Hidden Markov Models

L645 / B659
Dept. of Linguistics, Indiana University
Fall 2015
A Markov Model consists of:

- a finite set of states $\Omega = \{s_1, \ldots, s_n\}$;
- an signal alphabet $\Sigma = \{\sigma_1, \ldots, \sigma_m\}$;
- an $n \times n$ state transition matrix $P = [p_{ij}]$ where $p_{ij} = P(\xi_{t+1} = s_j | \xi_t = s_i)$;
- an $n \times m$ signal matrix $A = [a_{ij}]$, which for each state-signal pair determines the probability $a_{ij} = p(\eta_t = \sigma_j | \xi_t = s_i)$ that signal $\sigma_j$ will be emitted given that the current state is $s_i$;
- and an initial vector $v = [v_1, \ldots, v_n]$ where $v_i = P(\xi_1 = s_i)$. 
HMM Example 1: Crazy Softdrink Machine

with emission probabilities:

\[
\begin{array}{ccc}
\text{cola} & \text{ice-t} & \text{lem} \\
CP & 0.6 & 0.1 & 0.3 \\
IP & 0.1 & 0.7 & 0.2 \\
\end{array}
\]

from: Manning & Schütze, p. 321
Markov Models (2)
(Review)

\[ p^{(t)}(s_i, \sigma_j) = p^{(t)}(s_i) \cdot p(\eta_t = \sigma_j \mid \xi_t = s_i) \]

where \( p^{(t)}(s_i) \) is the \( i \)th element of the vector \( \mathbf{vP}^{t-1} \). The probability that signal \( \sigma_j \) will be emitted at time \( t \) is then:

\[ p^{(t)}(\sigma_j) = \sum_{i=1}^{n} p^{(t)}(s_i, \sigma_j) = \sum_{i=1}^{n} p^{(t)}(s_i) \cdot p(\eta_t = \sigma_j \mid \xi_t = s_i) \]

Thus if \( p^{(t)}(\sigma_j) \) is the probability of the model emitting signal \( \sigma_j \) at time \( t \), i.e., after \( t - 1 \) steps, then

\[ [p^{(t)}(\sigma_1), \ldots, p^{(t)}(\sigma_m)] = \mathbf{vP}^{t-1} \mathbf{A} \]
Let $O \in \Sigma^*$ be a known sequence of observed signals and $S \in \Omega^*$ the sequence of states in which $O$ is emitted.

If it is not possible to observe the sequence of states $S_1, \ldots, S_T$ of a Markov model, but only the sequence $n_1, \ldots, n_T$, the model is called a hidden Markov model (an HMM) (Krenn/Samuelsson, p. 43)

In an HMM, you don’t know the state sequence that the model passes through, but only some probabilistic function of it (Manning/Schütze, p. 320)

Our best guess at $S$ is the sequence maximizing

$$\max_S P(S \mid O)$$
Hidden Markov Models (2)

Prototypical tasks to which hidden Markov models are applied include the following . . .

Given a sequence of signals $\mathbf{O} = (\sigma_{i_1}, \ldots, \sigma_{i_T})$:

- Estimate the probability of observing this particular signal sequence.
  - e.g., language identification

- Determine the most probable state sequence that can give rise to this signal sequence.
  - e.g., POS tagging & speech recognition

- Determine the set of model parameters $\lambda = (\mathbf{P}, \mathbf{A}, \mathbf{v})$ maximizing the probability of this signal sequence.
Hidden Markov Models (3)

2 types of representations:

- **State emission HMM**
  - The symbol emitted at time $t$ depends only on the state at time $t$

- **Arc emission HMM**
  - The symbol emitted at time $t$ depends both on states at time $t$ and time $t + 1$

- Manning & Schütze do arc emission; Krenn & Samuelsson do state emission
  - We will do state emission
HMM Application 1: POS tagging

- The set of observable signals are the words of an input text.
- The states are the set of tags that are to be assigned to the words of input text.
- The task consists in finding the most probable sequence of states that explains the observed words.
  - This will assign a particular state to each signal, i.e., a tag to each word.
### Example HMM

Assume that we have DET, N, and VB as our hidden states, and we have the following transition matrix (A):

<table>
<thead>
<tr>
<th></th>
<th>DET</th>
<th>N</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>DET</td>
<td>0.01</td>
<td>0.89</td>
<td>0.10</td>
</tr>
<tr>
<td>N</td>
<td>0.30</td>
<td>0.20</td>
<td>0.50</td>
</tr>
<tr>
<td>VB</td>
<td>0.67</td>
<td>0.23</td>
<td>0.10</td>
</tr>
</tbody>
</table>

... emission matrix (B):

<table>
<thead>
<tr>
<th></th>
<th>dogs</th>
<th>bit</th>
<th>the</th>
<th>chased</th>
<th>a</th>
<th>these</th>
<th>cats</th>
</tr>
</thead>
<tbody>
<tr>
<td>DET</td>
<td>0.0</td>
<td>0.0</td>
<td>0.33</td>
<td>0.0</td>
<td>0.33</td>
<td>0.33</td>
<td>0.0</td>
</tr>
<tr>
<td>N</td>
<td>0.2</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.15</td>
</tr>
<tr>
<td>VB</td>
<td>0.1</td>
<td>0.6</td>
<td>0.0</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

... and initial probability matrix ($\pi$):

<table>
<thead>
<tr>
<th></th>
<th>DET</th>
<th>N</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>DET</td>
<td>0.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VB</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
State sequences

If we generate *the bit dogs*, we don’t know which tag sequence generated it:

- DET N VB?
- DET N N?
- DET VB N?
- DET VB VB?

Each has different probabilities:

- Need an algorithm to give the best sequence of states (i.e., tags) for a given sequence of words
HMM Application 2: Speech recognition

- The set of observable signals are (some representation of the) acoustic signals.
- The states are the possible words that these signals could arise from.
- The task consists in finding the most probable sequence of words that explains the observed acoustic signals.
  - This is a slightly more complicated situation, since the acoustic signals do not stand in a one-to-one correspondence with the words.
Three Fundamental Problems for HMMs (1)

- **Calculating the Probability of an Observation Sequence:**
  
  Given the observation sequence $O = O_1 O_2 \ldots O_T$, and a model $\lambda = \langle P, A, \nu \rangle$, how to (efficiently) compute the probability of the observation sequence, given the model: $P(O \mid \lambda)$

- **Method of Choice:**
  
  Trellis algorithm using forward or backward accumulator variables.
Three Fundamental Problems for HMMs (2)

- Finding the Optimal State Sequence:
  Given the observation sequence $O = O_1 O_2 \ldots O_T$ and the model $\lambda$, how do we choose the most likely state sequence that corresponds to $O$:
  \[ \max_S P(S \mid O) \]

- Method of Choice:
  Viterbi-style dynamic programming algorithm using a trellis and backpointers
Three Fundamental Problems for HMMs (3)

▶ Parameter Estimation:
  How to estimate the model parameters 
  \( \lambda = \langle P, A, \nu \rangle \) to maximize \( P(O | \lambda) \)

▶ Method of Choice:
  Baum-Welch algorithm using forward-backward re-estimation