Some worked eg's of Parameter Estimation with HMMs: a fully
observed case and the 'brute-force' EM unsupervised case

Martin Emms

November 13, 2019

In the following there is a worked example of an iteration of brute-force EM for
the HMM case.
It all follows the general pattern of EM seen earlier in the hidden coin scenario,
and the hidden alignment scenario.
Nonetheless it may help to see some of the specifics of what precisely has to be
'counted' in the HMM case

A fully observed (or 'supervised') case of estimation
first suppose a fully observed corpus of obs and state sequences (states in blue,
obsvs in red):

one:a three:x four:z five:#  (= o^0, s^0)
   one:a three:y three:x five:#  (= o^1, s^1)
   two:b four:x one:a three:y five:#  (= o^2, s^2)

The values for π, A and B derived by relative frequencies in this corpus are:

\[
\begin{align*}
A \quad & C(\text{one, three}) = 3 & C(\text{one}) = 3 & P(\text{three} \mid \text{one}) = 1 \\
& C(\text{three, four}) = 1 & C(\text{three}) = 4 & P(\text{four} \mid \text{three}) = 0.25 \\
& C(\text{three, five}) = 2 & & P(\text{five} \mid \text{three}) = 0.5 \\
& C(\text{two, four}) = 1 & C(\text{two}) = 1 & P(\text{four} \mid \text{two}) = 1 \\
& C(\text{four, five}) = 1 & C(\text{four}) = 2 & P(\text{five} \mid \text{four}) = 0.5 \\
& C(\text{four, one}) = 1 & & P(\text{one} \mid \text{four}) = 0.5 \\
B \quad & C(\text{one, a}) = 3 & C(\text{one}) = 3 & P(\text{a} \mid \text{one}) = 1 \\
& C(\text{three, x}) = 2 & C(\text{three}) = 4 & P(\text{x} \mid \text{three}) = 0.5 \\
& C(\text{three, y}) = 2 & & P(\text{y} \mid \text{three}) = 0.5 \\
& C(\text{two, b}) = 1 & C(\text{two}) = 1 & P(\text{b} \mid \text{two}) = 1 \\
& C(\text{five, #}) = 3 & C(\text{five}) = 3 & P(\# \mid \text{five}) = 1 \\
& C(\text{four, z}) = 1 & C(\text{four}) = 2 & P(\text{z} \mid \text{four}) = 0.5 \\
& C(\text{four, x}) = 1 & & P(\text{x} \mid \text{four}) = 0.5 \\
\pi \quad & C(s_1 = \text{one}) = 2 & P(s_1 = \text{one}) = 0.666 \\
& C(s_1 = \text{two}) = 1 & P(s_1 = \text{two}) = 0.333
\end{align*}
\]
Some worked eg of Parameter Estimation with HMMs: a fully observed case and the 'brute-force' EM unsupervised case

Outline

brute-force EM: conditional probs of paths

To apply (brute-force) EM, for each $o^d$, need for all possible $s$, $p(s|o^d)$ (ie. the 'responsibility' $\gamma^d(s)$). Given the forgoing we get:

- $o^0: a \times z$ 
  $s^0: one three four five$
  $$P(s^0|o^0) = 1 \quad (= \gamma^0(s^0))$$

- $o^1: a y \times$
  $s^1: one three four five$
  $$P(s^1|o^1) = 0.5 \quad (= \gamma^1(s^1))$$

- $o^2: b \times a y$
  $s^2: two one three five$
  $$P(s^2|o^2) = 0.5 \quad (= \gamma^2(s^2))$$

- $o^{11}: b x a y$
  $s^{11}: two four one three five$
  $$P(s^{11}|o^{11}) = 1 \quad (= \gamma^{11}(s^{11}))$$

brute-force EM: considering all the possible paths

On our initial settings what state sequences are possible for these observation sequences?

<table>
<thead>
<tr>
<th>observations</th>
<th>possible completions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o^0: a \times z$</td>
<td>one:a three:x four:z five:# $=$ $o^0$, $s^0$</td>
</tr>
<tr>
<td>$o^1: a y \times$</td>
<td>one:a three:y four:x five:# $=$ $o^1$, $s^1$</td>
</tr>
<tr>
<td>$o^2: b \times a y$</td>
<td>two:b four:x one:a three:y five:# $=$ $o^2$, $s^2$</td>
</tr>
</tbody>
</table>

Dito for $o^{11}$.

$o^0$ and the 10 copies of $o^2$ get actually get only 1 possible completion each$^1$.

though gets two possible completions, equally likely ones.

is because when you look to the terms that differ between $P(o^1, s^1)$ and $P(o^1, s^2)$, numerically there's no difference:

\[
P(\text{four}|\text{three}) = 0.25 \quad \text{P}(\text{four}|\text{four}) = 0.5 \quad \text{P}(\text{five}|\text{four}) = 0.5 \\
P(\text{three}|\text{three}) = 0.25 \quad \text{P}(\text{three}|\text{four}) = 0.5 \quad \text{P}(\text{five}|\text{four}) = 0.5 
\]

$^1$the 'hidden' states are not really very hidden!

A and B: expected counts and ratios

will compute expected counts of events relevant for A and B based on the virtual corpus of completions:

<table>
<thead>
<tr>
<th>observations</th>
<th>'count' of completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o^0: a \times z$</td>
<td>1× one:a three:x four:z five:#</td>
</tr>
<tr>
<td>$o^1: a y \times$</td>
<td>0.5× one:a three:y four:x five:#</td>
</tr>
<tr>
<td>$o^2: b \times a y$</td>
<td>10× two:b four:x one:a three:y five:#</td>
</tr>
</tbody>
</table>

Note although $x$ can be generated from four or three, in $o^2$ only four is possible, so $four:x$ is more frequent in this virtual corpus than it was in the fully observed corpus we started with.

We derive expected counts for 'state-then-state' and 'obs-with-state', then derive new A and B, eg.

\[
A^1 \quad E(\text{three, four}) = 1.5 \quad E(\text{three}) = 12.5 \quad P(\text{four}|\text{three}) = 1.5/12.5 \\
E(\text{three, three}) = 0.5 \quad \text{P}(\text{three}|\text{three}) = 0.5/12.5 \\
E(\text{three, five}) = 10.5 \quad \text{P}(\text{five}|\text{three}) = 10.5/12.5 \\
E(\text{four, five}) = 1.5 \quad E(\text{four}) = 11.5 \quad \text{P}(\text{five}|\text{four}) = 1.5/11.5 \\
E(\text{four, one}) = 10 \\
B^1 \quad E(\text{three, x}) = 1.5 \quad E(\text{three}) = 12.5 \quad P(\text{x}|\text{three}) = 1.5/12.5 \\
E(\text{three, y}) = 11 \quad E(\text{four, x}) = 1.5 \quad E(\text{four}) = 11.5 \quad \text{P}(\text{x}|\text{four}) = 10.5/11.5 \\
E(\text{four, z}) = 10.5 \quad E(\text{four, x}) = 1.5 \quad E(\text{four}) = 11.5 \quad \text{P}(\text{x}|\text{four}) = 10.5/11.5 
\]

Note that this estimation from a small set of examples has left quite a few options set to have zero probability. This will be handy in the following illustration of the unsupervised case as it keeps the number of alternatives down. In general such zeroes would be a bad idea. Let's use these probs as initial values in some unsupervised estimation, just using a corpus of observed words. Let's suppose this corpus to have much more of the final obs sequence than before:

1× $a x z #$
1× $a y x #$
10× $b x a y #$

Some worked eg of Parameter Estimation with HMMs: a fully observed case and the 'brute-force' EM unsupervised case
the virtual corpus of completions:

<table>
<thead>
<tr>
<th>observations</th>
<th>'count' of completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>o1</td>
<td>a x z #</td>
</tr>
<tr>
<td></td>
<td>1 × one : a three : x four : z five : #</td>
</tr>
<tr>
<td>o2</td>
<td>a y x #</td>
</tr>
<tr>
<td></td>
<td>0.5 × one : a three : y four : x five : #</td>
</tr>
<tr>
<td></td>
<td>0.5 × one : a three : y three : x five : #</td>
</tr>
<tr>
<td>o2</td>
<td>b x a y #</td>
</tr>
<tr>
<td></td>
<td>10 × one : two : b four : x one : a three : y five : #</td>
</tr>
</tbody>
</table>

derive expected counts of 'state at start', then new π:

\[
\begin{align*}
\pi^1 & \quad E(s_1 = \text{one}) = 2 & P(s_1 = \text{one}) = 2/12 \\
\pi^1 & \quad E(s_1 = \text{two}) = 10 & P(s_1 = \text{two}) = 10/12 \\
A^1 & \quad P(\text{start} = \text{one}) = 2/12 & P(\text{start} = \text{two}) = 10/12 \\
B^1 & \quad P(\text{four} | \text{three}) = 1.5/12.5 & P(\text{four} | \text{one}) = 10.5/11.5 \\
B^1 & \quad P(\text{five} | \text{three}) = 10.5/12.5 & P(\text{five} | \text{four}) = 1.5/11.5 \\
\end{align*}
\]

For example, concerning a y x #, whereas previously there were two state sequences that could pair equally well with this, now there is an asymmetry.

\[
\begin{align*}
P(\text{one : a three : four : x five : #}) & = 1.5/12.5 \times 10.5/11.5 \times 1.5/11.5 = 0.0143 \\
P(\text{one : a three : y three : x five : #}) & = 0.5/12.5 \times 1.5/12.5 \times 10.5/12.5 = 0.0040 \\
\end{align*}
\]

So one round of EM has definitely changed the probabilities; really all this should be every expectable given previous study of EM applied in (i) the coin tossing scenario – hidden A vs B choice (ii) the IBM Model 1 scenario – hidden alignment. We ‘just’ have a different hidden variable here – the hidden state sequence. As the main slides expound upon, the bigger point is that this brute-force approach is not computationally feasible.