CS 585 – Image and Video Computing

Lecture on Kalman Filter by Margrit Betke

Computer Science Department, Boston University

February 20, 2024

Kalman Filter

Background

Stochastic Process = sequence of random variables

In CS 585: The sequence we are interested in is the **state** $x(t)$ of an object we would like to track.

Markov Assumption:
$x(t)$ can be described by $x(t-1)$. No need to use $x(t-2)$ or $x(t-3)$, etc.

State Equation:
$x(t)$ is a function of $x(t-1)$ and some noise process q with a known distribution, that is:

$$x(t) = f(x(t-1), q(t-1))$$

Simple distribution for q: Gaussian = $\mathcal{N}$

Simple noise distribution q for a stochastic process: "white noise," which means:

$$E[q(t)] = 0$$

Interpretation of this equation: The sequence is uncorrelated in time. The distribution q is valid at all times, which is also called *stationary*. (As opposed to q changing through time, for example, having an increasing mean or variance.)

So $q(t) = \mathcal{N}(0, \sigma_q^2)$.

Figure 1: Object track x (thick line) and average object movement $A + Bt$ (thin line).

Example:

Assume we take a video of an object that slowly moves through the scene one pixel in the horizontal direction per time step t. In the visualization in Figure 1, we overlaid the object position in each frame of the video onto a single frame. For illustration purposes and to make things easier to explain, we assume the column index and the time index are the same.

The movement of the object can be described by the stochastic process

$$x[t] = A + Bt + q[t] \quad t = 0, 1, \dots, N-1,$$

where $x[t]$ stands for the state of the object at time t, here its vertical position in the video frame. Offset A is the beginning point, B is the slope of the movement, and $q = \mathcal{N}(0, \sigma_q^2)$ is white Gaussian noise. We use brackets $[t]$ instead of parentheses (t) to indicate that the state is discrete.

The average track of the object is described by the deterministic parameters

$$\Theta = (A, B)^T.$$

Assume the object state (vertical position) is $\mathbf{x} = (x[0], x[1], ..., x[N-1])^T$. Then the parameterized probability density function (PDF) that describes the process is

$$p(\mathbf{x}; \Theta) = \frac{1}{(2\pi\sigma_q^2)^{\frac{N}{2}}} \exp(-\frac{1}{2\sigma_q^2} \sum_{t=0}^{N-1} (x[t] - A - Bt)^2).$$

Extending the example to a Kalman Filter scenario:

Now we assume that the state $x[t]$ cannot be directly measured. Instead, we have another sequence of random variables $z[t]$, the measurements. The true vertical positions and the measured vertical positions are the same except for some additive noise.

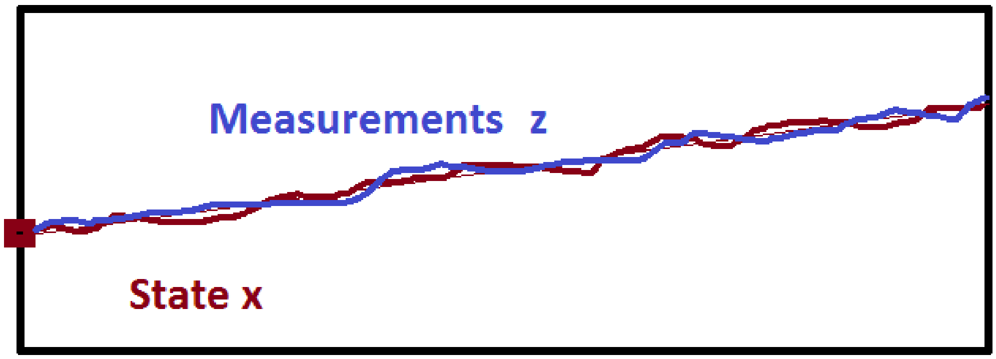

Figure 2: Object state x (red line) and its noisy measurements z.

Equations Describing the 1D Dynamics of an Object to be Tracked:

1D State Equation:

$$x[t] = a\, x[t-1] + u[t],$$

where $u[t]$ is a zero-mean Gaussian white noise process $\mathcal{N}(0, \sigma_q^2)$.

1D Measurement Equation:

$$z[t] = x[t] + w[t],$$

where $w[t]$ is a zero-mean Gaussian white noise process $\mathcal{N}(0, \sigma_w^2)$.

Purpose of a Kalman Filter:

A Kalman filter is used to estimate the true state $x[t]$ based on the measurements $z[0], ..., z[t]$ and the known noise processes $\mathcal{N}(0, \sigma_q^2)$ and $\mathcal{N}(0, \sigma_w^2)$.

Notation:

We use:

"hat" to indicate an estimated state, e.g., $\hat{x}$ means the estimate of state x.

"|" to show what the estimate is conditioned on, for example:

$\hat{x}[t|t-1]$ means that the state x at time t is estimated based on our knowledge up to time $t-1$.

$\hat{x}[t|t]$ means that the state x at time t is estimated based on our knowledge up to time t.

Optimality of Kalman Filter:

The Kalman filter is a Minimum Bayesian Mean Square Error Estimator:

$$E[\,(x[t]-\hat{x}[t|t])^2\,]$$

where the expectation is with respect to $p(z[0], z[1], .., z[t], x[t])$

Correction Term of Kalman Filter:

$$\hat{x}[t|t] = \hat{x}[t|t-1] + K[t]\ \ (z[t] - \hat{x}[t|t-1])$$

$K[t]$ is called the **Kalman gain factor**. It is always < 1. It weighs how much to rely on the measurement $z[t]$ versus the prediction $\hat{x}[t|t-1]$. It has the mean square error estimate based on $t-1$ measurements in numerator and denominator. The denominator also includes the variance of the measurement noise:

$$K[t] = \frac{E[\,(x[t]-\hat{x}[t|t-1])^2\,]}{E[\,(x[t]-\hat{x}[t|t-1])^2\,] + \sigma_w^2}$$

For notation simplicity, we use M to denote the mean square error M: $M[t|t] = E[\,(x[t]-\hat{x}[t|t])^2\,]$.

The Kalman gain can then be written as

$$K[t] = \frac{M[t|t-1]}{M[t|t-1] + \sigma_w^2}$$

The correction term is only one of five update equations that the Kalman filter computes to estimate the object state per iteration.

Why 5 equations?

Per iteration, the Kalman filter has three phases, one to predict, one to compute the Kalman gain, and one to estimate:

1) Prediction phase: Predict current state based on the previous state. Then predict the current mean square error from the previous mean square error.

2) Compute K[t]

3) Estimation phase: Estimate the current state based on the prediction, Kalman gain, and current measurement. Then estimate the current mean square error.

1D Kalman Filter Update Equations:

1. State Prediction:
$$\hat{x}[t|t-1] = a\,\hat{x}[t-1|t-1]$$

2. MSE Prediction:
$$M[t|t-1] = a^2\,M[t-1|t-1] + \sigma_q^2$$

3. Kalman Gain Computation:
$$K[t] = \frac{M[t|t-1]}{M[t|t-1] + \sigma_w^2}$$

4. State Estimation (= Correction):
$$\hat{x}[t|t] = \hat{x}[t|t-1] + K[t]\ (z[t] - \hat{x}[t|t-1])$$

5. MSE Estimation:
$$M[t|t] = (1 - K[t])M[t|t-1]$$

How to initialize?

Assume $x[-1] \sim \mathcal{N}(\mu_x, \sigma_x^2)$.
Then $\hat{x}[-1|-1] = E[x[-1]] = \mu_x$ and $M[-1|-1] = \sigma_x^2$.

Kalman Filter Versions:

1. scalar state x, scalar measurement z (above)
2. vector state $\mathbf{x}$, scalar measurement z
3. vector state $\mathbf{x}$, vector measurement $\mathbf{z}$

Typical Kalman Filter Scenarios in Computer Vision:

Estimating 2D position of object based on 2D observation of object in image

Estimating 2D position and 2D velocity vector of object based on 2D observations of object in image

Estimating 3D position of object based on 2D observations of objects in multi-camera video

Estimating 3D position of object based on RGB-Depth camera observations (3D)

Vector State/Vector Measurement Kalman Filter

$\mathbf{x}(t) \in \mathbb{R}^p$, $\mathbf{u}(t) \in \mathbb{R}^r$ with $\mathcal{N}(0, C_q)$, $\mathbf{x}[-1] \sim \mathcal{N}(\boldsymbol{\mu}_x, C_x)$

$\mathbf{z}(t) \in \mathbb{R}^m$, $\mathbf{w}(t) \in \mathbb{R}^m$ with $\mathcal{N}(0, C_w)$

Dynamics Equations:

State Equation:

$$\mathbf{x}[t] = A\mathbf{x}[t-1] + B\mathbf{q}[t]$$

A and B are known $p \times p$ by $p \times r$ matrices.

Measurement Equation:

$$\mathbf{z}[t] = H[t]\,\mathbf{x}[t] + \mathbf{w}[t]$$

$H[t]$ is a known $m \times p$ measurement matrix (which may be time varying).

Update Equations:

1. State Prediction:

$$\hat{\mathbf{x}}[t|t-1] = A\,\hat{\mathbf{x}}[t{-}1|t{-}1]$$

2. MSE Prediction:

$$\mathbf{M}[t|t{-}1] = A\,\mathbf{M}[t{-}1|t{-}1]A^T + BC_qB^T$$

3. Kalman Gain Computation:

$$\mathbf{K}[t] = M[t|t-1]H^T[t]\;\left(C_w[t] + H[t]\,M[t|t-1]H^T[t]\right)^{-1}$$

4. State Estimation (= Correction):

$$\hat{\mathbf{x}}[t|t] = \hat{\mathbf{x}}[t|t-1] + \mathbf{K}[t]\;\left(\mathbf{z}[t] - H[t]\,\hat{\mathbf{x}}[t|t-1]\right)$$

5. MSE Estimation:

$$\mathbf{M}[t|t] = (1 - K[t])\,H[t]\,\mathbf{M}[t|t-1]$$

Initialization: $\hat{\mathbf{x}}[-1|-1] = \boldsymbol{\mu}_x$ and $M[-1|-1] = C_x$.

Extended Kalman Filter

Student question:

What if we have a scenario with the state equation and/or measurement equation that is nonlinear?

Example:

We need to estimate the position of a car.

Instead of Cartesian location (l_x, l_y), we have its distance (range) R and its bearing α, $\mathbf{z}[t] = (R[t], \alpha[t])^T$, as measurements:

$$R[t] = \sqrt{l_x^2[t] + l_y^2[t]}$$

$$\alpha[t] = \arctan \frac{l_y[t]}{l_x[t]}$$

Our state and measurement equations are not linear.

Trick: Linearize the equations and then apply the regular Kalman filter.

Proof Example for 1D Kalman Filter:

Here is how you can prove that the first prediction equation of the 1D Kalman filter is indeed $\hat{x}[t|t-1] = a\ \hat{x}[t{-}1|t{-}1]$.

The state equation is $x[t] = ax[t{-}1] + u[t]$.

Let us define the notation $X[t] = x[0], ..., x[t]$.

The minimum mean square error estimator of $\hat{x}$ is the mean of the PDF:

$$\begin{aligned}
\hat{x}[t|t-1] &= E[\, x[t] \mid X[t{-}1]\,] \\
&= E[\, ax[t{-}1] + q[t] \mid X[t{-}1]\,] \\
&= E[ax[t{-}1] \mid X[t{-}1]\,] + E[q[t] \mid X[t{-}1]\,] \text{ since } x \text{ and } q \text{ are uncorrelated} \\
&= E[ax[t{-}1] \mid X[t{-}1]\,] \text{ since } q \text{ is white noise} \\
&= a\, E[x[t{-}1] \mid X[t{-}1]\,] \text{ since } a \text{ is a deterministic constant} \\
&= a\, \hat{x}[t{-}1 \mid t{-}1] \text{ applying the definition of } \hat{x}
\end{aligned}$$

Your Exercise:

Use similar arguments to prove the second prediction equation of the 2D Kalman filter:

$M[t|t{-}1] = a^2\, M[t{-}1|t{-}1] + \sigma_q^2$

Exercise 2:

Prove that the state estimation equation of the 1D Kalman filter can also be written as:

$$\hat{x}[t|t] = (1 - K[t])\, a\, \hat{x}[t-1|t-1] + K[t]\, z[t]$$

Interpret this equation for

(a) $K[t] = 0.5$

(b) $K[t] > 0.5$

(c) $K[t] < 0.5$

Exercise 3:

Which of the five update equations of the Kalman filter can be computed offline, which means in advance of obtaining any measurements?

Learning Outcomes

Be able to

- Define what a stochastic process and the Markov assumption are
- Define the concepts of 'white noise' and 'stationary process'
- Give an example of a 1D stochastic process that describes the linear movement of an object
- Explain the purpose of a Kalman filter
- Define the state and measurement equations of a 1D Kalman filter
- Give the Kalman filter optimality criterion
- Define and explain the Kalman state estimation (update) equation
- Explain what the other four Kalman update equations do, when you are given the expression, both in the scalar and vector case
- Explain what $\mathbf{x}$ and $\mathbf{z}$ might be in typical computer vision scenarios
- Mention the "Extended Kalman Filter" as a solution to a nonlinear tracking scenario
- Follow proofs of the Kalman equations