HMMs

CS224N
2004
(based on slides by David Blei, UCB)

HMM formalism

HMM = probabilistic FSA

HMM = states \( i_1, i_2, \ldots \)
(special start state \( i_s \)
(special end state \( i_e \)

Token alphabet \( a_1, a_2, \ldots \)

State transition probs \( P(i_j | i_s) \)

Token emission probs \( P(a_i | i_j) \)

Widely used in many language processing tasks,
\( \epsilon \), speech recognition [Lee, 1989], POS tagging
[Kupiec, 1992], topic detection [Zaman et al, 1998]

What is an HMM?

- Graphical Model Representation: Variables by time
- Circles indicate states
- Arrows indicate probabilistic dependencies between states

What is an HMM?

- Green circles are hidden states
- Dependent only on the previous state: Markov process
- “The past is independent of the future given the present.”

What is an HMM?

- Purple nodes are observed states
- Dependent only on their corresponding hidden state

HMM Formalism

\( \{ S, \Pi, A, B \} \)

\( S : \{ s_1, \ldots, s_m \} \) are the values for the hidden states

\( K : \{ k_1, \ldots, k_n \} \) are the values for the observations
HMM Formalism

- \( \Sigma, K, \Pi, A, B \)
- \( \Pi = \{ \pi_k \} \) are the initial state probabilities
- \( A = \{ a_{ij} \} \) are the state transition probabilities
- \( B = \{ b_i(o) \} \) are the observation state probabilities

Inference for an HMM

- Compute the probability of a given observation sequence
- Given an observation sequence, compute the most likely hidden state sequence
- Given an observation sequence and set of possible models, which model most closely fits the data?

Sequence Probability

Given an observation sequence and a model, compute the probability of the observation sequence

\[ O = (o_1, \ldots, o_T), \mu = (A, B, \Pi) \]

Compute \( P(O | \mu) \)

Sequence probability

\[
P(O | X, \mu) = b_{o_1} b_{o_2} \cdots b_{o_T} = \prod_{i=1}^{T} a_{o_i} \]

\[
P(X | \mu) = \pi_k a_{y_k} a_{y_k} \cdots a_{y_k} \]

\[
P(O, X | \mu) = P(O | X, \mu)P(X | \mu) \]

\[
P(O | \mu) = \sum_x P(O | X, \mu)P(X | \mu) \]

Sequence probability

\[
P(O | \mu) = \sum_{\{x_1, \ldots, x_T\}} \pi_{x_1} b_{x_2} \prod_{t=1}^{T-1} a_{x_t} b_{x_{t+1}} \]

Special structure gives us an efficient solution using dynamic programming.

Intuition: Probability of the first \( t \) observations is the same for all possible \( t + 1 \) length state sequences.

Define: \( \alpha_t(i) = P(o_1 \ldots o_t, x_t = i | \mu) \)
- Viterbi Algorithm (1967)
  The state sequence which maximizes the probability of seeing the observations to observation at time t and seeing the observation at time 1.

- Backward Procedure
  Probability of the rest of the states given the first state

- Forward Procedure
  Probability of the rest of the states given the first state

- Sequence Probability
  Forward Procedure Backward Procedure Combination

The state sequence that best explains the observations.

\[
\beta(t+1) = 1
\]

\[
\beta(t) = \sum_{i} a_{ij} \beta(j, t+1)
\]

\[
\alpha(t) = \max_{i} P(x_t \mid a_0, a_t, \ldots, a_{t-1}, x_1 = i, x_t)
\]

\[
\pi(\mu) = \sum_{i} \alpha(i, t)
\]

\[
\alpha(t) = \sum_{i} \pi(i, t) \beta(i, t+1)
\]

\[
\pi(\mu) = \sum_{i} \alpha(i, 1) \beta(i, T)
\]

\[
\alpha(t) = P(o_t \mid a_t) = P(o_t \mid a_0, a_1, \ldots, a_{t-1}, x_1 = i, x_t)
\]

\[
\pi(\mu) = \sum_{i} \alpha(i, 1) \beta(i, T)
\]

\[
\alpha(t) = P(o_t \mid a_t) = P(o_t \mid a_0, a_1, \ldots, a_{t-1}, x_1 = i, x_t)
\]

\[
\pi(\mu) = \sum_{i} \alpha(i, 1) \beta(i, T)
\]
Is it that easy?

- As often with text, the biggest problem is the sparseness of observations (words)
- Need to use many techniques to do it well
  - **Smoothing** (as in NB) to give suitable nonzero probability to unseens
  - **Featural decomposition** (capitalized?, number?, etc.) gives a better estimate
  - **Shrinkage** allows pooling of estimates over multiple states of same type (e.g., prefix states)
  - Well designed (or learned) HMM topology