Lecture Notes: Filtering

Model fitting is a useful technique for mitigating noise in sensor data. A set of sensor data can be used to calculate a model. The model can then be used to smooth out the noise and to make predictions about expected future readings. However, in the case of a tracking problem, the thing being tracked is usually not expected to follow the same behavior forever. (If it did, what would be the point of tracking it? Its behavior can be perfectly modeled and its future behavior is completely known.)

For example, consider the problem of tracking a plane traveling through the motion depicted in Figure 1. During one period of time, the plane is following a steady path that can be modeled by a straight line. At another time, the plane changes direction, following a different path for a period of time. Fitting a single model to all the data does not seem like a good way to smooth or predict future motion. At the transitions the model fit needs to be changed in order to be able to make better predictions. (It is important to differentiate between changing the model, i.e. the equations used to describe the behavior, and changing the model fit, i.e. the parameters of the model. In this case we are talking about the latter.) It is possible to conceive of methods that change model fits only at certain times, for example after a goodness-of-fit statistic goes below a threshold. (There are in fact filtering techniques that make use of this idea, but they are a topic for future discussion.) However, this could cause undesirable tracking during periods where the behavior of the thing being tracked is changing slowly but continuously. In this case, it is more appropriate to continuously change the model fit. In the extreme this leads to the idea of updating the model fit after every new sensor reading is obtained. This is how **filtering** works.

For example, suppose we are using a linear model $y = ax + b$ to track the plane motion in Figure 1. As each new radar reading is taken, we can update our estimates of a and b. We use the notation a_t and b_t to denote the time of the estimate of each of the model parameters. In the context of filtering, the model parameters are called **state variables**. The state variables are the quantities in one or more equations that describe the behavior of the thing being tracked.

When deciding what model to use, there is no "correct" answer. There can however be models that are less useful than others. In general the model should be useful for smoothing, making predictions, and capturing the range of expected behaviors of the thing being tracked. For the example problem above, the model provides trajectory direction but no information about velocity. In other words, knowing the equation $y = a_t x + b_t$ tells us the trajectory of the line being flown by the plane at time t, but the equation alone does not tell us how fast the plane is flying on that line. Therefore it is a poor model for making predictions.

Instead, we will adopt a new model. In order to simplify the math we will consider a 1D example of an object moving along an x axis. Our state variables will be $[x_t, \dot{x}_t]$, where x_t

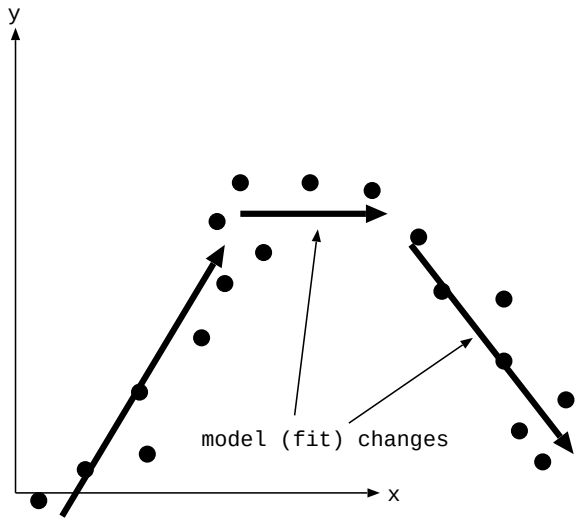

Figure 1: Tracking plane motion where the path can change at any time.

provides the position of the object and $\dot{x}_t$ provides the velocity of the object at time t.

The **state transition equations** are a set of equations that describe the range of expected behaviors of the thing being tracked. Like the state variables, these equations can be anything. For purposes of our example we will use the following:

$$x_t = x_{t-1} + \dot{x}_{t-1}T \quad (1)$$

$$\dot{x}_t = \dot{x}_{t-1} \quad (2)$$

where T is the interval of time between sensor readings. In practice the value T can be either a constant (if the sensor is operating on regular intervals) or a variable (if sensor readings are intermittent).

The **measurement equations** are a set of equations that describe the expected range of measurements given the current state of the thing being tracked. For our 1D example, we will use the following:

$$y_t = x_t \quad (3)$$

This equation simply states that we observe the position of the object along the x axis. (It is unfortunate that in filtering, y or z is often used to denote measurements, while x is used to denote state variables. This should not be confused with Cartesian coordinates.)

Now consider the situation shown in Figure 2. At time $t-1$ we estimated the object was at position $x_{t-1} = 0$ and traveling at a velocity of $\dot{x}_{t-1} = 200$ m/s. Our sensing interval T was 10 seconds. Therefore at time t we expect the object to be at position $x_t = 2000$. However, at time t we obtain a sensor reading of $y_t = 2060$. Do we believe the reading (the target is now moving at 206 m/s), or the prediction (the target is still moving at 200 m/s and we had a noisy measurement)? The answer is that we want to believe both, weighted appropriately. For the position, we can imagine:

$$x_t \text{ (updated)} = x_t + g_t(y_t - x_t) \quad (4)$$

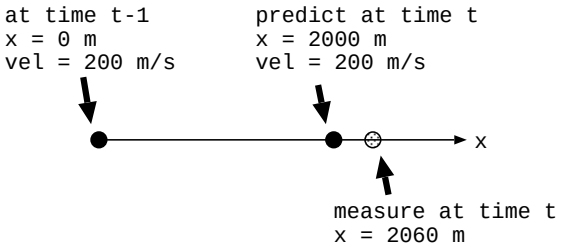

Figure 2: 1D example where prediction and measurement do not agree.

where $(y_t - x_t) = (2060 - 2000) = 60$ m, the difference between the predicted and observed positions. The variable g_t is a weight that indicates how much of the discrepancy is to be added into the updated estimate.

Note however the notation problem. The variable x_t is being used on the right-hand side of the equation to denote the predicted position, and on the left-hand side of the equation to denote the updated value of the position. To overcome this problem we adopt the following notation:

$$x_{t,t} = x_{t,t-1} + g_t(y_t - x_{t,t-1}) \tag{5}$$

where the first subscript indicates the current time and the second subscript indicates the time of the last measurement used to calculate the quantity. For the velocity we can write a similar equation:

$$\dot{x}_{t,t} = \dot{x}_{t,t-1} + h_t \frac{y_t - x_{t,t-1}}{T} \tag{6}$$

where $\frac{y_t - x_{t,t-1}}{T} = \frac{2060-2000}{10} = 6$ m/s, the difference between the predicted and observed velocities. The variable h_t is a weight that indicates how much of the discrepancy is to be added into the updated estimate. Equations 5-6 are known as the **state update equations**. Using the updated estimates and this notation, we can now rewrite the predictions (equations 1) as:

$$x_{t+1,t} = x_{t,t} + \dot{x}_{t,t}T \tag{7}$$

$$\dot{x}_{t+1,t} = \dot{x}_{t,t} \tag{8}$$

Equations 5-8 are an example of how a filter works. A filter operates in a perpetual cycle of **predict and update**, every time a new sensor reading is taken. Figure 3 shows the cycle. The current state estimate is used to make a prediction. When the next sensor reading is obtained, it is weighed against the prediction to update the state estimate. Each predict-update cycle is one iteration of filtering. The filter uses the model of motion (defined by the state transition equations) along with the sensed measurements (defined by the measurement equations) to estimate the state.

The specific example implemented in Equations 5-8 is a "constant velocity model", and is common for filtering problems. In early filtering papers these equations were called the g-h filter. They are a reasonable default model for tracking 1D signals over time when there is no more information about the expected behavior of the data that can be used to write

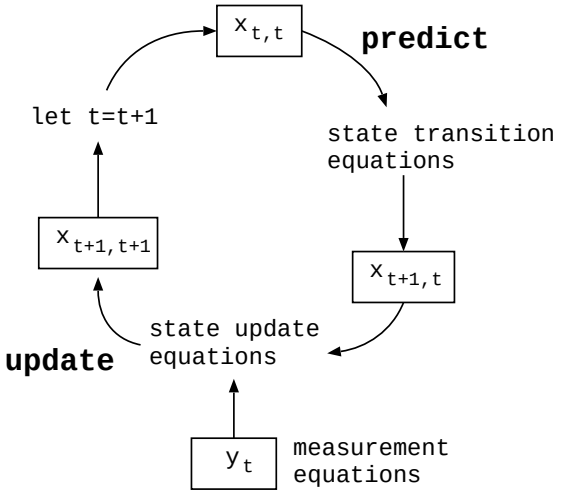

Figure 3: Filtering operates in a cycle of predict-update, combining information from the state transition equations with information from the measurement equations.

more detailed equations. We can use the Taylor expansion of a 1D signal to consider its accuracy:

$$x(t + \Delta t) = x(t) + \Delta t\dot{x}(t) + \frac{(\Delta t)^2}{2!}\ddot{x}(t) + \frac{(\Delta t)^3}{3!}\dddot{x}(t) + ... \tag{9}$$

For small Δt or for small $\ddot{x}(t)$ all but the first term are negligible. In other words, so long as the sampling rate is fast, or the acceleration and higher order terms are small, the accuracy is good.

The constant velocity model is not the only set of equations that can be used in filtering. For example, consider the problem of tracking a cannonball being fired from a cannon. The state transition equation can follow a parabolic arc. It may incorporate air resistance, and wind. For a second example, consider tracking a billiards ball as it moves around a table. The state transition equations can include piecewise functions that describe reflective bounces when the ball hits the boundary rails.

The measurement equations can also be far more complex. In the simple examples presented here, some or all of the state variables were directly observable. This is not always the case. For example, consider tracking the weather. The measurements include things like barometric pressure, temperature, and humidity, while the state variables include things like chance of precipitation and positions of cold fronts. For another example, consider tracking the stock market. The measurements include things like trading volumes, bond prices, interest rates and consumer indices, while the state variables include things like price predictions and volatility.

While the constant velocity model with direct observation of one or more state variables is useful for many problems, it is important to remember that it is only one of many possible sets of equations that can be used for a filtering problem.