

Recursive Optimization

Andrius Buinovskij - 18-940-270

1 Bayesian Tracking

1.1 Problem Statement

$x(k) \in \mathcal{X}$ is our goal. It can be vector valued if We like, so multi-dimensional, but it **must be finite**, so We can't have an infinite number of states that We may inhabit. This is because We have to sum over the states at some point, and having an infinite number of them would be problematic.

k is simply the time, so $x(k)$ is the state at time k .

$z(k)$ is the observation, whatever it may be, at time k . Note that in contrast to the state, $z(k)$ can have an infinite number of different values e.g. $z(k) \in \mathbb{R}^d$.

Our **model** of the situation then is:

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

So which bits are what. $x(k)$ is just the state.

$v(k-1)$ is the **process noise** at time $k-1$. I suppose every state gets a little randomness.

Then the measurement $z(k)$ is a function of the state at time k and some noise.

q_{k-1} is a known function (which can implicitly be parametrized with whatever user input We have), and h_k is also a known function.

Summing up then, the new state is a function of old state and noise, and measurements are functions of the current state and some other noise.

Objective

Okay, let $z(1:k)$ denote the set $\{z(1) \dots z(k)\}$.

Our goal then is to efficiently compute

$$p(x(k)|z(1:k)) \tag{1}$$

Where, recall, the above can be written as:

$$p_{X|Z(1:k)}(\bar{x}(k)|\bar{z}(1:k)) \tag{2}$$

Where X is the random variable actually generating the states and \bar{x} is just a particular value that X could take at time k , and likewise for the noise variables.

1.2 Recursive Estimation Equations

The idea is simple - decompose the problem of computing $p(x(k)|z(1:k))$ into parts such that the end algorithm only depends on $p(x(k-1)|z(1:k-1))$, $z(k)$ and the known model parameters (q etc.).

So, in the name of recursion, We assume $p(x(k-1)|z(1:k-1))$ is known, which is fine since We are given $p(x(0))$.

Prior Update

Step 1: use $z(1:k-1)$ to get the distribution of $x(k)$:

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

So what did We do here?

Almost literally nothing, We simply added a variable to condition on - the previous state $x(k-1)$.

But there are many possible previous states, so We have to sum over all of them. Furthermore this was all conditioned on previous observations $z(1:k-1)$, hence everything remains conditioned on them.

However, this was very clever since We introduced something that We know - that "assumed to be known" bit.

I mean, ultimately, this conditioning on the previous state is a kind of obvious move if You think about it - think about the causality DAG of this whole process and You'll see We are creating a D-separated path.

So, the state $x(k)$ is independent of previous measurements given previous state. Works for any state obviously.

$$\begin{aligned} p(x(k)|z(1:k-1)) &= \sum_{x(k-1) \in \mathcal{X}} p(x(k)|z(1:k-1), x(k-1)) \underbrace{p(x(k-1)|z(1:k-1))}_{\text{assumed to be known}} \\ &\quad \sum_{x(k-1) \in \mathcal{X}} p(x(k)|x(k-1)) p(x(k-1)|z(1:k-1)) \end{aligned}$$

And the point of all that was to capture the information the previous state gives us about the current state. We can now augment this with information yielded by $z(k)$.

I suppose this is called the prior update because this is the distribution of $x(k)$ given the past measurements $z(1:k-1)$? And now we'll add $z(k)$ to the mix with a

Measurement update

So, now We add conditioning on $z(k)$:

$$p(x(k)|z(1:k)) = p(x(k)|z(k), z(1:k-1)) \quad (3)$$

$$= \frac{p(z(k)|x(k), z(1:k-1)) \cdot p(x(k)|z(1:k-1))}{p(z(k)|z(1:k-1))} \quad (4)$$

And that looks like a lot, but it's not so bad. Take it term by term.

$p(z(k)|x(k), z(1:k-1))$ can be simplified - if You think about the causal DAG again, You'll notice that measurement $z(k)$ is generated by the state $x(k)$, so knowing that blocks the path to all the ancestor measurements $z(1:k-1)$.

We already took a look at $p(x(k)|z(1:k-1))$ in the prior section.

Finally for the denominator. We can condition on the current state to turn this denominator into a summation We know:

$$p(z(k)|z(1:k-1)) = \sum_{x \in \mathcal{X}} p(z(k)|z(1:k-1), x(k)) \cdot p(x(k)|z(1:k-1)) \quad (5)$$

$$= \sum_{x \in \mathcal{X}} p(z(k)|x(k)) \cdot p(x(k)|z(1:k-1)) \quad (6)$$

And then We can calculate $p(z(k)|x(k))$ since it's a function of $x(k)$ and the other term We calculated

Summary

Prior update, which is state prediction since We are looking at probability of the state x at time k :

$$p(x(k)|z(1:k-1)) = \sum_{x(k-1) \in \mathcal{X}} \underbrace{p(x(k)|x(k-1))}_{\text{model function}} \cdot \underbrace{p(x(k-1)|z(1:k-1))}_{\text{previous iteration}} \quad (7)$$

So the first term can be calculated via the model function - We have functions to go from previous state to next state (given noise etc).

The second bit is of course a recursive assumption.

Then We observe $z(k)$ and update our belief accordingly.

2 Computer Implementation

1. Enumerate the state space \mathcal{X} . Seems reasonable.

2. Define $\mathbf{a}_{k|k}^i = p_{x(k)|z(1:k)}(i|\bar{z}(1:k))$, $i = 0, 1 \dots N-1$. This array stores the posterior, or rather our estimation of the posterior. Element i is just the probability of state i at time k given all the measurements, including the measurement at time k .

3. Define $\mathbf{a}_{k|k-1}^i = p_{x(k)|z(1:k-1)}(i|\bar{z}(1:k-1))$, $i = 0 \dots N-1$. This array stores our prior, i.e. $p(x(k)|z(1:k-1))$.

Algorithm

1. Initialization:

$$a_{0|0}^u = p_{x(0)}(i), i = 0 \dots N - 1$$

Which is given.

2. Recursion - for $k > 0$:

$$\mathbf{a}_{k|k-1}^k = \sum_{j=0}^{N-1} p_{x(k)|x(k-1)}(i|j) \mathbf{a}_{k-1|k-1}^j, \quad i = 0 \dots N - 1 \quad (1)$$

$$\mathbf{a}_{k|k}^i = \frac{p_{z(k)|x(k)}(\bar{z}(k)|i) \mathbf{a}_{k|k-1}^i}{\sum_{j=0}^{N-1} p_{z(k)|x(k)} \mathbf{a}_{k|k-1}^j} \quad (2)$$

3 Example

Alright so We have an object which is moving in a circle and We want to track it's location.

The $x(k)$ We care about is:

$$\theta(k) = 2\pi \frac{x(k)}{N} \quad (1)$$

And the new state as a function of the old state is:

$$x(k) = \text{mod}(x(k-1) + v(k-1), N) \quad (2)$$

Where mod is just the modulo operator.

The noise is then $v(k) = 1$ with probability r , $v(k) = 0$ with probability $1 - r$.

Oh boy and now the measurement - the distance:

$$z(k) = \sqrt{(L - \cos(\theta(k)))^2 + \sin^2(\theta(k)) + w(k)} \quad (3)$$

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

4 Appendix

4.1 Change of Variables

For discrete variables I think it's kind of trivial - the pdf stays the same (aside from summing identical mappings), no worries there.

Now for the continuous case.

First up us the univariate case yet again. We have that $Y = g(X)$, and We want F_Y , the cumulative distribution function of Y .

$$F_Y = P(Y \leq y) = P(g(X) \leq y) =$$