Lecture Notes: Covariances and matrix notation for filtering

Filtering can be generalized to tracking a state of arbitrary dimension. The 1D example problem we have been using consists of two variables, position and velocity. In order to move to an arbitrary state we will switch to matrix notation. This requires an understanding of covariances. Similar to how a single variable can be associated with a variance, a set of variables is associated with a covariance.

Formally, the covariance of two variables x and y is defined as:

$$\text{covariance(x, y)} = E[(x - E[x])(y - E[y])] \tag{1}$$

where $E[]$ is the expected value (or mean). Thus covariance is a measure of how much one variable deviates from its mean multiplied by how much another variable deviates from its mean. This is closely related to the definition of correlation, which is defined as:

$$\text{correlation(x, y)} = E[(x - E[x])(y - E[y])]/(\sigma_x \sigma_y) \tag{2}$$

The difference between them is that correlation is dimensionless (has no units), while covariance is defined in units of the two variables multiplied together.

Informally, two variables are said to be correlated if a linear change in one is related to a linear change in the other. For example, the height and weight of humans tends to be correlated. The taller someone is, the heavier. When data is plotted, the more linear it looks, the higher the correlation. Figure 1 shows two sets of data. The data on the left have a fairly strong correlation, while the data on the right are are almost completely uncorrelated.

Covariance can be imagined as how much one variable tends to vary with another variable. For example, consider the weight and blood pressure of a person. Both of these tend to waiver up and down over time, defining the variance of each. They also co-vary, in that when weight is above its usual mean value, blood pressure tends to be above its usual mean value. Thus, the variances are to some degree related. Numerically, a covariance value of zero indicates the two variables are independent. A positive value indicates a directionally similar variance, as in the weight and blood pressure example, while a negative value indicates a directionally opposite variance.

There are also occasions where it is useful to visualize covariance as an N-dimensional ellipse. For example, consider a tracking problem in two dimensions where the state variables are $[x, y]$. The state can be envisioned as an ellipse where the center of ellipse is the mean position, and the radii of the ellipse define the uncertainty. Figure 2 illustrates an example.

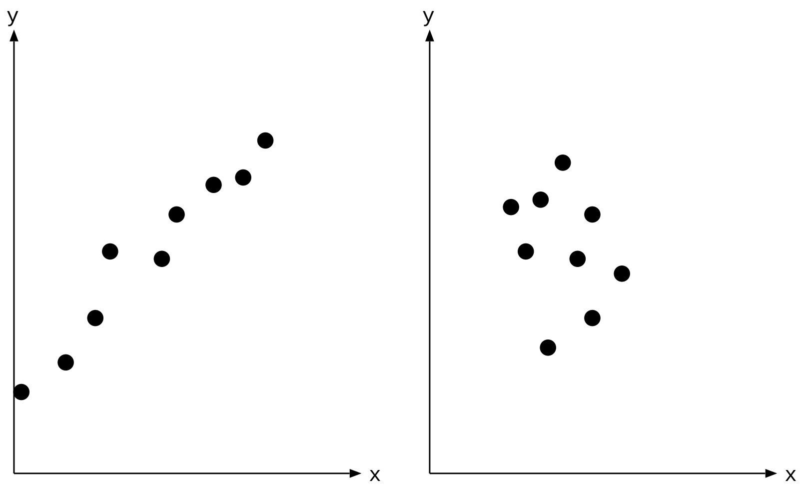

Figure 1: The data on the left show a strong correlation; the data on the right do not show a correlation.

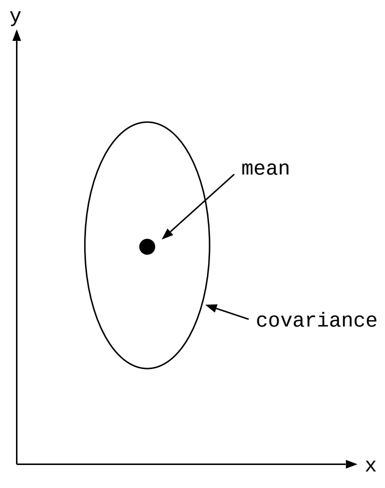

Figure 2: Visualizing covariance as an ellipse.

A covariance matrix gives the covariances of a set of variables. For example, given the matrix X defined as

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{3}$$

The covariance of matrix X is

$$\text{COV}(X) = \begin{bmatrix} \sigma_{x_1}^2 & \sigma_{x_1,x_2} \\ \sigma_{x_1,x_2} & \sigma_{x_2}^2 \end{bmatrix} \tag{4}$$

The diagonal elements of a covariance matrix are the variances of the individual variables. The non-diagonal elements are covariances. Note that $\sigma_{x_1,x_2} \neq \sigma_{x_1}\sigma_{x_2}$ (see the definition from above if this is not clear). Covariance matrices are symmetric.

In filtering, there are three covariance matrices of interest: the measurement noise covariance, the dynamic noise covariance, and the state estimate covariance. Each of these will be defined below as the related equations are discussed.

In switching to matrix notation, my notes use capital letters for matrices. All of the equations discussed previously, including those for measurements, state transitions, predictions and updates, can all be written in matrix form for an arbitrary number of state variables.

The **measurement equation** can be written as

$$Y_t = MX_t + N_t \tag{5}$$

where Y_t is the measurements, X_t is the actual state of the system at time t, M is the observation matrix, and N_t is the random noise during sensing. The measurement equation from the 1D example problem can be seen as one exemplar of equation 5 as follows:

$$\begin{bmatrix} y_t \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_t \\ \dot{x}_t \end{bmatrix} + \begin{bmatrix} N(0, \sigma_n^2) \end{bmatrix} \tag{6}$$

It is important to note that in this notation, X_t representing the true state is different from $X_{t,t}$ representing the filtered estimate of the state. We never know the true state. Equation 5 is a conceptual equation, not something that is coded or implemented.

The **measurement noise covariance** is defined as $\text{COV}(N)$, the covariance of the set of measured noises. For our 1D example problem, the measurement noise covariance is

$$R = \text{COV}(N) = [\sigma_n^2] \tag{7}$$

Because there is only one measurement, the position y_t, the covariance matrix contains only the variance of the measurement noise of position.

The **state transition equation** can be written as

$$X_{t+1} = \Phi X_t + A_t \tag{8}$$

where X_t is the current actual state, X_{t+1} is the next actual state, Φ is the state transition matrix, and A_t is the random dynamics during the sensing interval. The state transition equation from the 1D example problem can be seen as one exemplar of equation 8 as follows:

$$\begin{bmatrix} x_{t+1} \\ \dot{x}_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ \dot{x}_t \end{bmatrix} + \begin{bmatrix} 0 \\ N(0, \sigma_a^2) \end{bmatrix} \tag{9}$$

As with the measurement equation, this is a conceptual equation and is not something that is coded or implemented during the normal operation of a filter.

The **dynamic noise covariance** is defined as COV(A), the covariance of the set of dynamic noises. For our 1D example problem, the measurement noise covariance is

$$Q = \text{COV}(A) = \begin{bmatrix} 0 & 0 \\ 0 & \sigma_a^2 \end{bmatrix} \tag{10}$$

Because there is no dynamic noise on the position portion of the transition equation, its variance is zero. Therefore the only covariance term is for the velocity portion of the dynamic noise.

The **state prediction equation** can be written as

$$X_{t+1,t} = \Phi X_{t,t} \tag{11}$$

Unlike equations 5 and 8, this equation is implemented during filtering. Between sensor readings, we assume that the system undergoes zero dynamic noise in order to predict its next state. It is this matrix that is weighted against the next sensor readings in order to find the updated (also called filtered) estimate.

The **state estimate covariance** is defined as COV(X), the covariance of the set of state variables. For our 1D example problem, the state estimate covariance is

$$S = \text{COV}(X) = \begin{bmatrix} \sigma_x^2 & \sigma_{x,\dot{x}} \\ \sigma_{x,\dot{x}} & \sigma_{\dot{x}}^2 \end{bmatrix} \tag{12}$$

Figure 2 is useful as an interpretation for the state estimate covariance, in that it provides a picture of the uncertainty of the values in the state estimate matrix $X_{t,t}$. The diagonal elements represent the uncertainty in the estimates of position and velocity, and the off-diagonal elements represent their covariances.

The **state update equation** can be written as

$$X_{t,t} = X_{t,t-1} + K_t(Y_t - MX_{t,t-1}) \tag{13}$$

where $X_{t,t-1}$ is the predicted state from the last filter iteration, and $X_{t,t}$ is the updated state for the current filter iteration given the new sensor readings Y_t. The matrix K_t is the Kalman gain matrix, which is the weights used to combine the estimates. The update equation from the 1D example problem can be seen as one exemplar of equation 13 as follows:

$$\begin{bmatrix} x_{t,t} \\ \dot{x}_{t,t} \end{bmatrix} = \begin{bmatrix} x_{t,t-1} \\ \dot{x}_{t,t-1} \end{bmatrix} + \begin{bmatrix} f(\text{variances}) \\ f(\text{variances}) \end{bmatrix} \left(\begin{bmatrix} y_t \end{bmatrix} - \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_{t,t-1} \\ \dot{x}_{t,t-1} \end{bmatrix} \right) \tag{14}$$

We know that the gain matrix is a function of the variances, but previously we only derived it for one variable. What does it look like for an arbitrary number of variables? Skipping the derivation, the Kalman gain matrix is calculated as

$$K_t = S_{t,t-1}M^T[R + MS_{t,t-1}M^T]^{-1} \tag{15}$$

In this equation we see two of the covariance matrices that were defined above. In the last lecture we derived the Kalman filter weights for a single variable (one combined estimate of two separate measurements). We saw during that derivation that the weighting term was a function of the variances of the two measurements. When the derivation is done in matrix form for an arbitrary number of variables, these terms involve covariances of the estimates. The equivalent of dividing by a variance (one of the steps during the derivation) is multiplying by the inverse of the covariance.

The matrix S, giving the covariance of the state estimate, is calculated iteratively just like the state is calculated iteratively. The **state covariance prediction equation** can be written as

$$S_{t+1,t} = \Phi S_{t,t} \Phi^T + Q \tag{16}$$

where Q is the dynamic noise covariance matrix defined above. The **state covariance update equation** can be written as

$$S_{t,t} = [I - K_t M] S_{t,t-1} \tag{17}$$

where I is the identity matrix. These equations are one of the neat points of the Kalman filter. We have an estimate at each time t of the uncertainty of the filtered estimate of state.

Now that we have all the notation and equations for the general Kalman filter, next lecture we will focus on the implementation details and how it works in practice.