Intro to the Kalman filter

ISTA 410 / INFO 510: Bayesian Modeling and Inference

U. of Arizona School of Information

December 2, 2020

Outline

Last time:

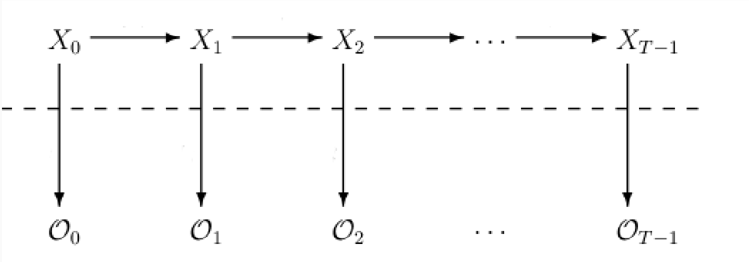
- Filtering, smoothing, and fitting in HMMs

Today:

- Filtering for linear Gaussian dynamical systems

Linear dynamical systems and the Kalman filter

Type of system to estimate



Difference from last time:

- Arrows are linear transformations with noise added.

Type of system to estimate

Basic system structure:

- Sequence of *states* x_k evolves according to a linear difference equation:

  $f(x_{k-1}) \leftarrow$ nonlinear

  $$x_k = Ax_{k-1} + w_k$$

  where w_k is a random noise term
- Sequence of observations z_k is a linear function of the state:

  $$z_k = Hx_k + v_k$$

  again, v_k is a noise term $h(x_k) \leftarrow$ nonlinear

Assumption: $w_k \sim \mathrm{Normal}(0, Q)$, $v_k \sim \mathrm{Normal}(0, R)$.

Goal: given the sequence of z_k, estimate the states x_k, along with error covariance estimates P_k.

A Bayesian estimation step

The Kalman filter equations are based on an iterative application of Bayes' theorem. If we assume that we have, by magic, a prior estimate of the kth state, $\hat{x}_k^-$ and this estimate's error covariance P_k^-, then, the distribution of the true x_k is:

$$x_k \sim \mathrm{Normal}(\hat{x}_k^-, P_k^-)$$

Conditional on x_k, the distribution of the measurement z_k is

$$z_k | x_k \sim \mathrm{Normal}(H x_k, R)$$

Finally, the marginal distribution of the measurement z_k is

$$z_k \sim \mathrm{Normal}(H \hat{x}_k^-, H P_k^- H^T + R)$$

A Bayesian estimation step

Plugging everything into Bayes' theorem, we get the posterior density for x_k.

$$p(x_k|z_k) = \frac{N(\hat{x}_k^-, P_k^-)N(Hx_k, R)}{N(H\hat{x}_k^-, HP_k^-H^T + R)}$$

Then, through a bunch of algebra, one can find that the mean of x_k is:

$$\hat{x}_k^- + (P_k^- H^T (HP_k^- H + R)^{-1}) \overbrace{(z_k - H\hat{x}_k^-)}^{\text{residual}}$$

The highlighted factor is often denoted K_k, the *Kalman gain* or *blending factor.*

We also have the posterior covariance:

$$P_k = (I - K_k H)P_k^-$$

Interpretation

So, the posterior mean for the kth state is

$$\hat{x}_k^- + (P_k^- H^T (H P_k^- H^T + R)^{-1})(z_k - H\hat{x}_k^-)$$

which is our prior estimate plus a correction based on $(z_k - H\hat{x}_k^-)$, the residual (disagreement between observation z_k and predicted observation $H\hat{x}_k^-$).

Kalman gain determines the weight of each part.

This is best understood in two separate limits.

Role of the Kalman gain

If our measurements are really good, we should trust the observed values.

why if H isn't invertible?

$$P_k^- H^T (HP_k^- H^T + \boxed{R})^{-1} \to \boxed{H^{-1}} \text{ as } R \to 0$$

Then, in the limit of perfect measurement, the posterior mean is

$$\hat{x}_k^- + (P_k^- H^T (HP_k^- H^T + R)^{-1})(z_k - H\hat{x}_k^-) = H^{-1} z_k$$

(Remember H is the linear transformation mapping x_k to z_k.)

$$x \xrightarrow{H} z$$

H^{-1} doesn't exist → multiple x's give same z

If H^{-1} exists, the model is <u>observable</u>

Role of the Kalman gain

The "opposite" limit occurs when the prior estimate's error covariance goes to 0:

$$P_k^- H^T (H P_k^- H^T + R)^{-1} \to 0 \text{ as } P_k^- \to 0$$

in which case the posterior mean becomes

$$\hat{x}_k^- + (P_k^- H^T (H P_k^- H^T + R)^{-1})(z_k - H\hat{x}_k^-) = \hat{x}_k^-$$

The filtering algorithm

This assumed we had a prior estimate for the state and error covariance.

Where do we get these?

The filtering algorithm

This assumed we had a prior estimate for the state and error covariance.

Where do we get these?

From the previous time step! If we have estimates $\hat{x}_{k-1}, P_{k-1}$, the dynamical system gives:

$$\left.\hat{x}_k^- = A\hat{x}_{k-1}\right\} f(\hat{x}_{k-1})$$

$$P_k^- = AP_{k-1}A^T + Q$$

So this is a Bayes' update step where the dynamical system transforms the posterior from step $k-1$ into the prior for step k.

The filtering algorithm

So, after setting initial estimates for $\hat{x}_0^-, P_0^-$, the filtering algorithm proceeds in two steps:

1. Obtain prior estimates $\hat{x}_k^-, P_k^-$ by applying the dynamical system to $\hat{x}_{k-1}, P_{k-1}$.
2. Compute the Kalman gain K_k and adjust estimates to $\hat{x}_k, P_k$.

Notes:

- We're taking advantage of normality and linearity here; all conditional and marginal distributions remain normal, so we can work only in terms of the mean/covariance.
- If Q, R are constant, then the estimate error covariance P_k and the gain K_k stabilize quickly and then stay constant.

Parameters and tuning

A couple of comments on choosing the parameters Q, R:

- Measuring R empirically is usually practical, because it's a property of our measurement
- Q is trickier. Can come from a scientific model (ideally). Can "compensate" for a poor process model by adding more uncertainty to state estimates.

Initial values of $\hat{x}_0, P_0$ less important if the filter will run for some time.

Simplest example

Simplest possible example: estimating a constant

A very simple example comes down to estimation of an unknown constant.

For example: we are trying to estimate a voltage, but our instruments are faulty, introducing an amount of noise to each measurement.

This implies the following parameters:

- $A = 1$ (no deterministic time evolution of states)
- $H = 1$ (measuring voltage directly)
- $Q \approx 0$ (assume negligible fluctuation in states)
- $R = R_0$ (fixed measurement error)

Kalman filter equations

We can write down the Kalman filter equations for this version easily:

Dynamics update step:

$$\hat{x}_k^- = \hat{x}_{k-1} \quad \leftarrow \text{estimated state}$$

$$P_k^- = P_{k-1} + Q \quad \leftarrow \text{covariance}$$

We can in principle set $Q = 0$, but we can also adjust it to allow for some fluctuations in the true voltage.

Kalman filter equations

The correction step:

$$K_k = \frac{P_k^-}{P_k^- + R}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - \hat{x}_k^-)$$

$$P_k = (1 - K_k)P_k^-$$

Initial parameters

We need to pick values for:

- Q
- $\hat{x}_0$
- P_0
- R (presuming this might be unknown)

Initial parameters

We need to pick values for:

- Q: we'll use 10^{-5} for something very small but nonzero.
- $\hat{x}_0$:
- P_0:
- R: (presuming this might be unknown)

Initial parameters

We need to pick values for:

- Q: we'll use 10^{-5} for something very small but nonzero.
- $\hat{x}_0$: we'll start with 0
- P_0:
- R: (presuming this might be unknown)

Initial parameters

We need to pick values for:

- Q: we'll use 10^{-5} for something very small but nonzero.
- $\hat{x}_0$: we'll start with 0
- P_0: this one determines how quickly we converge to a stable estimate – we'll try a few examples
- R: (presuming this might be unknown)

Initial parameters

We need to pick values for:

- Q: we'll use 10^{-5} for something very small but nonzero.
- $\hat{x}_0$: we'll start with 0
- P_0: this one determines how quickly we converge to a stable estimate – we'll try a few examples
- R: this determines how much we "trust" the noisy measurements – we'll try a few examples

A slightly less trivial example

Kalman filter with control

Kalman initially developed the filter for applications in control theory. So, alternate form:

$$x_k = Ax_{k-1} + \boxed{Bu_{k-1}} + \boxed{w_k}$$

normal noise

(states)

$$z_k = Hx_k + \boxed{v_k}$$

noise

(measurements)

- u_{k-1}, the control input, represents some linear forcing we can do to the system
- B defines how the control input influences the system

Modifying the filter equations

It turns out the only modification we need to make is to the prediction step:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$
$$P_k^- = AP_{k-1}A^T + Q$$

(correction step remains the same)

A toy example: position tracking for a robot

Imagine we're trying to track the position of a robot that moves through the following idealized physical environment:

- 2 dimensions (x/y or lat/long)
- No friction
- Fluctuations/turbulence $\rightarrow$ white noise added to velocity

Our robot has four thrusters that allow us to apply a constant force in any of four directions

Suppose we also get periodic measurements of x/y position (quite noisy) and velocity (not so noisy)

Setting up the dynamics

We need four variables:

- x, y – position coordinates
- $\dot{x}, \dot{y}$ – velocities

We'll use discrete time with a time step Δt.

Setting up the dynamics

If we assume no control, what is the dynamical relationship?

$$\vec{x}_{k+1} = Ax_k$$

with

$$\vec{x}_k = \begin{pmatrix} x_k \\ \dot{x}_k \\ y_k \\ \dot{y}_k \end{pmatrix} \qquad A = \begin{pmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Setting up the dynamics

If we assume no control, what is the dynamical relationship?

$$x_{k+1} = Ax_k$$

with

$$A = \begin{pmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Adding control

We'll assume our control vectors look like

$$u = \begin{pmatrix} T_x \\ T_x \\ T_y \\ T_y \end{pmatrix}$$

where $T_x, T_y \in \{-1, 0, 1\}$ give the thruster state (+/- thrust or off), and that the thrusters produce a constant acceleration.

Adding control

This gives rise to the B matrix:

$$B = \begin{pmatrix} \frac{1}{2}k\Delta t^2 & 0 & 0 & 0 \\ 0 & k\Delta t & 0 & 0 \\ 0 & 0 & \frac{1}{2}k\Delta t^2 & 0 \\ 0 & 0 & 0 & k\Delta t \end{pmatrix}$$

k = acceleration produced by my thruster

Mechanics: change in position from applying accel k for Δt = $\frac{1}{2}k\Delta t^2$

change in velocity: $k\Delta t$

Adding control

This gives rise to the B matrix:

$$B = \begin{pmatrix} \frac{1}{2}k\Delta t^2 & 0 & 0 & 0 \\ 0 & k\Delta t & 0 & 0 \\ 0 & 0 & \frac{1}{2}k\Delta t^2 & 0 \\ 0 & 0 & 0 & k\Delta t \end{pmatrix}$$

This gives us enough to implement the model.

Summary

Today:

- Simple discrete-time Kalman filter

Next week: nonlinear filtering

- Extended Kalman filter (for nonlinear dynamics)
- Unscented Kalman filter / particle filter / etc.