Implementation of Kalman Filter

Georgios Stagakis

April 26, 2023

Outline

Semantics

We assume vector p, containing the coordinates and respective speed of a point,

$$p = [s, u]^T = [x, y, u_x, u_y]^T.$$

where $s = [x, y]$ and $u = [u_x, u_y]$

By $a.$, we represent the respective accelaration for coordinate "$\cdot$".

Kalman main hypotheses

The main Kalman assumption for a point in position k is that it follows the two equations,

$$p_k^1 = Ap_{k-1} + Bq_{k-1} + e_1 \tag{1}$$
$$p_k^2 = Hp_k + e_2 \tag{2}$$

The final estimation is based on weighting the results of the assumptions,

$$\hat{p}_k = w_1 p_k^1 + w_2 p_k^2$$

Equation 1

The first equation from slide in page 4 is associated with the fact that every point is associated with the previous step p_{k-1}, plus an optional handling q_{k-1}, plus Gaussian Noise.

In our case $q_{k-1} = [a_x, a_y]$ and A, B are regulated by the Laws of motion (page 6).

Equation 2

The second equation from the slide in page 4 is associated with the fact every point contains Gaussian Noise, not associated with the previous position p_{k-1}, just by our lack of information in the specific current location.

e_1 and e_2 are independent of each other.

Matrices from Equation 1

$$p_k^1 = \begin{bmatrix} s_k \\ u_k \end{bmatrix}$$

$$p_k^1 = \begin{bmatrix} s_{k-1} + \Delta t \times u_{k-1} + \Delta t^2 \times a_{k-1}/2 \\ u_{k-1} + \Delta t \times a_{k-1} \end{bmatrix}$$

$$p_k^1 = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_{k-1} \\ u_{k-1} \end{bmatrix} + \begin{bmatrix} \Delta t^2/2 \\ \Delta t \end{bmatrix} a_{k-1}$$

$$p_k^1 = A\ p_{k-1} + B\ a_{k-1} + e_1$$

Matrix H from Equation 2

In Kalman's simplest version, speed on new points doesn't contain extra noise, so

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

Process noise covariance matrix Q

The covariance matrix for the noise e_1 is set as

$$Q = \begin{bmatrix} \Delta t^2/2 \times \Delta t^2/2 & 0 & \Delta t^2/2 \times \Delta t & 0 \\ 0 & \Delta t^2/2 \times \Delta t^2/2 & 0 & \Delta t^2/2 \times \Delta t \\ \Delta t^2/2 \times \Delta t & 0 & \Delta t \times \Delta t & 0 \\ 0 & \Delta t^2/2 \times \Delta t & 0 & \Delta t \times \Delta t \end{bmatrix} \sigma_1^2.$$

σ_1^2 represents a general estimation for the noise in the data. The rest of the values in the matrix are chosen respectively to provide the impact of each value to the final model, based on the laws of motion.

Measurement noise covariance matrix R

Again, the covariance matrix for the noise e_2,

$$R = \begin{bmatrix} \Delta t^2/2 \times \Delta t^2/2 & 0 \\ 0 & \Delta t^2/2 \times \Delta t^2/2 \end{bmatrix} \sigma_2^2.$$

Steps: p_k^1

We estimate p_k^1, from the previous points, as discussed above,

$$p_k^1 = A p_{k-1}^* + B q_{k-1},$$

where p_{k-1}^* is the previous "clean" estimation.

Steps: p_k^2

For p_k^2, we need,

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

and

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}.$$

When we receive the observation of the new location o_k, by eq.2,

$$p_k^2 = o_k = Hp_k^1 + e_1.$$

Steps: p_k

Finally,

$$p_k = p_k^1 + K_k(o_k - Hp_k^1),$$

and for the next iteration,

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

Thank you!!!

References

https://machinelearningspace.com/2d-object-tracking-using-kalman-filter/

https://machinelearningspace.com/object-tracking-python/