4. Bayesian Tracking

This is at the heart of many optimal, recursive estimation algorithms.

- Let \( x(k), k = 0, 1, \ldots \), be the vector valued state we want to estimate – a discrete random variable which can only take on a finite number of values – \( X \) is finite.
- Let \( z(k), k = 1, 2, \ldots \), be a vector valued measurement, what we can observe. It can be a continuous or discrete random variable.

4.1 Model

\[
x(k) = q_k(x(k - 1), v(k)), \quad k = 1, 2, \ldots
\]
\[
z(k) = h_k(x(k), w(k))
\]
\(x(0), \{v(\cdot)\} \) and \( \{w(\cdot)\} \) are independent. Note that even though the known input \( u(k) \) is not explicitly included in the model, it can be implicitly modeled by absorbing it into \( q_k \) and \( h_k \).

4.2 Recursive equations

Let \( z(1 : k) \) denote the set \( \{z(1), \ldots, z(k)\} \). We want to calculate \( f(x(k)|z(1 : k)) \) efficiently.

- Assume \( f(x(k - 1)|z(1 : k - 1)) \) is known. For \( k = 1 \), we simply have \( f(x(0)) \), which is known. The recursion is then:

Prior update:

\[
f(x(k)|z(1 : k - 1)) = \sum_{x(k-1) \in X} f(x(k)|z(1 : k - 1), x(k-1)) f(x(k-1)|z(1 : k - 1))
\]

where the formula above follows from the total probability theorem. Note that \( x(k) \) and \( z(1 : k - 1) \) are conditionally independent, given \( x(k - 1) \):

- \( x(k) = q_k(x(k - 1), v(k)) \), a function of \( v(k) \) only.
- \( z(k - 1) = h_{k-1}(x(k - 1), w(k - 1)) \)
- \( z(k-2) = h_{k-2}(x(k - 2), w(k - 2)) \), \( x(k - 2) = q_{k-2}(x(k - 3), v(k - 2)) \), etc.
- \( \therefore z(1 : k - 1) = \text{function} (x(k - 1), v(1 : k - 2), w(1 : k - 1), x(0)) \)

Therefore \( x(k) \) and \( z(1 : k - 1) \) are conditionally independent, given \( x(k - 1) \). (HMWK)

\[
\therefore f(x(k)|z(1 : k - 1)) = \sum_{x(k-1) \in X} f(x(k)|x(k-1)) f(x(k-1)|z(1 : k - 1))
\]
and \( f(x(k)|x(k-1)) \) can be calculated from \( f(v(k)) \) and \( q_k(\cdot, \cdot) \) (although it may not be straight-forward to do so). This is an intuitive result: we use the process model to push our estimate forward in time. Note that conditional independence is crucial.

**Measurement update:**

\[
f(x(k)|z(1 : k)) = f(x(k)|z(k), z(1 : k-1)) \]

\[
= \frac{f(z(k)|x(k), z(1 : k-1)) f(x(k)|z(1 : k-1))}{f(z(k)|z(1 : k-1))} \quad \text{(Bayes’ rule)}
\]

Note that \( z(k) \) and \( z(1 : k-1) \) are conditionally independent, given \( x(k) \):

- \( z(k) = h_k(x(k), w(k)) \), a function of \( w(k) \) only.
- \( z(1 : k-1) = \text{function of } w(1 : k-1), x(0) \).

Therefore \( z(k) \) and \( z(1 : k-1) \) are conditionally independent, given \( x(k) \).

Therefore \( f(z(k)|x(k), z(1 : k-1)) = f(z(k)|x(k)) \), and furthermore, it can be calculated from \( f(w(k)) \) and \( h_k(\cdot, \cdot) \). Finally, \( f(z(k)|z(1 : k-1)) \) is simply a normalization constant that can be calculated using the total probability theorem:

\[
f(z(k)|z(1 : k-1)) = \sum_{x(k) \in \mathcal{X}} f(z(k)|x(k)) f(x(k)|z(1 : k-1))
\]

\[
\therefore \quad f(x(k)|z(1 : k)) = \frac{f(z(k)|x(k)) f(x(k)|z(1 : k-1))}{\sum_{x(k) \in \mathcal{X}} f(z(k)|x(k)) f(x(k)|z(1 : k-1))}
\]

**Summary**

\[
f(x(k)|z(1 : k-1)) = \sum_{x(k-1) \in \mathcal{X}} \frac{\text{process model}}{f(x(k)|x(k-1)) f(x(k-1)|z(1 : k-1)), \quad k = 1, 2, \ldots}
\]

\[
f(x(k)|z(1 : k)) = \frac{\text{measurement model}}{f(z(k)|x(k)) f(x(k)|z(1 : k-1))} \quad \frac{\text{prior}}{\sum_{x(k) \in \mathcal{X}} f(z(k)|x(k)) f(x(k)|z(1 : k-1))}
\]

### 4.3 Computer implementation

- Enumerate the state: \( \mathcal{X} = \{1, 2, \ldots, N\} \)
- Define \( \omega_{k|i}^0 := \Pr(x(k) = i|z(1 : k)) \), \( i = 1, \ldots, N \)
- Define \( \omega_{k|i}^k := \Pr(x(k) = i|z(1 : k - 1)) \), \( i = 1, \ldots, N \)

Then

**Initialization**, \( k = 0 \):

\[
\omega_{0|i}^0 = \Pr(x(0) = i), \quad i = 1, \ldots, N
\]
\( k > 0 \):

\[
\omega^i_{k|k-1} = \sum_{j=1}^{N} \Pr(x(k) = i|x(k-1) = j) \omega^i_{k-1|k-1}, \quad i = 1, \ldots, N
\]

\[
\omega^j_{k|k} = \frac{f(z(k)|x(k) = j) \omega^j_{k|k-1}}{N} \sum_{j=1}^{N} f(z(k)|x(k) = j) \omega^j_{k|k-1}, \quad i = 1, \ldots, N
\]

\( \text{Iterate.} \)

Note that \( \Pr(x(k) = i|x(k-1) = j) \) can be calculated from \( x(k) = q_k(x(k-1), v(k)) \) and \( f(v(k)) \). Similarly, \( f(z(k)|x(k) = i) \) can be calculated from \( z(k) = h_k(x(k), w(k)) \) and \( f(w(k)) \).

### 4.4 Example

Consider an object moving randomly on a circle. We can measure the distance to the object and want to estimate its location.

- Define \( x(k) \) as the object’s location on the circle, \( x(k) \in \{0, 1, \ldots, N - 1\} \), with \( \theta(k) = 2\pi \frac{x(k)}{N} \).
- We model the dynamics as \( x(k) = x(k-1) + v(k) \), where

\[
v(k) = \begin{cases} 
1 & \text{with probability } p \\
-1 & \text{with probability } 1 - p
\end{cases}
\]

\((N - 1) + 1 := 0 \\
0 - 1 := N - 1 .
\)

- Measurement:

\[ z(k) = \left( (L - \cos \theta(k))^2 + \sin^2 \theta(k) \right)^{\frac{1}{2}} + w(k), \]

where \( w(k) \), uniformly distributed on \([ -e, e ]\), is the sensor noise.

- We can now construct the PDFs of the process and sensor models:

\[
f(x(k)|x(k-1)) = \begin{cases} p & \text{if } x(k) = x(k-1) + 1 \\
1 - p & \text{if } x(k) = x(k-1) - 1 \\
0 & \text{otherwise}
\end{cases}
\]

\[
f(z(k)|x(k)) = \begin{cases} \frac{1}{2\pi} & \text{if } |z(k) - \left( (L - \cos \theta(k))^2 + \sin^2 \theta(k) \right)^{\frac{1}{2}}| < e \\
0 & \text{otherwise.}
\end{cases}
\]
• Initialization: \( f(x(0)) = \frac{1}{N} \forall x(0) \in \{0, 1, \ldots, N - 1\} \). This captures the state of maximum ignorance (entropy).

Simulations

\( N = 100 \Rightarrow 3.6^\circ \). Initial location: \( x(0) = N/4 = 25 \Rightarrow 90^\circ \). \( e = 0.50 \).

1) \( L = 2.0, p = 0.50 \). Notice bimodal distribution. The mean is a horrible estimate!

2) \( L = 2.0, p = 0.55 \). Decay.

3) \( L = 0.1, p = 0.55 \). Works!

4) \( L = 0.0, p = 0.55 \). Uniform for all time.

How robust is this? Set the nominal parameters to \( L = 2.0, e = 0.50, p = 0.55 \), but use different values for the estimator.

5) \( \hat{p} = 0.45, \hat{e} = e \). Of course we get it completely wrong.

6) \( \hat{p} = 0.50, \hat{e} = e \). We can’t differentiate.

7) \( \hat{p} = 0.90, \hat{e} = e \). Makes incorrect assumption.

8) \( \hat{p} = p, \hat{e} = 0.90 \). Washes things out.

9) \( \hat{p} = p, \hat{e} = 0.49 \). Crashes!