Mathematics and numerics for data assimilation and state estimation – Lecture 6

Mathematics for Uncertainty Quantification

RWTH AACHEN UNIVERSITY

Summer semester 2020

Summary of lecture 5

- Markov property:

$$\mathbb{P}(Z_{n+1} = z_{n+1}, Z_n = z_n, \ldots, Z_0 = z_0) \\ = \mathbb{P}(Z_{n+1} = z_{n+1} \mid Z_n = z_n)\, \mathbb{P}(Z_n = z_n, \ldots, Z_0 = z_0)\,. \quad (1)$$

- time-homogeneous chains *Markov*(π, p) with transition function

$$\mathbb{P}(Z_{n+1} = j \mid Z_n = i) = p(i,j) \quad \text{whenever } \mathbb{P}(Z_n = i) > 0.$$

- evolution of distributions

$\pi^n(i) = \mathbb{P}(Z_n = i)$

$$\pi^n = \pi^0 p^n$$

and invariant distributions

$$\pi = \pi p$$

- aperiodicity of states and irreduciblity and recurrence of p.

Overview

1. Recurrence and construction of invariant distributions for time-homogeneous finite Markov chains
2. Filtering
3. Prediction
4. Smoothing

Overview

1. Recurrence and construction of invariant distributions for time-homogeneous finite Markov chains
2. Filtering
3. Prediction
4. Smoothing

Recurrence and construction of invariant distributions

Definition 1

Consider an **irreducible** transition function p associated to a state-space $\mathbb{S}$. Then we say that p is recurrent if it for any state $i \in \mathbb{S}$ and Markov chain $\{Z_n^i\} \sim Markov(\mathbb{1}_{\{i\}}, p)$ holds that

$$\mathbb{P}(Z_n^i = i \quad \text{for infinitely many } n) = 1, \tag{2}$$

which for the hitting time $T_i := \inf\{n \geq 1 \mid Z_n^i = i\}$ is equivalent to

$$\mathbb{P}(T_i < \infty) = 1.$$

Lemma 2

If p is **irreducible** *and the state-space is finite, then p is recurrent.*

Proof: Let us write $\mathbb{S} = \{1, 2, \ldots, d\}$. Since $\mathbb{S}$ is finite, there must be at least one pair of states $i, j \in \mathbb{S}$ satisfying

$$\mathbb{P}(Z_n^i = j \quad \text{for infinitely many } n) > 0 \tag{3}$$

since otherwise we reach the contradiction

$$\begin{aligned} 0 &= bP(Z_n^i \notin \mathbb{S} \quad \text{for infinitely many } n) \\ &\geq 1 - \sum_{j \in \mathbb{S}} bP(Z_n^i = j \quad \text{for infinitely many } n) = 1. \end{aligned}$$

And

$$\begin{aligned} &\mathbb{P}(Z_n^j = j \quad \text{for infinitely many } n) \\ &= \mathbb{P}(Z_n^i = j \quad \text{for infinitely many } n \cap \{Z_n^i = j \quad \text{for some } n\}) \qquad (4) \\ &= \mathbb{P}(Z_n^i = j \quad \text{for infinitely many } n) > 0. \end{aligned}$$

Theorem 9, Lecture 4 extends to the current setting, so by defining

$$N^j := \sum_{n \in \mathbb{N}} \mathbb{1}_{Z_n^j = j} \quad \text{(total visits at state } j\text{)},$$

we obtain for $\lambda_j := \mathbb{P}(T^j < \infty)$ that

$$\mathbb{P}(N^j = k) = \begin{cases} (1 - \lambda_j)\lambda_j^{k-1} & \text{if } \lambda_j < 1 \\ \mathbb{1}_{k=\infty} & \text{if } \lambda_j = 1 \end{cases}$$

Consequently,

$$0 < \mathbb{P}(Z_n^j = j \quad \text{for infinitely many } n) = \mathbb{P}(N^j = \infty) = \begin{cases} 0 & \text{if } \lambda_j < 1 \\ 1 & \text{if } \lambda_j = 1. \end{cases}$$

Conclusion: λ_j must equal 1 and j is a recurrent state.

It remains to verify that $N^k = \infty$ a.s. for all $k \in \mathbb{S} \setminus \{j\}$. Observe first that

$$\mathbb{P}(N^k = \infty) = 1 \iff \mathbb{P}(N^k = \infty) > 0$$
$$\iff \mathbb{E}\left[N^k \right] = \infty \iff \sum_{n \in \mathbb{N}} p^n_{kk} = \infty$$

where the last $\iff$ follows from

$$\mathbb{E}\left[N^k \right] = \sum_{n \in \mathbb{N}} \mathbb{E}\left[\mathbb{1}_{Z^k_n = k} \right] = \sum_{n \in \mathbb{N}} \mathbb{P}(\mathbb{1}_{Z^k_n = k}) = \sum_{n \in \mathbb{N}} p^n_{kk}.$$

Since $\mathbb{P}(N^j = \infty) = 1$, we know that $\sum_{n \in \mathbb{N}} p^n_{jj} = \infty$. And by the irreducibility of p, there exist $m_1, m_2 \geq 1$ such that $p^{m_1}_{kj} p^{m_2}_{jk} > 0$. So for any $n \geq m_1 + m_2$,

$$p^n_{kk} \geq p^{m_1}_{kj} p^{n-(m_1+m_2)}_{jj} p^{m_2}_{jk}$$

and

$$\sum_{n \in \mathbb{N}} p^n_{kk} \geq p^{m_1}_{kj} p^{m_2}_{jk} \sum_{n \in \mathbb{N}} p^n_{jj} = \infty.$$

Q.E.D.

Construction of invariant measures

For an irreducible transition function p associated to $\mathbb{S} = \{1, 2, \dots, d\}$, we fix a state $k \in \mathbb{S}$, the chain $\{Z_n^k\} \sim \mathit{Markov}(\mathbb{1}_{\{k\}}, p)$ and introduce

$$\gamma_j^k := \mathbb{E}\left[\sum_{n=0}^{T^k-1} \mathbb{1}_{Z_n^k = j}\right] \quad \text{for} \quad j \in \mathbb{S}.$$

(the expected number of visits spent at state j in between vists to k).

Theorem 3 (Theorem 1.7.5, Norris, Markov Chains)

For every $k \in \mathbb{S}$,

$$\gamma^k = \gamma^k p,$$

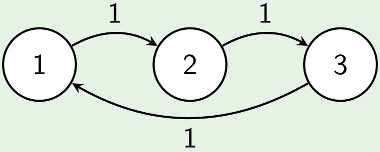
which makes

$$\pi := \frac{\gamma^k}{\sum_{j\in\mathbb{S}} \gamma_j^k}$$

is an invariant distribution.

Example 4

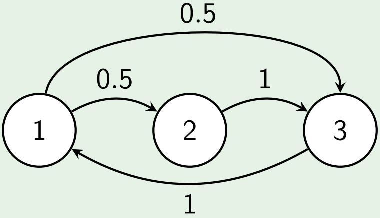
$$p = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$



Irreducible but periodic chain. $p_{ii}^n > 0$ only for $n = 3, 6, 9, \ldots$. So Lemma 19 does not apply.
But $\gamma^1 = \gamma^2 = \gamma^3 = [1,\ 1,\ 1]$, giving rise to $\pi = \gamma^1/3$.

Example 5

$$p = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$



Irreducible chain with aperiodic state 3. So Lemma 19 does apply.
But theorem 22 also:

$$\gamma^1 = [1, 0.5, 1], \quad \gamma^2 = [2, 1, 2], \quad \gamma^3 = [1, 0.5, 1]$$

Simulation of a time-homogeneous Markov chain

For $\{Z_n\} \sim Markov(\pi^0, p)$ on $\mathbb{S} = \{1, 2, \ldots, d\}$ the main challenges for simulation are to draw the inital state and the transitions:

1. Draw $Z_0 \sim \pi^0 = (\pi_1^0, \ldots, \pi_d^0)$
2. ...
3. given $Z_n = i$, draw $Z_{n+1} \sim [p_{i1}, p_{i,2}, \ldots, p_{id}]$

Same challenge for every step: draw a sample/new state from a distribution $f = [f_1, \ldots, f_d]$.

Sampling method:

1. construct a vector

$\bar{f}$ = cumsum(f)

$$\bar{f} = \begin{pmatrix} f_1 \\ f_1 + f_2 \\ \vdots \\ \sum_{j=1}^{d-1} f_j \\ 1 \end{pmatrix}$$

u = rand();

2. Draw a uniformly distributed rv $U \sim U[0,1]$ and determine new state by:

$$j(U) := \min\{k \in \{1, 2, \dots, d\} \mid \bar{f}_k > U\}.$$

Exercise: verify that $\mathbb{P}(j(U) = \ell) = f_\ell$.

Data assimilation of Markov Chains

Let $\{Z_n\}$ with $Z_n = (X_n, Y_n)$ denote a time-homogeneous Markov chain.

$Y_{0:n} = (Y_0, \dots, Y_n)$

For observations $Y_{0:n}$ related to a signal of interest $X_{0:n}$ we consider the following conditional estimation problems:

- Prediction: $X_k | Y_{0:j}$ for $j < k$,
- Filtering: $X_k | Y_{0:k}$,
- Smoothing $X_k | Y_{0:T}$ for $T > k$.

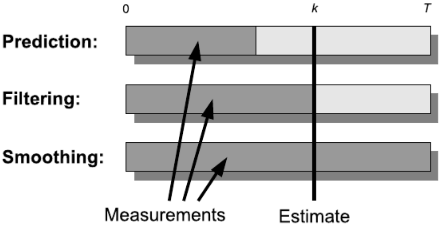


Figure: From “Bayesian Filtering and Smoohting” by S. Särrkä.

Overview

Filtering setting

Time homogeneous Markov chain $\{Z_n\} = \{(X_n, Y_n)\}$ with

- countable state-space C, since

$$(X_n, Y_n) : \Omega \to A \times B =: C,$$

- and transition function $p : C \times C \to C$ satisfying

$$\mathbb{P}(Z_{n+1} = c_{n+1} \mid Z_n = c_n) = p(c_n, c_{n+1}) \quad \text{whenever } \mathbb{P}(Z_n = c_n) > 0.$$

- For every $n \geq 0$, recall that $Y_{0:n} = (Y_0, Y_1, \ldots, Y_n)$ is the history of observations
- and we seek the state of the signal of interest X_n given $Y_{0:n}$.

Examples

- Random walk $Z_n = (X_n, Y_n)$ on $\mathbb{Z}^2$.
- Discrete Markov chain X_n on $\mathbb{S} = \mathbb{Z}^d$ with $Y_n = HX_n + W_n$ for some matrix $H \in \mathbb{Z}^{k \times d}$ and with W_n a random walk on $\mathbb{Z}^k$.
- Discrete Markov chain X_n on $\mathbb{S}$ with $Y_n = X_{\lfloor n/5 \rfloor}$ (new observation every fifth time unit).
- Hidden Markov models: X_n a discrete Markov chain and

$$Y_n = \gamma(X_n, W_n)$$

  where $\{W_n\}$ are iid and $\{X_n\}$ and $\{W_n\}$ are independent.

Note Z_n being a Markov chain does not imply that either of X_n or Y_n is:

Example 6

1,2 1,2

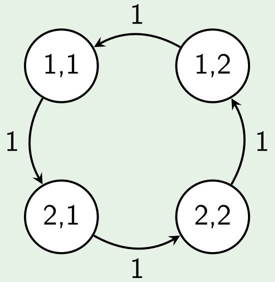
Consider chain Z_n on $\{0,1\} \times \{0,1\}$ and X_n and Y_n discrete processes on $A = B = \{\cancel{0,1}\}$ 1,2, say with uniformly random initial condition, to make the chain stochastic.

It is then clear that for $n > 1$,

$$\mathbb{P}(X_n = 1 \mid X_{n-1} = 2) = 1/2,$$

while

$$\mathbb{P}(X_n = 1 \mid X_{n-1} = 2, X_{n-2} = 2) = 1.$$



How detailed state-information do we seek?

- Best approximation in mean-square sense:

$$\tilde{X}_n := \mathbb{E}[X_n \mid Y_{0:n}] = \sum_{a \in A} a \mathbb{P}(X_n = a \mid Y_{0:n}).$$

- or perhaps the (more informative) conditional distribution

$$\mathbb{P}(X_n = a \mid Y_{0:n}) \quad \text{for relevant } a \in A.$$

Example 7 (Comparison of conditional expectation and distribution)

Let the sequence $Z_n = (X_n, Y_n)$ be a simple symmetric random walk on $\mathbb{Z}^2$ with $Z_0 = (0,0)$. Then for any $n \geq 0$ and observation sequence $b_{0:n}$,

$$\mathbb{E}[X_n \mid Y_{0:n} = b_{0:n}] = 0$$

since

$$\mathbb{P}(X_n = a \mid Y_{0:n} = b_{0:n}) = \mathbb{P}(X_n = -a \mid Y_{0:n} = b_{0:n}) \quad \forall a \in A.$$

Conclusion: $\mathbb{P}(X_n = a \mid Y_{0:n} = b_{0:n})$ is not always needed to compute the associated conditional expectation.

Filtering setting 2

- We will consider observations of the kind $Y_{0:n} = b_{0:n}$, accumulating as $n \mapsto n+1$.
- We assume that $\mathbb{P}(Y_{0:n} = b_{0:n}) > 0$ for $n = 0, 1, \ldots$ (since these observations have occurred).
- Iteratively in time $n = 0, 1, \ldots$, we seek the conditional distribution

$$\mathbb{P}(X_n = a_n \mid Y_{0:n} = b_{0:n}) \quad \text{for relevant} \quad a_n \in A \tag{5}$$

- For efficiency, we seek a recursive algorithm, using the new measurement b_n to update the previous calculations of

$$\{\mathbb{P}(X_{n-1} = a_{n-1} \mid Y_{0:n-1} = b_{0:n-1})\}_{a_{n-1} \in A}$$

when computing (5).

Recursive algorithm

By definition,

$$\mathbb{P}(X_n = a \mid Y_{0:n} = b_{0:n}) = \frac{\mathbb{P}(X_n = a, Y_{0:n} = b_{0:n})}{\mathbb{P}(Y_{0:n} = b_{0:n})}, \tag{6}$$

Idea: Apply law of total probability

$$\mathbb{P}(X_n = a_n, Y_{0:n} = b_{0:n}) = \sum_{a_{0:n-1} \in A^n} \mathbb{P}\Big(X_n = a_n, X_{0:n-1} = a_{0:n-1}, Y_{0:n} = b_{0:n}\Big)$$

and use the Markov property to render every summand computable

$$\begin{aligned}
&\mathbb{P}\Big(X_{0:n} = a_{0:n}, Y_{0:n} = b_{0:n}\Big) \\
&\qquad = \mathbb{P}\Big(X_n = a_n, Y_n = b_n \mid X_{n-1} = a_{n-1}, Y_{n-1} = b_{n-1}\Big) \\
&\qquad\qquad \times \mathbb{P}\Big(X_{0:n-1} = a_{0:n-1}, Y_{0:n-1} = b_{0:n-1}\Big) = \dots
\end{aligned}$$

Simplification of idea

By the law of total probability and Markovianity [FJK Corrollary 2.2.7] yields

$$
\begin{aligned}
&\mathbb{P}(X_n = a, Y_{0:n} = b_{0:n}) \\
&\qquad = \sum_{r \in A} \mathbb{P}\Big(X_n = a, X_{n-1} = r, Y_{0:n} = b_{0:n}\Big) \\
&= \sum_{r \in A} \mathbb{P}\Big((X_n, Y_n) = (a, b_n), (X_{n-1}, Y_{n-1}) = (r, b_{n-1}), Y_{0:n-2} = b_{0:n-2}\Big) \\
&\qquad = \sum_{r \in A} \mathbb{P}\Big((X_n, Y_n) = (a, b_n) \mid (X_{n-1}, Y_{n-1}) = (r, b_{n-1})\Big) \\
&\qquad\qquad \times \mathbb{P}\Big((X_{n-1}, Y_{n-1}) = (r, b_{n-1}), Y_{0:n-2} = b_{0:n-2}\Big)
\end{aligned}
$$

Motivation last equality?

Recursive algorithm

Recalling that on positive probability conditioned events,

$$\mathbb{P}\Big((X_n, Y_n) = (a, b_n) \mid (X_{n-1}, Y_{n-1}) = (r, b_{n-1})\Big) = p((r, b_{n-1}), (a, b_n)) \\ =: q^{ra}(b_{n-1}, b_n),$$

we have that

$$\mathbb{P}(X_n = a, Y_{0:n} = b_{0:n}) = \sum_{r \in A} q^{ra}(b_{n-1}, b_n)\, \mathbb{P}\Big(X_{n-1} = r, Y_{0:n-1} = b_{0:n-1}\Big) \tag{7}$$

Algorithm 1: Recursive relationship joint density

Let $\varphi_n^a(b_{0:n}) := \mathbb{P}(X_n = a, Y_{0:n} = b_{0:n})$. Then (7) yields

$$\varphi_n^a(b_{0:n}) = \sum_{r \in A} q^{ra}(b_{n-1}, b_n)\, \varphi_{n-1}^r(b_{0:n-1})$$

Algorithm 1 continued

Moreover,

$$\mathbb{P}(Y_{0:n} = b_{0:n}) = \sum_{r \in A} \varphi_n^r(b_{0:n})$$

and thus

$$\mathbb{P}(X_n = a \mid Y_{0:n} = b_{0:n}) = \frac{\varphi_n^a(b_{0:n})}{\sum_{r \in A} \varphi_n^r(b_{0:n})}$$

Verification:

$$\varphi_n^r(b_{0:n}) = \mathbb{P}(X_n = r, Y_{0:n} = b_{0:n})$$

$$\sum_{r \in A} \varphi_n^r(b_{0:n}) = \sum_{r \in A} \mathbb{P}(X_n = r, Y_{0:n} = b_{0:n})$$

$$= \mathbb{P}(X_n \in A, Y_{0:n} = b_{0:n})$$

Iterations

$\mathbb{P}(X_0 = a \mid Y_0 = b_0)\ \mathbb{P}(Y_0 = b)$

- Compute $\varphi_0^a(b_0) := \mathbb{P}(X_0 = a, Y_0 = b_0)$ for relevant non-zero probability outcomes $a \in A$.

- When observation b_1 is obtained, compute $\varphi_1^a(b_{0:1})$ for all relevant outcomes $a \in A$ using Algorithm 1 and the pre-computed values $\{\varphi_0^a(b_0)\}_a$.

- Similar iteration "$\{\varphi_n^r(b_{1:n})\}_r \mapsto \{\varphi_{n+1}^r(b_{1:n+1})\}_r$" for each $n \mapsto n+1$.

The iterations based on Alg 1 are called **online learning**, here meaning that you recursively update your estimate for every new observation.

An alternative would be **offline/batch learning**, here meaning to learn/precompute $\varphi_n^a(\tilde{b}_{0:n})$ for all relevant $n \geq 0$, $a \in A$ and $\tilde{b}_{0:n} \in B^{n+1}$ before filtering.

Remarks

- If instead of conditioning on the observation $b_{0:n}$, we condition on $Y_{0:n}$, then we get the following rv associated to filtering:

$$\mathbb{P}(X_n = a \mid Y_{0:n})(\omega) = \frac{\varphi_n^a(Y_{0:n}(\omega))}{\sum_{r \in A} \varphi_n^r(Y_{0:n}(\omega))}$$

- Extension of Alg 1 to when $\{Z_n\}$ is not a time-homogeneous Markov chain is straightforward. Replace time-independent transition functions by the time-dependent ones

$$q^{ra}(n, b_n, b_{n+1}) := p(n, (r, b_n), (a, b_{n+1}))$$

- Given a finite state-space A, we may view $\{q^{ra}(b_n, b_{n+1})\}_{(r,a) \in A^2}$ as a matrix q_n and $\{\varphi_n^a(b_{0:n})\}_a = \varphi_n$ as a column vector. The iterations in Alg 1 then becomes

$$\varphi_{n+1} = q_n^T \varphi_n.$$

Example 8 (Hidden Markov model)

Let X_n be a simple symmetric RW on $\mathbb{Z}$ and $Y_n = X_n + W_n$, where $\{W_n\}$ is iid and independent of $\{X_n\}$ with $\mathbb{P}(W_n = k) = 1/5$ for all $|k| \leq 2$. Assume $X_0 = 0$. **Compute** $\mathbb{P}(X_2 = 0 \mid Y_{0:2} = (0, 2, 1))$.

Some steps in the solution:

1. Identify transition function

$\{X_n = a\} \cap \{X_n + W_n = d\} = \{X_n = a\} \cap \{W_n = d-a\}$

$$q^{ra}(c, d) = \mathbb{P}\Big((X_n, Y_n) = (a, d) \mid (X_{n-1}, Y_{n-1}) = (r, c)\Big)$$
$$= \mathbb{P}(X_n = a, W_n = d-a \mid X_{n-1} = r, W_{n-1} = c-r)$$

$W_{n-1} \perp$ everything else same for W_n

$$= \mathbb{P}(X_n = a \mid X_{n-1} = r)\mathbb{P}(W_n = d - a)$$

(above eq holds $\mathbb{P}((X_{n-1}, Y_{n-1}) = (r, c)) > 0$).

2. Observe that $\{\varphi_n^a(b_{0:n})\}_a$ is only possibly non-zero for $a \in \{-n, -n+1, \ldots, n\}$

$\varphi_0^a(b_0) = \mathbb{P}(X_0 = a, Y_0 = b_0) = \mathbb{P}(X_0 = a) \cdot \mathbb{P}(W_0 = b_0 - a)$

3. Use Alg 1.

Overview

Prediction problem

The prediction problem is to estimate

$$\mathbb{P}(X_n = a \mid Y_{0:m} = b_{0:m})$$

for some $n > m \geq 0$.

Derivation of recursive algorithm:

$$\mathbb{P}(X_n = a \mid Y_{0:m} = b_{0:m}) = \frac{\mathbb{P}(X_n = a, Y_{0:m} = b_{0:m})}{\mathbb{P}(Y_{0:m} = b_{0:m})}$$
$$= \frac{\sum_{\bar{b}\in B} \mathbb{P}(X_n = a, Y_n = \bar{b}, Y_{0:m} = b_{0:m})}{\mathbb{P}(Y_{0:m} = b_{0:m})}$$

Idea for obtaining computable terms:

$$\mathbb{P}(X_n = a, Y_{0:n} = b_{0:m}) = \sum_{a_{0:n-1}\in A^n,\ b_{m+1:n}\in B^{n-m}} \mathbb{P}(X_n = a_{0:n}, Y_{0:n} = b_{0:n})$$

For $n \geq m$, introduce

$$\varphi_n^{a,\bar{b}}(b_{0:m}) := \mathbb{P}(X_n = a, Y_n = \bar{b}, Y_{0:m} = b_{0:m}).$$

Then, for $n = m$,

$$\varphi_n^{a,\bar{b}}(b_{0:m}) = \varphi_n^a(b_{0:m})\underline{\mathbb{1}_{b_m}(\bar{b})}$$

Verification

$$\varphi_m^{a,b}(b_{0:m}) = \mathbb{P}(X_m = a, Y_m = \bar{b}, Y_{0:m} = b_{0:m})$$

$$= \mathbb{P}(X_m = a, \underline{Y_m = \bar{b}, Y_m = b_m}, Y_{0:m-1} = b_{0:m-1})$$

And, for $n > m$,

$$\varphi_n^{a,\bar{b}}(b_{0:m}) = \sum_{r\in A, s\in B} \mathbb{P}(X_n = a, Y_n = \bar{b}, X_{n-1} = r, Y_{n-1} = s, Y_{0:m} = b_{0:m})$$

$$= \sum_{r,s} \mathbb{P}(X_n = a, Y_n = \bar{b} \mid X_{n-1} = r, Y_{n-1} = s)\, \mathbb{P}(X_{n-1} = r, Y_{n-1} = s, Y_{0:m} = b_{0:m})$$

$$= \sum_{r\in A, s\in B} \mathbb{P}(X_n = a, Y_n = \bar{b} \mid X_{n-1} = r, Y_{n-1} = s)\varphi_{n-1}^{r,s}(b_{0:m})$$

$$= \sum_{r\in A, s\in B} q^{ra}(s, \bar{b})\varphi_{n-1}^{r,s}(b_{0:m})$$

Summary

We seek a recursive algorithm for

$$\mathbb{P}(X_n = a \mid Y_{0:m} = b_{0:m}) = \frac{\sum_{\bar{b}\in B} \mathbb{P}(X_n = a, Y_n = \bar{b}, Y_{0:m} = b_{0:m})}{\mathbb{P}(Y_{0:m} = b_{0:m})}$$

$$= \frac{\sum_{\bar{b}\in B} \varphi_n^{a,\bar{b}}(b_{0:m})}{\sum_{r\in A} \varphi_m^r(b_{0:m})}$$

Every summand in the numerator satisfies recursive equation

$$\varphi_n^{a,\bar{b}}(b_{0:m}) = \sum_{r\in A, s\in B} q^{ra}(s, \bar{b})\varphi_{n-1}^{r,s}(b_{0:m}) \quad n > m \tag{8}$$

with "initial condition"

$$\varphi_m^{r,s}(b_{0:m}) = \varphi_m^r(b_{0:m})\mathbb{1}_{b_m}(s).$$

Algorithm 2 – Prediction

1. Compute $\{\varphi_m^r(b_{0:m})\}_{r\in A}$ by Algorithm 1.
2. "Initialize" $\varphi_m^{r,s}(b_{0:m}) = \varphi_m^r(b_{0:m})\mathbb{1}_{b_m}(s)$ for relevant $(r,s)\in A\times B$.
3. Compute $\{\varphi_k^{r,s}(b_{0:m})\}_{r\in A, s\in B}$ for $k = m+1, m+2, \ldots, n-1$ by the recursive formula (8).
4. Compute $\{\varphi_n^{a,\bar{b}}(b_{0:n})\}_{\bar{b}\in B}$ by (8) (i.e., for the fixed state $a\in A$ only).
5. Output:

$$\mathbb{P}(X_n = a \mid Y_{0:m} = b_{0:m}) = \frac{\sum_{\bar{b}\in B}\varphi_n^{a,\bar{b}}(b_{0:m})}{\sum_{r\in A}\varphi_m^r(b_{0:m})}.$$

Remark: Algorithm 2 simplifies in many settings, e.g., hidden Markov models [FJK 3.4.3].

cost naive: $|A|^n \cdot |B|^{n-m}$

recursive : $m|A| + (n-m)|A||B|$

Exercise Simplify the "recursive" equation predictions when $n = m+1$.

Overview

Smoothing problem

The smoothing/interpolation problem is to estimate

$$\mathbb{P}(X_n = a \mid Y_{0:m} = b_{0:m})$$

for some $m > n \geq 0$.

Property: More information leads to improved approximations: for $m > n > k \geq 0$

$$\underbrace{\mathbb{E}\left[\,|X - \mathbb{E}\left[X_n|Y_{0:m}\right]|^2\right]}_{\text{smoothing}} \leq \underbrace{\mathbb{E}\left[\,|X - \mathbb{E}\left[X_n|Y_{0:n}\right]|^2\right]}_{\text{filtering}} \leq \underbrace{\mathbb{E}\left[\,|X - \mathbb{E}\left[X_n|Y_{0:k}\right]|^2\right]}_{\text{prediction}}$$

$$\leq \mathbb{E}\left[\,|X - g(Y_{0:m})|^2\right]$$

for any $g(Y_{0:m})$

Derivation of a recursive algorithm:

$$\mathbb{P}(X_n = a \mid Y_{0:m} = b_{0:m}) = \frac{\mathbb{P}(X_n = a, Y_{0:m} = b_{0:m})}{\mathbb{P}(Y_{0:m} = b_{0:m})} = \frac{\varphi_n^a(b_{0:m})}{\sum_{r \in A} \varphi_m^r(b_{0:m})}$$

Using the Markov property [FJK 2.2.7]

$$\{Y_{n+1:m} = b_{n+1:m}\}$$
$$= \{(X_{n+1:m}, Y_{n+1:m}) \in A^{m-n+1} \times \{b_{n+1:m}\}\}$$

$$\begin{aligned}
\varphi_n^a(b_{0:m}) &= \mathbb{P}\left(X_n = a, Y_{0:n} = b_{0:n}, Y_{n+1:m} = b_{n+1:m}\right) \\
&= \mathbb{P}\left(Y_{n+1:m} = b_{n+1:m} \mid X_n = a, Y_{0:n} = b_{0:n}\right) \mathbb{P}\left(X_n = a, Y_{0:n} = b_{0:n}\right) \\
&= \mathbb{P}\left(Y_{n+1:m} = b_{n+1:m} \mid X_n = a, Y_n = b_n\right) \varphi_n^a(b_{0:n})
\end{aligned} \tag{9}$$

Next seek to obtain recursive formula for first factor when $\mathbb{P}(X_n = a, Y_n = b_n) > 0$.

Otherwise, also $\varphi_n^a(b_{0:n}) = 0$, and the value of the first-factor value is not needed.

By the law of total probability,

$$
\begin{aligned}
&\mathbb{P}\left(Y_{n+1:m} = b_{n+1:m} \mid X_n = a, Y_n = b_n\right) \\
&= \sum_{\bar{a}_{n+1:m} \in A^{m-n}} \mathbb{P}\left(X_{n+1:m} = \bar{a}_{n+1:m}, Y_{n+1:m} = b_{n+1:m} \mid X_n = a, Y_n = b_n\right) \\
&= \sum_{\bar{a}_{n+1:m} \in A^{m-n}} p\left((a, b_n), (\bar{a}_{n+1}, b_{n+1})\right) p\left((\bar{a}_{n+1}, b_{n+1}), (\bar{a}_{n+2}, b_{n+2})\right) \dots \\
&\qquad\qquad \dots p\left((\bar{a}_{m-1}, b_{m-1}), (\bar{a}_m, b_m)\right)
\end{aligned}
$$

Algorithm for κ [FJK problem 3.3.4]

Whenever $\mathbb{P}(X_n = a, Y_n = b_n) > 0$

$$
\kappa^a_{n,m}(b_{n:m}) := \begin{cases} \mathbb{P}\left(Y_{n+1:m} = b_{n+1:m} \mid X_n = a, Y_n = b_n\right) & \text{if } n < m \\ 1 & \text{if } n = m \end{cases}
$$

solves the following backward recurrence equation

$$
\kappa^a_{n-1,m}(b_{n-1:m}) = \sum_{r \in A} \underbrace{p((a, b_{n-1}), (r, b_n))}_{q^{ar}(b_{n-1}, b_n)} \kappa^r_{n,m}(b_{n:m}), \qquad n = 1, 2, \dots, m.
$$

Summary

We seek

$$\mathbb{P}(X_n = a \mid Y_{0:m} = b_{0:m}) = \frac{\mathbb{P}(X_n = a, Y_{0:m} = b_{0:m})}{\mathbb{P}(Y_{0:m} = b_{0:m})} = \frac{\varphi_n^a(b_{0:m})}{\sum_{r \in A} \varphi_m^r(b_{0:m})} \tag{10}$$

and by (9) and Algorithm for κ,

$$\varphi_n^a(b_{0:m}) = \kappa_{n,m}^a(b_{n:m})\varphi_n^a(b_{0:n}) \qquad (11)$$

Algorithm 3 - smoothing/interpolation

1. Compute $\{\varphi_m^r(b_{0:m})\}_r$ by Algorithm 1,
2. Compute $\kappa_{n,m}^a(b_{n:m})$ by Algorithm for κ (11), and the output (10).

Next time

Continuous random variables, probability density functions, conditional densities . . .