mu_{P_{n-1}})$$
$$V_{F_n} = (1 - K_n)V_{P_{n-1}}$$

- **3.** Minimum MSE Estimation: $\hat{x}_n = \mu_{F_n}$
- **4.** Prediction: Compute the updated prediction $P_n(s_n)$:

$$\mu_{P_n} = a\mu_{F_{n-1}}$$
$$V_{P_n} = a^2 V_{F_{n-1}} + \sigma_u^2$$

Code

```
Kalman.m
                Sun Apr 10 21:14:56 2011
                                                   1
clear all
N = 1000;
sigu = 0.1;
sigw = 0.5;
a = 0.99;
s0 = randn(1)*sigu;
mu P = 0;
CP = siqu;
s = zeros(N, 1);
s(1) = s0;
s hat = zeros(N,1);
x = zeros(N, 1);
for n = 1:N
  x(n) = s(n) + randn(1) * sigw;
  K = C P/(C P + sigw^2);
  mu F = mu P + K^*(x(n) - mu P);
  s hat(n) = mu F;
  CF = (1-K)*CP;
  mu P = a * mu F;
  C \overline{P} = a^{2*C} \overline{F} + sigu^{2};
  if n < N
    s(n+1) = a*s(n) + randn(1)*sigu;
  end
end
figure(1):
subplot(311); plot(s);xlabel('n');vlabel('s')
subplot(312); plot(x);xlabel('n');ylabel('x')
subplot(313); plot(s hat):xlabel('n'):vlabel('\hat{s}')
```

Output



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:

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

where ϕ, ψ nonlinear **Linearization**:

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

$$x_n \approx \underbrace{\phi(\mu_{F_n}) + \nabla_{\phi}(\mu_{F_n})(x_n - \mu_{F_n})}_{\bullet} + Bu_n$$

Taylor series approximation of $\phi(x_n)$ at point μ_{F_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).

Example Particles for $P_n(x_n)$.



More samples in regions of higher density provide very efficient approximations

Condensation Alogorithm (Isard & Blake 1998)

Start with samples from $P_n(x_n)$



Resample accodrding 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.
- The choice of resampling method is critical. Poor choices lead to "particle degeneracy", where all the particles except on have weights close to zero.

Example: Orange Tracker (thanks to Rui Castro!)

Prior:

$$\begin{aligned} x_n &= \begin{bmatrix} P_n \\ V_n \end{bmatrix} \text{ position } \text{ velocity } \text{ time } n \text{ (}4 \times 1 \text{ state)} \\ x_n &= Ax_n + B\mu_n \\ &= \begin{bmatrix} I & \Delta \\ 0 & I \end{bmatrix} \begin{bmatrix} P_n \\ V_n \end{bmatrix} + \begin{bmatrix} 0 \\ \sigma^2 \end{bmatrix} \mu_n \text{ , } \mu_n \overset{iid}{\sim} \mathcal{N}(0, 1) \end{aligned}$$

Observation:

$$z_n = \psi(x_n)$$

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, move out of the scene, or be confused with the other orange.

Original

Tracking results