SFU NatLangLab

CMPT-413: Computational Linguistics

HMM4: Viterbi algorithm for Hidden Markov Models

Anoop Sarkar
http://www.cs.sfu.ca/∼anoop

February 28, 2013

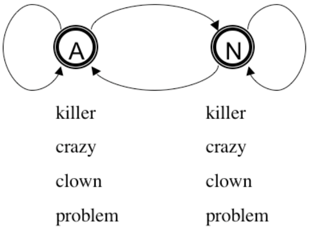
Viterbi Algorithm for HMMs

- For input of length T: $o_1, \ldots, o_T$, we want to find the sequence of states $s_1, \ldots, s_T$
- Each s_t in this sequence is one of the states in the HMM.
- So the task is to find the most likely sequence of states:

$$\underset{s_1,\ldots,s_T}{\operatorname{argmax}} P(o_1, \ldots, o_T, s_1, \ldots, s_T)$$

- The Viterbi algorithm solves this by creating a table $V[s, t]$ where s is one of the states, and t is an index between $1, \ldots, T$.

Viterbi Algorithm for HMMs



- Consider the input *killer crazy clown problem*
- So the task is to find the most likely sequence of states:

$$\underset{s_1, s_2, s_3, s_4}{\operatorname{argmax}} P(\textit{killer crazy clown problem}, s_1, s_2, s_3, s_4)$$

- A sub-problem is to find the most likely sequence of states for *killer crazy clown*:

$$\underset{s_1, s_2, s_3}{\operatorname{argmax}} P(\textit{killer crazy clown}, s_1, s_2, s_3)$$

Viterbi Algorithm for HMMs

- In our example there are two possible values for s_4:

$$\max_{s_1,\ldots,s_4} P(\mathit{killer\ crazy\ clown\ problem}, s_1, s_2, s_3, s_4) =$$

$$\max\left\{\max_{s_1,s_2,s_3} P(\mathit{killer\ crazy\ clown\ problem}, s_1, s_2, s_3, N),\right.$$

$$\left.\max_{s_1,s_2,s_3} P(\mathit{killer\ crazy\ clown\ problem}, s_1, s_2, s_3, A)\right\}$$

- Similarly:

$$\max_{s_1,\ldots,s_3} P(\mathit{killer\ crazy\ clown}, s_1, s_2, s_3) =$$

$$\max\left\{\max_{s_1,s_2} P(\mathit{killer\ crazy\ clown}, s_1, s_2, N),\right.$$

$$\left.\max_{s_1,s_2} P(\mathit{killer\ crazy\ clown}, s_1, s_2, A)\right\}$$

Viterbi Algorithm for HMMs

- Putting them together:

$$P(\textit{killer crazy clown problem}, s_1, s_2, s_3, N) =$$
$$\max\{P(\textit{killer crazy clown}, s_1, s_2, N) \cdot a_{N,N} \cdot b_N(\textit{problem}),$$
$$P(\textit{killer crazy clown}, s_1, s_2, A) \cdot a_{A,N} \cdot b_N(\textit{problem})\}$$

$$P(\textit{killer crazy clown problem}, s_1, s_2, s_3, A) =$$
$$\max\{P(\textit{killer crazy clown}, s_1, s_2, N) \cdot a_{N,A} \cdot b_A(\textit{problem}),$$
$$P(\textit{killer crazy clown}, s_1, s_2, A) \cdot a_{A,A} \cdot b_A(\textit{problem})\}$$

- The best score is given by:

$$\max_{s_1,\ldots,s_4} P(\textit{killer crazy clown problem}, s_1, s_2, s_3, s_4) =$$
$$\max_{N,A}\left\{\max_{s_1,s_2,s_3} P(\textit{killer crazy clown problem}, s_1, s_2, s_3, N),\right.$$
$$\left.\max_{s_1,s_2,s_3} P(\textit{killer crazy clown problem}, s_1, s_2, s_3, A)\right\}$$

Viterbi Algorithm for HMMs

- Provide an index for each input symbol:
  1:killer 2:crazy 3:clown 4:problem

$$
\begin{aligned}
V[N,3] &= \max_{s_1,s_2} P(\textit{killer crazy clown}, s_1, s_2, N) \\
V[N,4] &= \max_{s_1,s_2,s_3} P(\textit{killer crazy clown problem}, s_1, s_2, s_3, N)
\end{aligned}
$$

- Putting them together:

$$
\begin{aligned}
V[N,4] &= \max\{V[N,3] \cdot a_{N,N} \cdot b_N(\textit{problem}), \\
&\qquad V[A,3] \cdot a_{A,N} \cdot b_N(\textit{problem})\}
\end{aligned}
$$

$$
\begin{aligned}
V[A,4] &= \max\{V[N,3] \cdot a_{N,A} \cdot b_A(\textit{problem}), \\
&\qquad V[A,3] \cdot a_{A,A} \cdot b_A(\textit{problem})\}
\end{aligned}
$$

- The best score for the input is given by:
  $\max\{V[N,4], V[A,4]\}$
- To extract the best sequence of states we backtrack (same trick as obtaining alignments from minimum edit distance)

Viterbi Algorithm for HMMs

- For input of length T: $o_1, \ldots, o_T$, we want to find the sequence of states $s_1, \ldots, s_T$
- Each s_t in this sequence is one of the states in the HMM.
- For each state q we initialize our table: $V[q, 1] = \pi_q \cdot b_q(o_1)$
- Then compute for $t = 1 \ldots T - 1$ for each state q:

$$V[q, t+1] = \max_{q'} \left\{ V[q', t] \cdot a_{q',q} \cdot b_q(o_{t+1}) \right\}$$

- After the loop terminates, the best score is $\max_q V[q, T]$

Learning from Fully Observed Data

$\pi =$

A	0.25
N	0.75

$a =$

$a_{i,j}$	A	N
A	0.0	1.0
N	0.5	0.5

$b =$

$b_i(o)$	*clown*	*killer*	*problem*	*crazy*
A	0	0	0	1
N	0.4	0.3	0.3	0

Viterbi algorithm:

V	killer:1	crazy:2	clown:3	problem:4
A				
N				

Learning from Fully Observed Data

$\pi =$

A	0.25
N	0.75

$a =$

$a_{i,j}$	A	N
A	0.0	1.0
N	0.5	0.5

$b =$

$b_i(o)$	*clown*	*killer*	*problem*	*crazy*
A	0	0	0	1
N	0.4	0.3	0.3	0

Viterbi algorithm:

V	killer:1	crazy:2	clown:3	problem:4
A	0	0.1125	0	0
N	0.225	0	0.045	0.00675