Hidden Markov models

FORWARD UPDATE

$$\alpha_t(j) = e_t(j) \sum_i \alpha_{t-1}(i) T_{ij}$$

COMPLETE EXAMPLE

Two states, three readings

```
flatppl_compat = "0.1"
pi0 = [0.6, 0.4]
T = rowstack([[0.8, 0.2], [0.3, 0.7]])
p = [0.2, 0.8]
y = [0, 1, 1]
E = colstack([1 .- p, p])
L = log.(E)
H = log.(T)
advance(a, M) = [
    logsumexp(a .+ M[:, 1]),
    logsumexp(a .+ M[:, 2])]
forward(a, v, q) = (
    advance(a, q.M) .+ q.L[:, v + 1])
backward(b, v, q) = (
    advance(b .+ q.L[:, v + 1], q.M))

n = lengthof(y)
rest = y[indicesof(zeros(n - 1)) .+ 1]
a1 = log.(pi0) .+ L[:, y[1] + 1]
fwd = scan(forward, a1, rest, (M = H, L = L))
A = cat([a1], fwd)

bn = zeros(2)
bwd = scan(backward, bn, reverse(rest),
    (M = transpose(H), L = L))
B = reverse(cat([bn], bwd))
filt = rowstack(softmax.(A))
smooth_at(t) = softmax(A[t] .+ B[t])
smooth = rowstack(smooth_at.(indicesof(y)))
ll = logsumexp(A[n])
pred = transpose(T) * filt[n, :]
next1 = sum(pred .* p)

e = E[:, y[[1, 2, 3]] .+ 1]
path[.i, .j, .k] := (
    pi0[.i] * e[.i, 1] * T[.i, .j] *
    e[.j, 2] * T[.j, .k] * e[.k, 3])
exact = sum(path)
outputs = record(ll = ll, exact = exact,
    path = path, filt = filt, smooth = smooth,
    pred = pred, next1 = next1)
```

Change the reading sequence

Append readings to `y`. The scans use every reading. The path check uses the first three. With `p = [0.5, 0.5]`, any three readings have likelihood $\frac{1}{8}$.

ANNOTATED LINES

A Combine transitions in log space 9–11

```
advance(a, M) = [
    logsumexp(a .+ M[:, 1]),
    logsumexp(a .+ M[:, 2])]
```

State j emits a 1 with probability p_j. The columns of `E` give the probabilities of readings 0 and 1. Select column `v + 1` for reading `v`. Each column of `H` gives the log transitions into one state. Combine the incoming scores with `logsumexp`.

B Scan forward over the readings 19–21

```
a1 = log.(pi0) .+ L[:, y[1] + 1]
fwd = scan(forward, a1, rest, (M = H, L = L))
A = cat([a1], fwd)
```

Seed the scan with $\log \alpha_1(j) = \log(\pi_0(j) e_1(j))$. `rest` holds readings 2 through `n`. Each scan step updates the state vector. The result excludes the seed, so prepend it with `cat`. The final vector gives `ll = logsumexp(A[n])`.

C Scan backward over the readings 23–26

```
bn = zeros(2)
bwd = scan(backward, bn, reverse(rest),
    (M = transpose(H), L = L))
B = reverse(cat([bn], bwd))
```

Start with zero log future mass at the final time. Read `rest` in reverse. The callback applies the emission before the transition. Transpose `H` to sum over destinations. Restore time order in `B` and apply `softmax` to `A[t] .+ B[t]`.

D Enumerate the eight paths 34–38

```
e = E[:, y[[1, 2, 3]] .+ 1]
path[.i, .j, .k] := (
    pi0[.i] * e[.i, 1] * T[.i, .j] *
    e[.j, 2] * T[.j, .k] * e[.k, 3])
exact = sum(path)
```

The axes `.i`, `.j`, and `.k` select states at the first three times. Retain all three to get a $2 \times 2 \times 2$ array of joint path masses. Their sum matches `exp(ll)` for the three readings shown.

What the scans carry

The fourth argument passes the matrices as one record to each callback. With K states and dense transitions, each update costs $O(K^2)$. The two scans give the full state history.

LIKELIHOOD AND STATE ESTIMATES

All eight paths

Path	Joint mass	Path	Joint mass
111	0.012288	211	0.000768
112	0.012288	212	0.000768
121	0.004608	221	0.002688
122	0.043008	222	0.025088

Each mass is joint with the observed sequence. For path 122, multiply $0.6 \cdot 0.8 \cdot 0.2 \cdot 0.8 \cdot 0.7 \cdot 0.8 = 0.043008$. The eight masses sum to 0.101504, so `ll` is −2.287657.

Filter or smooth state 2

Apply `softmax` to the forward log scores to use readings through t. Smoothing uses all three readings. The two estimates agree at $t = 3$.

Time	Reading	Filtered	Smoothed
1	0	0.142857	0.288777
2	1	0.598425	0.742749
3	1	0.799496	0.799496

Forecast one step ahead

Left-multiply the last filtered state vector by $T^\top$ to get `pred`. Its second component is 0.599748. The next reading is a 1 with probability $\sum_j p_j \mathrm{pred}_j \approx 0.559849$.

Infer a path or an event

The most probable path is 122. Its posterior mass is $\frac{0.043008}{0.101504} \approx 0.423707$. For a switch between readings 1 and 2, sum paths 121, 122, 211, and 212 before dividing by 0.101504. The switch probability is 0.484237.

Check one forward update

`T[1, :]` predicts state 2 with probability 0.2 from state 1. `T[:, 2]` lists both ways to enter state 2. The incoming mass at time 2 is $0.48 \cdot 0.2 + 0.08 \cdot 0.7$. Multiply by the emission probability 0.8 to get $\alpha_2(2) = 0.1216$.

Track the evidence after each reading

Sum the two forward masses at each time. The ratios of consecutive sums give the probability of each new reading given the preceding ones.

Through time	1	2	3
Joint probability	0.56	0.2032	0.101504
Conditional	0.56	0.362857	0.499528

Jessica Cox · bcox@mpp.mpg.de