Recursive Updating – Fixed Parameter

So far we've considered the scenario where we collect a bunch of data and then use that (and the prior PDF) to compute the conditional PDF… from which we can then get the MMSE, MAP, or any other Bayesian estimator we wish.

Given batch (vector) of data $\mathbf{x}_N = (x_1, x_2, x_3, \ldots x_N)$ and the prior $p(\theta)$

Then Bayes' rule gives the conditional PDF we need:

$$p(\theta \mid \mathbf{x}_N) = \frac{p(\theta)\, p(\mathbf{x}_N \mid \theta)}{p(\mathbf{x}_N)} \qquad \text{or…} \qquad p(\theta \mid \mathbf{x}_N) \propto p(\theta)\, p(\mathbf{x}_N \mid \theta)$$

Proportionality constant… easy to find: makes this integrate to 1

So… no need to fuss about keeping track of the proportionality constant!
In fact, for MAP it is never needed!

$$p(\theta \mid \mathbf{x}_N) \propto p(\theta)\, p(\mathbf{x}_N \mid \theta) \quad \Rightarrow \quad \text{Posterior} \propto \text{Prior} \times \text{Likelihood}$$

"New" Name!

But… suppose our data arrives sequentially as $x_1, x_2, x_3, \ldots x_N, \ldots$

Each time we get a new piece of data we want to update our conditional PDF (then… extract the MMSE, MAP, etc. as desired)

But… we do NOT want to compute with ALL the PAST data!

Just use current conditional PDF and the one new data point!!

A useful result for dealing with this sequential setting is:

$$p\left(x_1, x_2, \ldots, x_n \mid \theta\right) = p\left(x_1 \mid \theta\right) p\left(x_2 \mid x_1, \theta\right) p\left(x_3 \mid x_1, x_2, \theta\right) \cdots p\left(x_n \mid \mathbf{x}_{n-1}, \theta\right)$$

Note that this a generalization of what we saw earlier and called "**Decomposing the Joint PDF**" $p_{XY}(x, y) = p_{Y|X}(y \mid x) p_X(x)$

Now we can use this decomposition to see how to sequentially update the posterior PDF on which Bayes' estimation is based:

Consider the **posterior PDF** after the first data point x_1 and then as we collect more and more data:

$$p(\theta \mid x_1) \propto p(\theta)\, p(x_1 \mid \theta)$$

Acts like a new "Prior"!

Acts like a new "Likelihood"!

$$p(\theta \mid x_1, x_2) \propto p(\theta)\, p(x_1, x_2 \mid \theta) = p(\theta)\, p(x_1 \mid \theta)\, p(x_2 \mid x_1, \theta) \propto p(\theta \mid x_1)\, p(x_2 \mid x_1, \theta)$$

$$p(\theta \mid x_1, x_2, x_3) \propto p(\theta)\, p(x_1 \mid \theta)\, p(x_2 \mid x_1, \theta)\, p(x_3 \mid x_1, x_2, \theta) \propto p(\theta \mid x_1, x_2)\, p(x_3 \mid x_1, x_2, \theta)$$

⋮

This leads to the Bayesian Recursive Update Viewpoint:

$$p(\theta \mid \mathbf{x}_N) \propto p(\theta \mid \mathbf{x}_{N-1})\, p(x_N \mid \mathbf{x}_{N-1}, \theta)$$

New Posterior ∝ Old Posterior × New Likelihood

Generalization of Posterior ∝ Prior × Likelihood

Recursive Updating – Evolving "State"

Based on pp. 580 – 582 of Moon & Stirling, *Mathematical Methods & Algorithms for Signal Processing*, Prentice-Hall, 2000

In the previous we consider fixed state – does not change w/ time

Now the state changes w/ time: $\mathbf{s}_n$ (we now abandon θ notation for "parameter")

Goal: Observe $x_0, x_1, x_2, \ldots$ and estimate $\mathbf{s}_0, \mathbf{s}_1, \mathbf{s}_2, \ldots$

Note: State is vector! Measurements here are scalars (but could be generalized to vectors)

Need some Notation!

$$\mathbf{X}_n = [x_0 \quad x_1 \quad x_2 \cdots x_n]^T$$ (All measurements up to time *n*)

$\hat{\mathbf{s}}_{n|k}$ = estimate of state $\mathbf{s}_n$ based on $\mathbf{X}_k$ (data up to time *k*)

We can only proceed if we have a appropriate models!

State Dynamics Model: The state is a "Markov" Random Process:

$$p\left(\mathbf{s}_{n+1} \mid \mathbf{s}_n, \mathbf{s}_{n-1}, \ldots, \mathbf{s}_0\right) = p\left(\mathbf{s}_{n+1} \mid \mathbf{s}_n\right)$$

Next state is conditionally indep of all past states except most recent one!

This captures the "dynamics" of the state…
How does the state "transition" from time *n* to time *n*+1

$$p\left(f\left(\mathbf{s}_{n+1}\right) \mid \mathbf{s}_n, \mathbf{s}_{n-1}, \ldots, \mathbf{s}_0\right) = p\left(f\left(\mathbf{s}_{n+1}\right) \mid \mathbf{s}_n\right)$$

Holds also for function of state $\mathbf{s}_{n+1}$

Measurement Model:

$$p\left(x_{n+1} \mid \mathbf{s}_{n+1}, x_n, x_{n-1}, \ldots, x_0\right) = p\left(x_{n+1} \mid \mathbf{s}_{n+1}\right)$$

or equivalently $p\left(x_{n+1} \mid \mathbf{s}_{n+1}, \mathbf{X}_n\right) = p\left(x_{n+1} \mid \mathbf{s}_{n+1}\right)$

e.g., Meas. $\mathbf{x}_{n+1}$ depends only on state $\mathbf{s}_{n+1}$ plus noise that is indep sample-to-sample

Combining these two gives a useful result:

$$p\left(x_{n+1} \mid \mathbf{s}_{n+1}, \mathbf{s}_n, \mathbf{s}_{n-1}, \ldots, \mathbf{s}_0, \mathbf{X}_n\right) = p\left(x_{n+1} \mid \mathbf{s}_{n+1}\right)$$

Basic First Steps of Recursion

Given Prior PDF on initial state we can get an estimate

$$p(\mathbf{s}_0) \Rightarrow \hat{\mathbf{s}}_{0|-1}$$ **(via MMSE, MAP, Etc.)**

Think like this: $p(\mathbf{s}_0 \mid x_{-1})$

We don't have data @ n = –1 so view this as "prior to data"

Update Posterior and State Estimate using measurement x_0:

$$p(\mathbf{s}_0 \mid x_0) = \frac{p(x_0 \mid \mathbf{s}_0)\, p(\mathbf{s}_0)}{p(x_0)} \Rightarrow \hat{\mathbf{s}}_{0|0}$$ **(via MMSE, MAP, Etc.)**

"**Propagate**" (somehow!) this posterior to $p(s_1|x_0)$

$$p(\mathbf{s}_0 \mid x_0) \xrightarrow{\text{Prop.}} p(\mathbf{s}_1 \mid x_0) \Rightarrow \hat{\mathbf{s}}_{1|0}$$ **(via MMSE, MAP, Etc.)**

Iterate: Treat $p(s_1|x_0)$ as a "new prior" and go to "Update"

That results in $p(s_1|\mathbf{X}_1)$, etc.

We now need to flesh this idea out!

Bayesian Recursion for Evolving State

From the above "Basic" view we can see that we really want some function $\mathcal{F}\{\ \}$ to go from $p(\mathbf{s}_n|\mathbf{X}_n)$ to $p(\mathbf{s}_{n+1}|\mathbf{X}_{n+1})$; that is

$$p\left(\mathbf{s}_{n+1} \mid \mathbf{X}_{n+1}\right) = \mathcal{F}\left\{p\left(\mathbf{s}_n \mid \mathbf{X}_n\right), x_{n+1}\right\}$$

We'll see that this is done in two steps: "Update" & "Propagate"

From Bayes' Theorem we can write

$$p\left(\mathbf{s}_{n+1} \mid \mathbf{X}_{n+1}\right) = p\left(\mathbf{s}_{n+1} \mid \mathbf{X}_n, x_{n+1}\right)$$

$$= \frac{p\left(x_{n+1} \mid \mathbf{s}_{n+1}, \mathbf{X}_n\right)}{p\left(x_{n+1} \mid \mathbf{X}_n\right)} p\left(\mathbf{s}_{n+1} \mid \mathbf{X}_n\right)$$

Normalizer

From measurement model, this $= p(x_{n+1}|\mathbf{s}_{n+1})$… so we can write:

$$p\left(\mathbf{s}_{n+1} \mid \mathbf{X}_{n+1}\right) \propto p\left(x_{n+1} \mid \mathbf{s}_{n+1}\right) p\left(\mathbf{s}_{n+1} \mid \mathbf{X}_n\right)$$

Update Step

Posterior — Meas. Model — "Prior"

Measurement Model gives the Likelihood Function!

But the update step needs that "prior" $p(\mathbf{s}_{n+1}|\mathbf{X}_n)$

Finding it yields the "Propagation" step!

We find it using the concept of "Marginal from Joint":

$$p\left(\mathbf{s}_{n+1} \mid \mathbf{X}_n\right) = \int p\left(\mathbf{s}_{n+1}, \mathbf{s}_n \mid \mathbf{X}_n\right) d\mathbf{s}_n$$

Then decomposing the PDF inside the integral gives

$$p\left(\mathbf{s}_{n+1}, \mathbf{s}_n \mid \mathbf{X}_n\right) = p\left(\mathbf{s}_{n+1} \mid \mathbf{s}_n, \mathbf{X}_n\right) p\left(\mathbf{s}_n \mid \mathbf{X}_n\right)$$

depends on $\mathbf{s}_n$ and noise that is indep sample-to-sample

$$= p\left(\mathbf{s}_{n+1} \mid \mathbf{s}_n\right) p\left(\mathbf{s}_n \mid \mathbf{X}_n\right)$$

Using this inside the integral gives the "propagation" step:

$$p\left(\mathbf{s}_{n+1} \mid \mathbf{X}_n\right) = \int p\left(\mathbf{s}_{n+1} \mid \mathbf{s}_n\right) p\left(\mathbf{s}_n \mid \mathbf{X}_n\right) d\mathbf{s}_n$$

Propagation Step

Prior for Update

State "Dynamic" Model

Posterior

Results for Bayesian Recursion for Evolving State

Update Step

$$p(\mathbf{s}_{n+1} \mid \mathbf{X}_{n+1}) \propto p(x_{n+1} \mid \mathbf{s}_{n+1})\, p(\mathbf{s}_{n+1} \mid \mathbf{X}_n)$$

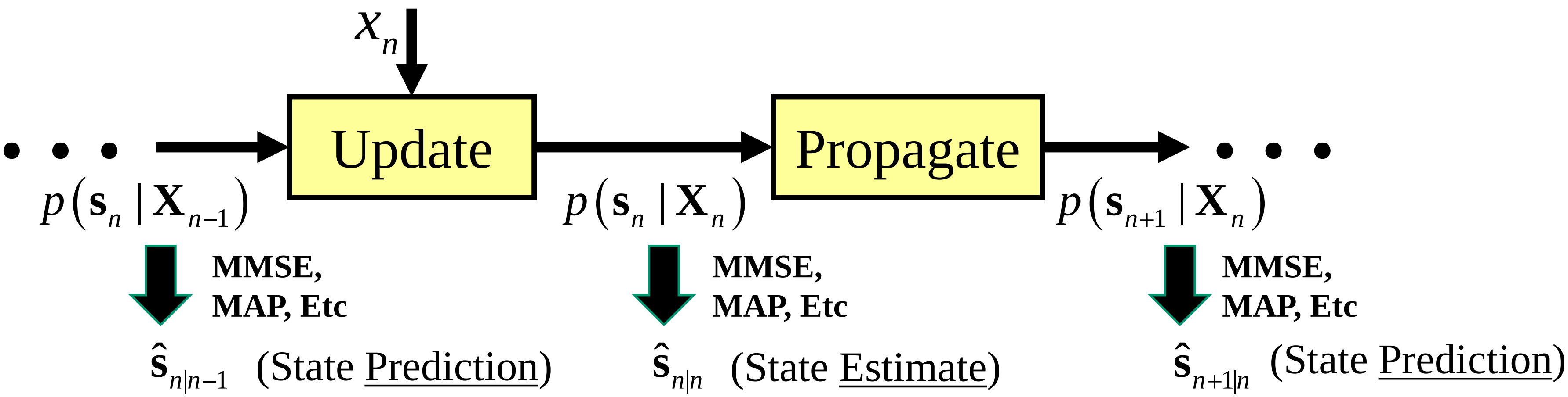
Posterior — Measurement Model — "Prior"

Propagation Step

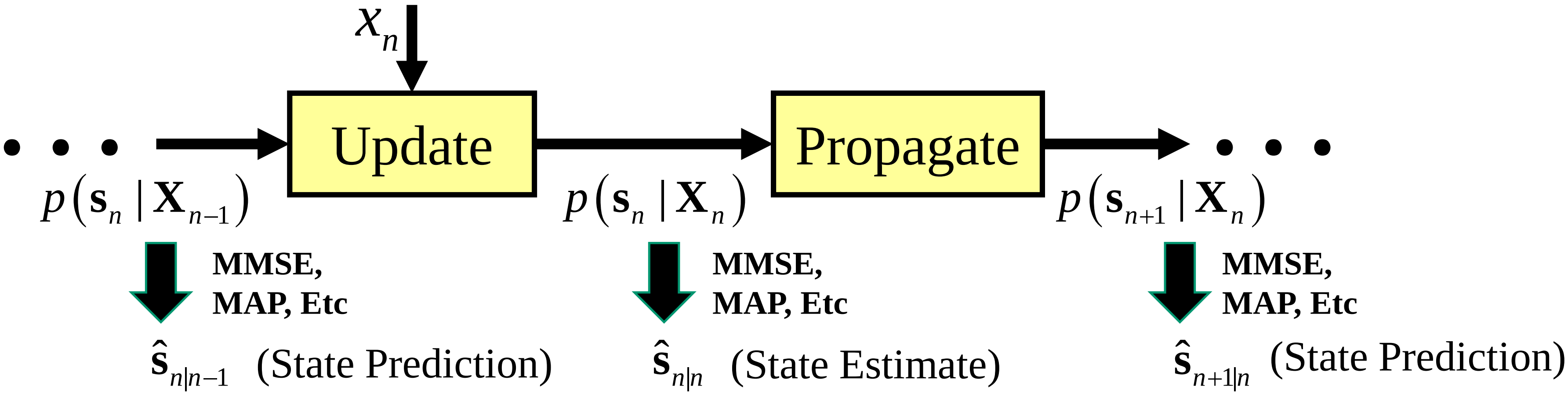
$$p(\mathbf{s}_{n+1} \mid \mathbf{X}_n) = \int p(\mathbf{s}_{n+1} \mid \mathbf{s}_n)\, p(\mathbf{s}_n \mid \mathbf{X}_n)\, d\mathbf{s}_n$$

Prior for Update — State "Dynamic" Model — Posterior

We need: (i) Initial Prior, (ii) State Model, and (iii) Measurement Model



Can also extract covariances from these generated PDFs



What happens when all the PDFs are guaranteed to be Gaussian?

Only need to update & propagate cond. means and covariances!