Lecture 22
Kalman Filter

CS 520: Intro AI
Jingjin Yu | Rutgers
Kalman Filter: the Problem

Problem: estimate the state of a system \( \dot{x} = f(x, u) \) using noisy observations taken at discrete time steps, i.e.,

- System is at some state \( x_0 \), with initial control input \( u_0 \)
- After some \( \Delta_1 \) time, system is at a new state \( x_1 \)
- A noisy observation \( z_1 \) of \( x_1 \) is made
- ...
- System is at state \( x_t \) and we give it a control input \( u_t \)
- After some \( \Delta_t \) time, system is at a new state \( x_{t+1} \)
- A noisy observation \( z_{t+1} \) of \( x_{t+1} \) is made

Question: estimate the distribution of system state at \( t + 1 \) given observations \( z_1, \ldots, z_{t+1} \), written as \( P(X_{t+1}|z_{1:t+1}) \)

To simplify things a little, we do not consider control \( u_{1:t} \)

The system is at some state \( x \), but we are interested in the distribution \( X \)
Kalman Filter

⇒ **Filtering**: estimating state variables (e.g., position, velocity, acceleration, ...) from noisy observations over time.

⇒ **Kalman filter** is a **continuous** time **optimal** filter for **linear** systems with **Gaussian** noise

  ⇒ Continuous time: system variables, $x$, is continuous
    ⇒ E.g., position/velocity of a rocket, or temperature of a room

  ⇒ Linear system with Gaussian noises $(\omega_k, \nu_k)$
    ⇒ Motion: $x_{k+1} = F_k x_k + B_k u_k + \omega_k$
    ⇒ Observation: $z_k = H_k x_k + \nu_k$

⇒ Why Kalman filter?

  ⇒ Numerous applications, including
    ⇒ GPS navigation
    ⇒ Tracking of aircraft, missiles, submarines
  ⇒ Works well even if the linear assumption does not hold

Source: wikipedia
Application Scenario: GPS

How does GPS work?

- 20+ GPS satellites with known locations
- Synchronized clock
  - Actually uses (both special and general) theory of relativity
- Need at least four satellites to work
- A form of trilateration
- Lots of possible sources of disturbances
  - Clock
  - Atmosphere interference
  - Satellite locations
  - Multipath
  - ...

Without Kalman filtering, error can be tens of meters and jump around

- Can be highly problematic for GPS based navigation

Source: space.com
Bayesian Network Perspective

We may view the process as a Bayesian network

\[ X_0 \rightarrow X_1 \rightarrow \cdots \rightarrow X_t \rightarrow X_{t+1} \]

\[ Z_1 \rightarrow \cdots \rightarrow Z_t \rightarrow Z_{t+1} \]

\( X_t \) is the position of the system at time \( t \)

\( Z_t \) is the observation made at time \( t \) about \( X_t \)

Both are assumed to be linear with Gaussian noise

- Motion: \( x_{k+1} = F_k x_k + B_k u_k + \omega_k \)
- Observation: \( z_{k+1} = H_{k+1} x_{k+1} + \nu_{k+1} \)

We want to compute \( P(X_{t+1}|z_{1:t+1}) \)

Given \( X_t, X_{t+1} \) does not depend on \( X_{1:t-1} \) and \( z_{1:t} \) (Markov blanket)
Time Update

Kalman filtering has two iterative steps

\[ P(X_{t+1}|z_{1:t}) \] from \[ P(X_t|z_{1:t}) \]

\[ P(X_{t+1}|z_{1:t+1}) \] from \[ P(X_{t+1}|z_{1:t}) \] and \[ P(z_{t+1}|X_{t+1}) \]

Time update

\[
P(X_{t+1}|z_{1:t}) = \int_{x_t} P(X_{t+1}, x_t|z_{1:t}) dx_t
\]

Marginalization over \( x_t \)

Definition of conditional probability

\[
= \int_{x_t} \frac{P(X_{t+1}, x_t, z_{1:t})}{P(z_{1:t})} dx_t
\]

Chain rule

\[
= \int_{x_t} \frac{P(X_{t+1}, x_t, z_{1:t})}{P(x_t, z_{1:t})} \frac{P(x_t, z_{1:t})}{P(z_{1:t})} dx_t
\]

\[
= \int_{x_t} P(X_{t+1}|x_t, z_{1:t}) P(x_t|z_{1:t}) dx_t
\]

Conditional independence of \( X_{t+1} \) over \( z_{1:t} \) given \( x_t \)

\[
= \int_{x_t} P(X_{t+1}|x_t) P(x_t|z_{1:t}) dx_t
\]

Transition model

\[ x_{k+1} = A_{k+1} x_k + B_{k+1} u_k + \omega_{k+1} \]

\[ P(X_t|z_{1:t}), \text{ from previous computation or } x_0 \]
Measurement Update

\[ P(X_{t+1}|z_{1:t+1}) \text{ from } P(X_{t+1}|z_{1:t}) \text{ and } P(z_{t+1}|X_{t+1}) \]

\[
P(X_{t+1}|z_{1:t+1}) = \frac{P(X_{t+1}, z_{1:t}, z_{t+1})}{P(z_{1:t+1})} \\
= \frac{P(z_{t+1}|X_{t+1}, z_{1:t})P(X_{t+1}, z_{1:t})}{P(z_{1:t+1})} \\
= P(z_{t+1}|X_{t+1}) \frac{P(X_{t+1}, z_{1:t})}{P(z_{1:t+1})} \\
= P(z_{t+1}|X_{t+1})P(X_{t+1}|z_{1:t}) \frac{P(z_{1:t})}{P(z_{1:t+1})} \\
= \alpha P(z_{t+1}|X_{t+1})P(X_{t+1}|z_{1:t})
\]

Definition of conditional probability

Product rule

Conditional independence of \(z_{t+1}\) over \(z_{1:t}\) given \(X_{t+1}\)

\[
\frac{P(z_{1:t})}{P(z_{1:t+1})} \text{ is a normalization term}
\]

Measurement (sensor) model

Directly from time update
Putting it together

⇒ Two iterative steps

⇒ Time update: \( P(X_{t+1} | z_{1:t}) = \int_{x_t} P(X_{t+1} | x_t) P(x_t | z_{1:t}) dx_t \)

⇒ Measurement update: \( P(X_{t+1} | z_{1:t+1}) = \alpha P(z_{t+1} | X_{t+1}) P(X_{t+1} | z_{1:t}) \)

⇒ Key observation

⇒ \( P(X_{t+1} | z_{1:t}) \) is Gaussian if \( P(X_{t+1} | x_t) \) and \( P(x_t | z_{1:t}) \) are both Gaussian

⇒ \( P(X_{t+1} | z_{1:t+1}) \) is Gaussian when \( P(z_{t+1} | X_{t+1}) \) and \( P(X_{t+1} | z_{1:t}) \) are Gaussian

⇒ This enables Kalman filter to compactly represent the system’s state

⇒ Only need a Gaussian distribution, i.e., \( N(X, \Sigma) \)

⇒ \( X \) is the state vector mean

⇒ \( \Sigma \) is the covariance matrix
A One Dimensional Example (from Textbook)

⇒ Consider 1D random walk, for 1D, $N(X, \Sigma)$ is simply $N(\mu, \sigma^2)$

⇒ $x_0 \sim N(\mu_0, \sigma_0^2)$, i.e., $P(x_0) = \alpha \exp\left(-\frac{1}{2} \left( \frac{(x_0-\mu_0)^2}{\sigma_0^2} \right) \right)$

⇒ Transition model: $P(x_{t+1}|x_t) = \alpha \exp\left(-\frac{1}{2} \left( \frac{(x_{t+1}-x_t)^2}{\sigma_x^2} \right) \right)$

⇒ If $x_{t+1} = x_t + \nu \Delta t$, then $P(x_{t+1}|x_t) = \alpha \exp\left(-\frac{1}{2} \left( \frac{(x_{t+1}-(x_t+\Delta t))}{\sigma_{t+1}^2} \right) \right)$

⇒ Sensor model: $P(z_t|x_t) = \alpha \exp\left(-\frac{1}{2} \left( \frac{(z_t-x_t)^2}{\sigma_z^2} \right) \right)$

⇒ Time update for $x_1$

$$P(x_1) = \int_{-\infty}^{\infty} P(x_1|x_0)P(x_0)dx_0$$

$$= \int_{-\infty}^{\infty} \alpha \exp\left(-\frac{1}{2} \left( \frac{(x_0-\mu_0)^2}{\sigma_0^2} \right) \right) \exp\left(-\frac{1}{2} \left( \frac{(x_1-x_0)^2}{\sigma_x^2} \right) \right) dx_0$$

$$= \alpha \int_{-\infty}^{\infty} \exp\left(-\frac{1}{2} \left( \frac{(x_0-\mu_0)^2}{\sigma_0^2} \right) \right) \exp\left(-\frac{1}{2} \left( \frac{(x_1-x_0)^2}{\sigma_x^2} \right) \right) dx_0$$

$$= \alpha \exp\left(-\frac{1}{2} \left( \frac{(x_1-\mu_0)^2}{\sigma_0^2 + \sigma_x^2} \right) \right)$$

Note: the normalization constant $\alpha$s are different in different expressions!
A One Dimensional Example (from Textbook)

⇒ Measurement update for $x_1$

$$P(x_1|z_1) = \alpha P(z_1|x_1)P(x_1) = \alpha \exp \left( -\frac{1}{2} \left( \frac{(z_1 - x_1)^2}{\sigma_z^2} \right) \right) \exp \left( -\frac{1}{2} \left( \frac{(x_1 - \mu_0)^2}{\sigma_0^2 + \sigma_x^2} \right) \right)$$

$$= \alpha \exp \left( -\frac{1}{2} \left( \frac{(x_1 - \mu_1)^2}{\sigma_1^2} \right) \right)$$

⇒ In general, $x_{t+1} \sim N(\mu_{t+1}, \sigma_{t+1}^2)$ with

$$\mu_{t+1} = \frac{(\sigma_t^2 + \sigma_x^2)z_{t+1} + \sigma_z^2 \mu_0}{\sigma_t^2 + \sigma_x^2 + \sigma_z^2}$$

$$\sigma_{t+1}^2 = \frac{(\sigma_t^2 + \sigma_x^2)\sigma_z^2}{\sigma_t^2 + \sigma_x^2 + \sigma_z^2}$$
A One Dimensional Example (from Textbook)

\[ \mu_0 = 0.0, \sigma_0 = 0 \]
\[ \sigma_x = 2.0 \]
\[ z_i = 2.5, \sigma_z = 1.0 \]
General Case

⇒ In general, \( P(x) = N(\mu, \Sigma)(x) = \alpha \exp \left( -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right) \)

⇒ Transition model: \( P(x_{t+1}|x_t) = N(Fx_t, \Sigma_x)(x_{t+1}) \)
   
   ⇒ \( F \) is a matrix describing the linear transition model
   
   ⇒ \( \Sigma_x \) is the covariance matrix modeling transition noise
   
   ⇒ This models the system \( x_{t+1} = Fx_t + \omega_t \) and ignores control \( u_t \)

⇒ Sensor model: \( P(z_t|x_t) = N(Hx_t, \Sigma_z)(z_t) \)
   
   ⇒ \( H \) is a matrix describing the sensor model
   
   ⇒ \( \Sigma_z \) is the covariance matrix modeling measurement noise

⇒ Update rule: \( x_{t+1} \sim N(\mu_{t+1}, \Sigma_{t+1}) \)
   
   ⇒ \( \mu_{t+1} = F\mu_t + K_{t+1}(z_{t+1} - HF\mu_t) \)
   
   ⇒ \( \Sigma_{t+1} = (I - K_{t+1}H)(FS_tF^T + \Sigma) \)
   
   ⇒ Kalman gain: \( K_{t+1} = (FS_tF^T + \Sigma_x)H^T(H(FS_tF^T + \Sigma_x)H^T + \Sigma_z)^{-1} \)
Filtering versus Smoothing

- Kalman filtering is an “online” process, i.e., $P(x_t | z_{1:t})$
- If we know all the data from 1, ..., $t$, ..., $T$, then we can do better with Kalman smoothing, i.e., we can compute $P(x_t | z_{1:T})$

- With more data, smoothing yields better results
Extensions: Extended Kalman Filter (EKF)

⇒ Kalman filter applies to only linear systems, i.e.,

\[ x_{t+1} = Fx_t + Bu_t + \omega_t, \quad z_{t+1} = H_{t+1}x_{t+1} + v_{t+1} \]

with \( F, B, H \) being matrices with no dependency on \( x \)

⇒ Most systems are non-linear, i.e.,

\[ x_{t+1} = f(x_t, u_t, \omega_t), \quad z_t = h(x_t, \nu_t) \]

⇒ One way to overcome the limitation is to linearize the system

⇒ E.g., using Taylor expansion around \( x_t, u_t, \omega_t \) and take the first order terms

⇒ Apply KF over the linearized transition and measurement models

⇒ This is the Extended Kalman Filter (EKF)
Unscented Kalman Filter (UKF) and Particle Filter

- EKF will not work well if the system is highly non-linear

  - In this case, we simply sample the distribution with weights
  - Then, propagate the resulting samples through the non-linear system
  - UKF generates samples deterministically, PF does so randomly

Source: http://www.inference.phy.cam.ac.uk/tcs27/talks/sampling.html