

4. Bayesian Tracking

We derive the first recursive state estimation algorithm in this course for a system with a finite state space. The algorithm has two main steps: 1) The prior update, where the state estimate is predicted forward using the process model; and 2) the measurement update, where the prior is combined with observations/measurements (using Bayes' theorem). These two steps are at the heart of many optimal, recursive estimation methods, and they will recur in the methods that we will develop for other classes of problems throughout this course.

4.1 Problem Statement (Finite State Systems)

Let $x(k) \in \mathcal{X}$ be the vector-valued state at time k ($k = 0, 1, \dots$) which we want to estimate. We assume it is a discrete random variable which can only take on a finite number of values; that is, \mathcal{X} is finite.

Let $z(k)$, $k = 1, 2, \dots$, be a vector valued measurement that we can observe. It can be a continuous or a discrete random variable.

Model

$$\begin{aligned} x(k) &= q_{k-1}(x(k-1), v(k-1)), \quad k = 1, 2, \dots \\ z(k) &= h_k(x(k), w(k)), \end{aligned}$$

where $x(0)$, $\{v(\cdot)\}$ and $\{w(\cdot)\}$ are independent with known PDFs. Note that even though the known input $u(k-1)$ is not explicitly included in the model, it can be implicitly modeled by absorbing it into q_{k-1} and h_k .

Objective

Let $z(1:k)$ denote the set $\{z(1), \dots, z(k)\}$. We want to calculate $f(x(k)|z(1:k))$ *efficiently*. Note that we want to estimate the full conditional *probability density function* of the state.

4.2 Recursive Estimator Equations

We develop a recursive algorithm, where we compute $f(x(k)|z(1:k))$ from $f(x(k-1)|z(1:k-1))$, the observation $z(k)$, and the system model. This way, we do not need to store the entire history of measurements $z(1:k)$.

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 consists of the prior update and the measurement update.

Prior update

First step: predict the PDF of $x(k)$ based on past measurements $z(1:k-1)$.

- By the total probability theorem,

$$f(x(k)|z(1:k-1)) = \sum_{x(k-1) \in \mathcal{X}} f(x(k)|z(1:k-1), x(k-1)) \underbrace{f(x(k-1)|z(1:k-1))}_{\text{assumed to be known}}.$$

- Note that $x(k)$ and $z(1:k-1)$ are conditionally independent, given $x(k-1)$:
 - $x(k) = q_{k-1}(x(k-1), v(k-1))$, a function of $v(k-1)$ only.
 - $z(k-1) = h_{k-1}(x(k-1), w(k-1))$
 - $z(k-2) = h_{k-2}(x(k-2), w(k-2))$, $x(k-2) = q_{k-3}(x(k-3), v(k-3))$, etc.
 - and, therefore,
 - $z(1:k-1) = \text{FUNCTION}\left(x(k-1), \underbrace{v(1:k-3), w(1:k-1), x(0)}_{\text{independent of } v(k-1)}\right)$
 - Therefore, $x(k)$ and $z(1:k-1)$ are conditionally independent, given $x(k-1)$. Therefore, $f(x(k)|z(1:k-1), x(k-1)) = f(x(k)|x(k-1))$. (PSET 2: P10)

The prior update is therefore given by:

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

where $f(x(k)|x(k-1))$ can be calculated from $f(v(k-1))$ and $q_{k-1}(\cdot, \cdot)$ using change of variables (although it may not be straightforward 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

Second step: combine prior with new observation.

- By Bayes' rule,

$$\begin{aligned} f(x(k)|z(1:k)) &= f(x(k)|z(k), z(1:k-1)) \\ &= \frac{f(z(k)|x(k), z(1:k-1)) f(x(k)|z(1:k-1))}{f(z(k)|z(1:k-1))}. \end{aligned}$$

- 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.
 - $z(1:k-1) = \text{FUNCTION}\left(\underbrace{v(1:k-2), w(1:k-1), x(0)}_{\text{independent of } w(k)}\right)$.
 - 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:

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

Using the above, the measurement update reads:

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

Summary

Prior update (state prediction):

$$f(x(k)|z(1:k-1)) = \sum_{x(k-1) \in \mathcal{X}} \overbrace{f(x(k)|x(k-1))}^{\text{process model}} \overbrace{f(x(k-1)|z(1:k-1))}^{\text{previous iteration}}, \quad k = 1, 2, \dots$$

Measurement update:

$$f(x(k)|z(1:k)) = \frac{\overbrace{f(z(k)|x(k))}^{\text{measurement model}} \overbrace{f(x(k)|z(1:k-1))}^{\text{prior}}}{\sum_{x(k) \in \mathcal{X}} f(z(k)|x(k)) f(x(k)|z(1:k-1))}$$

4.3 Computer Implementation

- Enumerate the state: $\mathcal{X} = \{1, 2, \dots, N\}$.
- Define $\mathbf{a}_{k|k}^i := \Pr(x(k) = i|z(1:k))$, $i = 1, \dots, N$, an array with N elements that we use to store the posterior $f(x(k)|z(1:k))$.
- Define $\mathbf{a}_{k|k-1}^i := \Pr(x(k) = i|z(1:k-1))$, $i = 1, \dots, N$, to store the prior $f(x(k)|z(1:k-1))$.

- **Algorithm:**

Initialization, $k = 0$:

$$\mathbf{a}_{0|0}^i = \Pr(x(0) = i), \quad i = 1, \dots, N$$

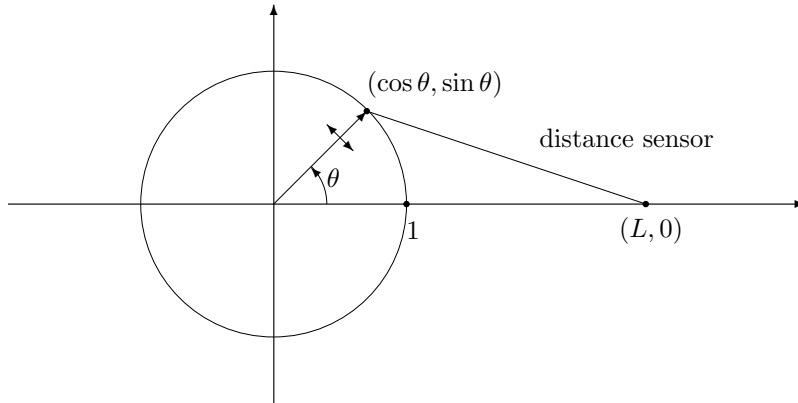
Recursion, $k > 0$:

$$\left. \begin{aligned} \mathbf{a}_{k|k-1}^i &= \sum_{j=1}^N \Pr(x(k) = i|x(k-1) = j) \mathbf{a}_{k-1|k-1}^j, \quad i = 1, \dots, N \\ \mathbf{a}_{k|k}^i &= \frac{f(z(k)|x(k) = i) \mathbf{a}_{k|k-1}^i}{\sum_{j=1}^N f(z(k)|x(k) = j) \mathbf{a}_{k|k-1}^j}, \quad i = 1, \dots, N \end{aligned} \right\} \text{Iterate.}$$

Note that $\Pr(x(k) = i|x(k-1) = j)$ can be calculated from $x(k) = q_{k-1}(x(k-1), v(k-1))$ and $f(v(k-1))$. 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 in discrete steps. We can measure the distance to the object and want to estimate its location.



- The object can move in discrete steps. Define $x(k)$ as the object's location on the circle, $x(k) \in \{0, 1, \dots, N-1\}$, with

$$\theta(k) = 2\pi \frac{x(k)}{N}.$$

- The dynamics are given by

$$x(k) = \text{mod}(x(k-1) + v(k-1), N),$$

where $\text{mod}(\cdot, N)$ denotes the modulo N operation; that is, $\text{mod}(N, N) = 0$ and $\text{mod}(-1, N) = N - 1$. The process noise is given by

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

- The model of the distance measurement is

$$z(k) = \sqrt{(L - \cos \theta(k))^2 + \sin^2 \theta(k)} + w(k),$$

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

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

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

$$f(z(k)|x(k)) = \begin{cases} \frac{1}{2e} & \text{if } \left| z(k) - \sqrt{(L - \cos \theta(k))^2 + \sin^2 \theta(k)} \right| \leq e \\ 0 & \text{otherwise.} \end{cases}$$

- Initialization: $f(x(0)) = 1/N$ for all $x(0) \in \{0, 1, \dots, N-1\}$. This captures the state of maximum ignorance.

Simulations

We choose $N = 100$, i.e. the object moves in steps of 3.6° . Initial location: $x(0) = N/4 = 25$, i.e. start at 90° . For the sensor noise, use $e = 0.50$.

- 1) $L = 2.0$, $p = 0.50$. Bimodal distribution. From the distance measurement, the algorithm cannot distinguish the upper versus the lower half plane. Notice that the mean would be a horrible estimate!
- 2) $L = 2.0$, $p = 0.55$. One mode decays. On average, the object is more likely to move counterclockwise; thus, the mode that corresponds to clockwise movement decays.
- 3) $L = 0.1$, $p = 0.55$. Works! (Sensor inside the circle.)
- 4) $L = 0.0$, $p = 0.55$. Uniform distribution 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, the estimator gets it completely wrong. It assumes that clockwise movement is more likely, but the opposite is actually true.
- 6) $\hat{p} = 0.50$, $\hat{e} = e$. The estimator can't differentiate between the two modes.
- 7) $\hat{p} = 0.90$, $\hat{e} = e$. Makes incorrect assumption (wrong process model).
- 8) $\hat{p} = p$, $\hat{e} = 0.90$. Washes things out (assumes larger uncertainty in the measurement).
- 9) $\hat{p} = p$, $\hat{e} = 0.49$. Crashes (division by zero)! Happens when a measurement cannot be explained by the model. Can be avoided by resampling.

(We encourage you to reproduce these simulation results yourself (PSET 2: P12).)