Introduction To Hidden Markov Models

Yoshiko Ito
Dragon Systems
Review of Markov Chains

- States - $S_1, S_2, ...$
- State transitions and associated transition probabilities $a_{ij}$

Let $q_t =$ state at time $t$

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i)$$

Markov property:

$$P(q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, ...) = P(q_t = S_j | q_{t-1} = S_i)$$
Hidden Markov Model (HMM)

- States
- State transitions & transition probabilities
- Output (observation) & output probabilities
  \[ b_i(o_t) = P(\text{observing } o_t \text{ at time } t \mid q_t = S_i) \]

- States are "hidden", i.e., they are only indirectly observable through observation sequences.
- \( b_i(o_t) \) depends only on \( q_t \) and \( o_t \)
HMM Example ("Balls and Urns")
Each urn contains colored balls
Each ball is colored RED, BLUE, GREEN, or YELLOW.

\[ P(R|U_{i}) = b_{1}(R) = 0.7 \]
\[ b_{1}(B) = 0.2 \]
\[ b_{1}(G) = 0.05 \]
\[ b_{1}(Y) = 0.05 \]

\[ b_{2}(R) = 0.2 \]
\[ b_{2}(B) = 0.6 \]
\[ b_{2}(G) = 0.1 \]
\[ b_{2}(Y) = 0.1 \]

\[ b_{3}(R) = 0.1 \]
\[ b_{3}(B) = 0.3 \]
\[ b_{3}(G) = 0.1 \]
\[ b_{3}(Y) = 0.5 \]

- Pick an urn according to a Markov process
  \[ \alpha_{i,j} = P(\text{picking urn } j \mid \text{last urn picked } = i ) \]
- Pick a ball at random from the selected urn.
- The observer only sees the color of the selected ball.

State = urn, Observation = color of the selected ball
Three HMM Problems

1. Given an HMM and an observation sequence, find the probability that the observed sequence was generated by the model.

   Application: Recognition/Classification

2. Given an HMM and an observation sequence, find the best single state sequence that accounts for the observation.

   Application: Recognition, Segmentation

3. Given an HMM and an observation sequence, choose the model parameters so that the probability of observation is maximized.

   Application: Training
1. Given an HMM and an observation sequence find the probability that the observed sequence was generated by the model.

Let

\[ O_1, O_2, \ldots, O_T = \text{observation sequence} \]
\[ q_t = \text{state at time } t ; q_t \in \{ s_1, s_2, \ldots, s_N \} \]
\[ X_{q_1}, X_{q_2}, \ldots, X_{q_T} = \text{hidden state sequence} \]
\[ \pi_i = \text{probability of being in State } i \text{ at } t = 1 \]

IF we knew the state sequence \( X_{q_1} \ldots X_{q_T} \) then

\[ P(O_1, O_2 \ldots O_T | X_{q_1}, \ldots, X_{q_T}) = b_{q_1}(O_1) \ldots b_{q_T}(O_T) \]

But we don't know the state sequence, so add probabilities over all possible state sequences of length \( T \).

\[ P(O_1, \ldots, O_T) = \sum_{q_{T+1}} \sum_{q_T = 1} \sum_{q_1} P(O_1, \ldots, O_T | X_{q_1}, \ldots, X_{q_T}) P(X_{q_1}, \ldots, X_{q_T}) \]

Notation:
\( q_t = i \) means \( q_t = s_i \)
Evaluate

\[ P(o_1 \ldots o_T) = \sum_{q_1=1}^{N} \cdots \sum_{q_T=1}^{N} P(o_1 \ldots o_T | x_{q_1} \ldots x_{q_T}) P(x_{q_1} \ldots x_{q_T}) \]

\[ = \sum_{q_1=1}^{N} \cdots \sum_{q_T=1}^{N} b_q(o_1) \cdots b_q(o_T) \pi_{q_1} a_{q_1 q_2} \cdots a_{q_{T-1} q_T} \]

How long would it take to do this computation?

- \( N^T \) state sequences \( \Rightarrow \) \( N^T-1 \) additions
- \( 2T-1 \) multiplications / state sequence \( \Rightarrow \) \( N^T(2T-1) \) multiplications

\( \sim 2N^T \) operations

If \( N=5, T=100 \), \( 2N^T = 2 \cdot 5^{100} \sim 10^{72} \)

There are \( \sim 3.2 \times 10^7 \) seconds/year

But there is a better way!

\[ \begin{align*}
S_N & \quad \bullet \\
S_{N-1} & \\
& \quad \vdots \\
S_2 & \quad \bullet \\
S_1 & \quad 1 \quad 2 \quad 3 \quad \rightarrow \text{time} \\
\end{align*} \]
Define forward variable $\alpha_t(i)$:

$$\alpha_t(i) = P(O_1, \ldots, O_t, q_t = S_i)$$

= prob. of being in state $i$ at time $t$, having observed $O_1, \ldots, O_t$

Initialization:

$$\alpha_1(i) = P(O_1, q_1 = S_i) = P(q_1 = S_i) P(O_1 | q_1 = S_i) = \prod_i b_i(o_1)$$

Induction:

$$\alpha_{t+1}(i) = P(O_1, \ldots, O_t, q_{t+1} = S_i) = P(O_1, \ldots, O_t, q_{t+1} = S_i) b_i(O_{t+1})$$

$$= \left[ \sum_{k=1}^{N} \alpha_k(i) a_{ki} \right] b_i(O_{t+1})$$

Termination:

$$P(O_1, \ldots, O_T) = \sum_{i=1}^{N} \alpha_T(i)$$

Computation needed $O(N^{2T})$ compare with $N^{T}$
Can also evaluate prob. of observation "backwards"

Define backward variable $\beta_t(i)$

$$\beta_t(i) = P(O_{t+1}, \ldots, O_T | t = S_i)$$

Initialization:

$$\beta_T(i) = 1 \quad 1 \leq i \leq N$$

Induction:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$$
2. Given an HMM and an observation sequence find the single best state sequence that accounts for the observation.

Compute the probability for every possible state sequence and pick the best?

NO!

Let

\[ \delta_t(j) = \max_{q_1, \ldots, q_{t-1}} P(q_1, q_2, \ldots, q_t = j, O_t, \ldots, O_t) \]

over all state sequences of length \(t - 1\)

= highest probability along the single path, which at \(t\) accounts for \(O_t, \ldots, O_t\) and ends at state \(j\).

Note that

\[ \delta_t(j) = \max_i \delta_{t-1}(i) \cdot a_{ij} \cdot b_j(O_t) \]

For each \(t\) and \(j\) only need to record \(\delta_t(j)\) and \(i\) that maximized \(\delta_{t-1}(i) a_{ij}\).
Viterbi Algorithm

Initialization:
\[ S_t(i) = \pi_i b_i(o_t) \quad 1 \leq i \leq N \]
\[ \psi_t(i) = 0 \]

Recursion:
\[ S_t(j) = \max_i S_{t-1}(i) a_{ij} b_j(o_t) \quad 2 \leq t \leq T \]
\[ \psi_t(j) = \arg\max_i [S_{t-1}(i) a_{ij}] \quad 1 \leq i \leq N \]

i.e. \( \psi_t(j) \) is the state (that is i) for which \( S_{t-1}(i) a_{ij} \) is maximized

Termination:
\[ p^* = \max \{S_T(i)\} \]
\[ q_T^* = \arg\max_i [S_T(i)] \]

Back tracking:
\[ q_t^* = \psi_{t+1}(q_{t+1}^*) \quad t = T-1, T-2 \ldots, 1 \]
3. Given an HMM and an observation sequence, find the parameters so that the probability of observation is maximized.

IF we could observe states, then we would only need to count state transitions, etc. But instead we need to use an iterative procedure.

 Baum-Welch or forward-backward algorithm.
Reestimation formulas

Let

\[ \xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | o) \quad o = 0, 0_2, \ldots, 0_T \]

= prob. of being in state \( S_i \) at time \( t \),
and in state \( S_j \) at time \( t+1 \),
given the observation sequence

\[ = P(q_t = S_i, q_{t+1} = S_j, o) / P(o) \]

But

\[ P(q_t = S_i, q_{t+1} = S_j, o) = \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \]

And

\[ P(o) = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \]

\[ = \sum_{i=1}^{N} \alpha_T(i) \]

Let \[ \alpha_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \]

= prob. of being in state \( S_i \)
at time \( t \), given \( o \)
We get

$$\bar{T}_{i:} = \chi_1 (i) = \text{expected frequency in state } S_i \text{ at time } t=1$$

$$\bar{a}_{i: j} = \frac{\sum_{t=1}^{T-1} \chi_{t}(ij)}{\sum_{i} \chi_{t}(i)} = \frac{\text{expected # of transitions from } S_i \text{ to } S_j}{\text{expected # of times in } S_i}$$

$$\bar{b}_{j-k} = \frac{\sum_{t=1}^{T-1} \chi_{t}(j) \delta_{t}(k)}{\sum_{i} \chi_{t}(i)} = \frac{\text{expected # of times in } S_j \text{ and observing symbol } \delta_{k}}{\text{expected # of times in } S_j}$$

* If the observation consists of real numbers and the output probability is a parametric function (e.g. Gaussian) then the parameters of the output probability density have to be estimated iteratively.
Continuous Speech Recognition (CSR)

Continuous speech - string of spoken words with optional pauses between words.

What is CSR?

Let \( w = (w_1, w_2, \ldots, w_n) \) word sequence

\( w \in \text{vocabulary} \)

\( A = \text{acoustic observation}, \ i.e., \)

Speech to be recognized

CSR: Find the word sequence \( \hat{w} \) that maximizes \( p(w|A) \).

i.e.

\[ p(w|A) = \max \ p(w|A) \]

\[ p(w|A) = \frac{p(A|w)p(w)}{p(A)} \leftarrow \text{independent of } w \]

\( \Rightarrow \) Find \( \hat{w} \) such that

\[ p(A|\hat{w}) p(\hat{w}) = \max_w p(A|w) p(w) \]
Find $w$ to maximize $P(A|w)P(w)$

$p(A|w) = \text{prob. of acoustic observation given } w$ $\Rightarrow$ computed from Acoustic Models

$p(w) = \text{prob. of word sequence } w$ $\Rightarrow$ computed from Language Model

How CSR is to be done in principle.

For each $w = (w_1, w_2, \ldots)$ that can be formed from the words in the vocabulary $\{$

Compute $P(A|w)$: For every possible way of aligning $w$ with $A$, compute the probability of alignment & sum all probabilities.

Compute $p(w)$:

$\}$

Pick $w$ for which $P(A|w)p(w)$ is max.

$\Rightarrow$ Need fast search algorithm
A continuous Speech Recognizer

Speech → Signal Processing → Observation Sequence → Recognizer

Acoustic Models: a's, b's, t's
Language Model

Highest-probability word sequence

What do you use for the observation sequence?

1. Sequence of feature vectors computed at regular intervals (e.g. every 10 milliseconds).
   
   \( b_i(0t) \) is often a continuous function of the feature vector (also parametric.)

2. Sequence of labels (discrete symbols) obtained from the sequence of feature vectors by assigning one of \( M \) labels to each feature vector.

Example:

Divide the space of feature vectors into \( M \) cells. Label each cell. If a feature vector belongs to cell \( i \), its label is \( i \).

\( b_i(0t) = b_i("1") \) if \( 0t = "1" \)

etc.
Acoustic Models for computing \( P(A1w) \)

\[ \text{HMM of a word sequence =} \]
\[ \text{concatenation of word HMM's} \]
\[ + (\text{Additional modeling for contextual effects at word boundaries}) \]

\[ \text{popular HMM of a word = concatenation of phoneme HMM's.} \]

(whole word HMM's are not practical for large vocabulary.)
Phoneme HMM's: Issues for model designers

1. How many states per phoneme?
   A: A few (1, 2, 3, 4, 5...)

2. Which model topology?
   A: Some sort of left-to-right topology.

Examples.
3. Which feature vectors?
   A: Some sort of "spectral" information
      ( ~ 10 - 50 components)
      Examples:
      FFT magnitudes
      LPC coefficients
      Cepstral coefficients $\mathcal{F}^{-1} \left[ \log |X(w)| \right]$~ time function

4. What kind of $b_j(0t)$?
   A: Discrete symbol or continuous density?

5. Context dependent phonemes? "Should there be a separate model for 'a' in 'cat' and 'a' in 'sash'?"
   A: Some kind of context dependent phonemes usually used. (Possibly up to several thousand of them!)
      Context includes neighboring phonemes, stress, and word boundary.

6. Speaker dependent or speaker independent models?
Language Model

n-grams are popular

\[ p(\text{wi} | \text{wi-1}, \text{wi-2}, \ldots, \text{wi-(n-1)}) \]

\( n > 3 \) rarely used - need a lot of training text

Currently obtainable performance. (Nov., 1992)

- 20,000 word vocabulary based on Wall Street Journal articles
- Speaker independent recognition
- Training & test speech consist of read sentences from the Wall Street Journal

Word error rate in the low teens.

Near real-time performance possible.