Scalar Kalman Filter Implementation

Dhruv Ahuja

June 2023

1 Introduction

Because independence is both a necessary and sufficient condition for uncorrelatedness in Jointly Gaussian Random Variables, the estimator $\hat{X}(Y)$ that minimizes the mean square error: $E[(X-\hat{X}(Y))^2]$, the MMSE: $E[X|Y]$ is equal to the LLSE: $Proj_{Span(Y,1)}(X)$. This is beneficial because we can efficiently compute the MMSE (Computing the LLSE is generally easier and faster than computing the MMSE as the LLSE is simply a few arithmetic operations). The Kalman Filter leverages this property of Jointly Gaussian Random Variables for efficient online estimation. Jointly Gaussian RV's are fairly common in practice, and thus the Kalman Filter is widely used in fields such as position estimation(GPS, robot tracking, etc).

2 Algorithm Definition and Params

We define the following State Space Model.

$$X_n = aX_{n-1} + V_n$$

$$Y_n = cX_n + W_n$$

where X_n represents the state at time step n, a represents how X changes. In the multivariate case, a could be some vector with position and velocity. V and W are independent (generally zero mean) Gaussian Noise, with V denoted as process noise and W denoted as observation noise. Given some initial state X_0 which is Gaussian or constant, then the state of the system X_k is a jointly Gaussian random variable as it's a linear combination of other independent Gaussian's(the previous $X's$ and V and W).
The Kalman Filter uses the orthogonal updates equation for online estimation(Online meaning we continuously receive new observations at discrete time steps. When we receive a new observation, we update our guess of the state.) The goal of the Kalman Filter is to do this efficiently. Instead of recomputing: $L[X_n|Y_1, Y_2, ..., Y_n]$ in the brute force way which takes linear time with respect to the number of observations, the Kalman Filter utilizes orthogonal updates so that a single update takes constant time.

The Kalman Filter does online estimation using the orthogonal updates equation:

$$L[X_n|Y_1, Y_2, ..., Y_n] = L[X_n|Y_1, Y_2, ..., Y_{n-1}] + L[X|\tilde{Y_n}]$$

where $\tilde{Y_n} = Y_n - proj_{Y_1,..Y_{n-1}}Y_n$. This only holds if X is zero mean, however, a similar equation holds if X isn't zero mean (We simply don't double count E[X].)
The first term in the sum is our best guess of X given n-1 observations: $\hat{X}_{n|n-1}$, and it's computation represents the "prediction step", and is computed without the nth observation, Y_n. Additionally, in the prediction phase, we update the estimation variance(The variance of our error, $\sigma^2_{n|n-1}$). The second term in the summation is the projection of X_n onto the new observation which has been orthogonalized, $\tilde{Y_n}$. We call the coefficient of this projection the Kalman Gain: K_n. The Kalman Gain can also be computed in the prediction phase, for it doesn't depend on the value of the nth observation(See proof in appendix). In the update step, we receive the nth observation, and thus update our guess, $\hat{X}_{n|n}$ and additionally our estimation variance $\sigma^2_{n|n}$. The intuition for why the prediction phase is even required is that the update phase relies heavily on the prediction phase as proven below. Here are the equations used in the prediction and update steps. The derivations for these equations can be found in the appendix below.

2.1 Predictions

$$\hat{X}_{n|n-1} = a\hat{X}_{n-1|n-1} + \mu_V$$

$$\sigma^2_{n|n-1} = Var(X_n - \hat{X}_{n|n-1}) = a^2\sigma^2_{n-1|n-1} + \sigma^2_V$$

$$K_n = \frac{c\sigma^2_{n|n-1}}{c^2\sigma^2_{n|n-1} + \sigma^2_W}$$

2.2 Update

$$\hat{X}_{n|n} = \hat{X}_{n|n-1} + K_n\tilde{Y}_n$$

$$\sigma^2_{n|n} = (1 - cK_n)\sigma^2_{n|n-1}$$

As seen, the calculations made in the "prediction" phase are central to the update phase

3 Running the Code

To run the code on the default model, simply run "python3 main.py" in the command line. The default model sets both noises to be standard Normals and sets $a = c = 1$. Additionally, the initial state X_0 is set to zero in addition to the estimation variance $\sigma^2_{0|0}$To specify params, type –param param. For

example, if we wanted a to be 3,
we would run "python3 main.py –a 3" Here all of the possible params and names:
$"a", "v_variance", "v_mean", c, "w_variance", "w_mean", "initial_state", "estimation_variance"$.
The program will then continuously ask for measurements, and at each step, return $\hat{X}_{n|n}$. Optionally, you either place measurements in a text file called "measurements.txt" or you can either place $measurments, true_states$ in a csv file called measurements.csv. The program will default to the csv file, then the txt file, and if none of these file exists, only then will it ask for continuous observations. For the txt file and csv file functionalities, the code will plot the data over each time step. For convenience, I have written files that will generate sample csv or txt files(See MeasurementGenerator.py and MeasurementsWithStatesGenerator.py)

4 Appendix

4.1 Prediction

1.

$$\hat{X}_{n|n-1} = a\hat{X}_{n-1|n-1} + \mu_V$$

Pf: By definition,

$$\hat{X}_{n|n-1} = proj_{span(Y_1,Y_2,...,Y_{n-1})}X_n = L[X_n|Y_1,Y_2,...,Y_{n-1}] = L[aX_{n-1}+V_n|Y_1,Y_2,...,Y_{n-1}]$$

By Linearity of LLSE

$$L[aX_{n-1}+V_n|Y_1,Y_2,...,Y_{n-1}] = L[aX_{n-1}|Y_1,Y_2,...,Y_{n-1}]+L[V_n|Y_1,Y_2,...,Y_{n-1}]$$

Recall,

$$L[X_n|Y_1,Y_2,...,Y_{n-1}] = \langle X_n,\ 1\rangle 1 + \sum_{k=1}^{n-1} \frac{cov(X_n, \tilde{Y}_k)}{Var(\tilde{Y}_k)}\tilde{Y}_k$$

For the second term in the summation, note that the noise at the nth time step is independent of the first n -1 observations, so

$$L[V_n|Y_1,Y_2,...,Y_{n-1}] = \langle V_n,\ 1\rangle 1 = E[V_n] = \mu_V$$

For the first term, we can pull out the a by linearity of LLSE, and term reduces to

$$a\hat{X}_{n-1|n-1}$$

Thus,

$$\hat{X}_{n|n-1} = a\hat{X}_{n-1|n-1} + \mu_V$$

Now, we show: 2.

$$\sigma^2_{n|n-1} = a^2\sigma^2_{n-1|n-1} + \sigma^2_V$$

Pf:

$$\sigma^2_{n|n-1} = Var(X_n - \hat{X}_{n|n-1}) = Var(X_n - a\hat{X}_{n-1|n-1}) = Var(aX_{n-1}+V_n - a\hat{X}_{n-1|n-1})$$

$$Var(aX_{n-1} + V_n - a\hat{X}_{n-1|n-1}) = Var(aX_{n-1} - a\hat{X}_{n-1|n-1}) + Var(V_n)$$

by independence Then, we can pull the a out — $Var(aX) = a^2Var(X)$
Finally, we get:

$$\sigma^2_{n|n-1} = a^2Var(X_{n-1} - \hat{X}_{n-1|n-1}) + Var(V_n) = a^2\sigma^2_{n-1|n-1} + \sigma^2_V$$

Now we show:
3.

$$K_n = \frac{c\sigma^2_{n|n-1}}{c^2\sigma^2_{n|n-1} + \sigma^2_W}$$

Pf:

$$K_n = comp_{\tilde{Y}_n}X_n = cov(X_n, \tilde{Y}_n)Var(\tilde{Y}_n)^{-1}$$

Recall that

$$\tilde{Y}_n = Y_n - proj_{Span(1,Y_1,Y_2,...,Y_{n-1})}Y_n$$

(by Graham-Schmidt) which means that $\tilde{Y}_n$ is zero mean because it's orthogonal to 1. Lemma:

$$\tilde{Y_n} = Y_n - \hat{X}_{n|n-1} - \mu_W$$

Pf:

$$\tilde{Y}_n = Y_n - proj_{Span(1,Y_1,Y_2,...,Y_{n-1})}Y_n$$

$$proj_{Span(1,Y_1,Y_2,...,Y_{n-1})}Y_n =$$

$$proj_{Span(1,Y_1,Y_2,...,Y_{n-1})}(cX_n+W_n) = L[cX_n|Y_1, Y_2, ..., Y_{n-1}]+L[W_n|Y_1, Y_2, ..., Y_{n-1}]$$

$$L[cX_n|Y_1, Y_2, ..., Y_{n-1}] = c\hat{X}_{n|n-1}$$

Thus, we have

$$\tilde{Y_n} = Y_n - c\hat{X}_{n|n-1} - \mu_W$$

Going back to the original proof, we have

$$cov(X_n, \tilde{Y}_n) = cov(X_n, Y_n - c\hat{X}_{n|n-1} - \mu_w) = cov(X_n - \hat{X}_{n|n-1}, Y_n - c\hat{X}_{n|n-1} - \mu_w)$$

We can do the above because the innovation $\tilde{Y}$ is orthogonal to $\hat{X}_{n|n-1}$ Thus, if we were to split this up using bi-linearity of covariance, the term $cov(\hat{X}_{n|n-1}, Y_n - cX_n + W_n) = 0$.

$$cov(X_n - \hat{X}_{n|n-1}, Y_n - c\hat{X}_{n|n-1} - \mu_w) = cov(X_n - \hat{X}_{n|n-1}, cX_n + W_n - c\hat{X}_{n|n-1} - \mu_w)$$

$$cov(X_n - \hat{X}_{n|n-1}, cX_n + W_n - c\hat{X}_{n|n-1} - \mu_w) = c*cov(X_n - \hat{X}_{n|n-1}, X_n - \hat{X}_{n|n-1}) + cov(X_n - \hat{X}_{n|n-1}, W_n - \mu_w)$$

$$c * cov(X_n - \hat{X}_{n|n-1}, X_n - \hat{X}_{n|n-1}) = c\sigma^2_{n|n-1}$$

$$cov(X_n - \hat{X}_{n|n-1}, W_n - \mu_w) = cov(X_n - \hat{X}_{n|n-1}, W_n) + cov(X_n - \hat{X}_{n|n-1}, \mu_w) = 0+0 = 0$$

$Var(\tilde{Y}_n) = Var(cX_n + W_n - c\hat{X}_{n|n-1} - \mu_W) = Var(cX_n - c\hat{X}_{n|n-1}) + Var(-W_n - \mu_W) = c^2\sigma^2_{n|n-1} + \sigma^2_W$

We factor the c out and the -1 and realize $Var(X - c) = Var(X)$
Thus

$$K_n = comp_{\tilde{Y}_n} X_n = cov(X_n, \tilde{Y}_n) Var(\tilde{Y}_n)^{-1} = c\sigma^2_{n|n-1}(c^2\sigma^2_{n|n-1} + \sigma^2_W)^{-1}$$

Thus completing the derivation, and concludes the prediction step.

4.2 Update Step Calculations

1.

$$\hat{X}_{n|n} = \hat{X}_{n|n-1} + K_n\tilde{Y}_n$$

Pf: This follows directly from the orthogonal equation.

$\hat{X}_{n|n} = L[X_n|Y_1, Y_2, ..., Y_n] = L[X_n|Y_1, Y_2, ..., Y_n - 1] + L[X_n|\tilde{Y}_n] = \hat{X}_{n|n-1} + K_n\tilde{Y}_n$

2.

$$\sigma^2_{n|n} = (1 - K_n^2)\sigma^2_{n|n-1}$$

Pf:

$\sigma^2_{n|n} = Var(X_n - \hat{X}_{n|n}) = Var(X_n - \hat{X}_{n|n-1} - K_n\tilde{Y}_n = \sigma^2_{n|n-1} + Var(-K_n\tilde{Y}_n) + 2Cov(X_n - \hat{X}_{n|n-1}, -K_n\tilde{Y}_n)$

$$Var(-K_n\tilde{Y}_n) = K_n^2 Var(\tilde{Y}_n) = K_n^2(c^2\sigma^2_{n|n-1} + \sigma^2_W$$

Calculation in prediction step for Variance

$$2Cov(X_n - \hat{X}_{n|n-1}, -K_n\tilde{Y}_n) = 2Cov(X_n, -K_n\tilde{Y}_n)$$

by bi-linearity of Covariance.

$$2Cov(X_n, -K_n\tilde{Y}_n) = -2K_n c\sigma^2_{n|n-1}$$

Calculation in prediction step for Covariance Thus, we have:

$$\sigma^2_{n|n} = \sigma^2_{n|n-1} + K_n^2(c^2\sigma^2_{n|n-1} + \sigma^2_W) - 2K_n\sigma^2_{n|n-1} = \sigma^2_{n|n-1} + \frac{(c\sigma^2_{n|n-1})^2}{(c^2\sigma^2_{n|n-1} + \sigma^2_W)^2}(c^2\sigma^2_{n|n-1} + \sigma^2_W) - 2cK_n\sigma^2_{n|n-1}$$

We further simplify

$\sigma^2_{n|n} = \sigma^2_{n|n-1} + c\sigma^2_{n|n-1}K_n - 2cK_n\sigma^2_{n|n-1} = \sigma^2_{n|n-1} - cK_n\sigma^2_{n|n-1} = (1 - cK_n)\sigma^2_{n|n-1}$

Thus,

$$\sigma^2_{n|n} = (1 - cK_n)\sigma^2_{n|n-1}$$

This concludes the appendix.