Modeling Sequence Data: HMMs and Viterbi

CS4780/5780 – Machine Learning
Fall 2013
Thorsten Joachims
Cornell University

Reading:
Manning/Schuetze, Sections 9.1-9.3 (except 9.3.1)
Leeds Online HMM Tutorial (except Forward and Forward/Backward Algorithm)

Hidden Markov Model

• States: \( y \in \{s_1, \ldots, s_k\} \)
• Outputs symbols: \( x \in \{o_1, \ldots, o_m\} \)
• Starting probability \( P(Y_1 = y_1) \)
  – Specifies where the sequence starts
• Transition probability \( P(Y_i = y_i \mid Y_{i-1} = y_{i-1}) \)
  – Probability that one state succeeds another
• Output/Emission probability \( P(X_i = x_i \mid Y_i = y_i) \)
  – Probability that word is generated in this state

\[ P(x, y) = P(x_1, \ldots, x_L, y_1, \ldots, y_L) \]

\[ = P(y_1)P(x_1|y_1)\prod_{i=2}^{L} P(x_i|y_i)P(y_i|y_{i-1}) \]

Estimating the Probabilities

• Given: Fully observed data
  – Pairs of emission sequence with their state sequence
• Estimating transition probabilities \( P(Y_i = y_i \mid Y_{i-1} = y_{i-1}) \)

\[ \hat{P}(y_i = y_i \mid y_{i-1} = y_{i-1}) = \frac{\# \text{times state } y_i \text{ follows state } y_{i-1}}{\# \text{ of times state } y_{i-1} \text{ occurs}} \]
• Estimating emission probabilities \( P(X_i = x_i \mid Y_i = y_i) \)

\[ \hat{P}(x_i = x_i \mid y_i = y_i) = \frac{\# \text{ of times symbol } x_i \text{ is observed in state } y_i}{\# \text{ of times state } y_i \text{ occurs}} \]

• Smoothing the estimates
  – Laplace smoothing -> uniform prior
  – See naïve Bayes for text classification
• Partially observed data
  – Expectation Maximization (EM)

HMM Decoding: Viterbi Algorithm

• Question: What is the most likely state sequence given an output sequence
  – Given fully specified HMM:
    • \( P(Y_i = y_i \mid Y_{i-1} = y_{i-1}) \)
    • \( P(X_i = x_i \mid Y_i = y_i) \)

\[ \hat{P}(y_i = y_i \mid x_1, \ldots, x_{i-1}, y_1, \ldots, y_{i-1}) = \frac{\pi(y_1)\prod_{k=2}^{i} P(x_k|y_k)P(y_k|y_{k-1})}{\pi(y_1)\prod_{k=2}^{i} P(x_k|y_k)} \]

  – “Viterbi” algorithm has runtime linear in length of sequence
  – Example: find the most likely tag sequence for a given sequence of words

HMM’s for POS Tagging

• Design HMM structure (vanilla)
  – States: one state per POS tag
  – Transitions: fully connected
  – Emissions: all words observed in training corpus
• Estimate probabilities
  – Use corpus, e.g. Treebank
  – Smoothing
  – Unseen words?
• Tagging new sentences
  – Use Viterbi to find most likely tag sequence

Viterbi Example

\[
\begin{array}{c|c|c|c|c|c|c}
\text{P(X|Y)} & \text{I} & \text{bank} & \text{at} & \text{CFU} & \text{go} & \text{to} & \text{the} \\
\hline
\text{DET} & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.94 \\
\text{PRP} & 0.04 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 \\
\text{N} & 0.01 & 0.4 & 0.01 & 0.4 & 0.16 & 0.01 & 0.01 \\
\text{PREP} & 0.01 & 0.01 & 0.48 & 0.01 & 0.01 & 0.47 & 0.01 \\
\text{V} & 0.01 & 0.4 & 0.01 & 0.01 & 0.55 & 0.01 & 0.01 \\
\hline
\text{P(Y|I)} & \text{P(Y|bank)} & \text{P(Y|at)} & \text{P(Y|CFU)} & \text{P(Y|go)} & \text{P(Y|to)} & \text{P(Y|the)} \\
\hline
\text{DET} & 0.3 & \text{DET} & 0.01 & 0.01 & 0.96 & 0.01 & 0.01 \\
\text{PRP} & 0.3 & \text{PRP} & 0.01 & 0.01 & 0.01 & 0.2 & 0.77 \\
\text{N} & 0.1 & \text{N} & 0.01 & 0.2 & 0.3 & 0.3 & 0.19 \\
\text{PREP} & 0.1 & \text{PREP} & 0.3 & 0.2 & 0.3 & 0.19 & 0.01 \\
\text{V} & 0.2 & \text{V} & 0.2 & 0.19 & 0.3 & 0.3 & 0.01 \\
\end{array}
\]
Experimental Results

<table>
<thead>
<tr>
<th>Tagger</th>
<th>Accuracy</th>
<th>Training time</th>
<th>Prediction time</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM</td>
<td>96.80%</td>
<td>20 sec</td>
<td>18,000 words/s</td>
</tr>
<tr>
<td>TBL Rules</td>
<td>96.47%</td>
<td>9 days</td>
<td>750 words/s</td>
</tr>
</tbody>
</table>

- **Experiment setup**
  - WSJ Corpus
  - Trigram HMM model
  - Lexicalized
  - from [Pla and Molina, 2001]

### Discriminative vs. Generative

- **Bayes Rule**
  \[
  h_{\text{bayes}}(x) = \arg\max_{y \in Y} P(Y = y|X = x) = \arg\max_{y \in Y} P(X = x|Y = y)P(Y = y)
  \]

- **Generative**
  - Make assumptions about \(P(X = x|Y = y)\) and \(P(Y = y)\)
  - Estimate parameters of the two distributions

- **Discriminative**
  - Define set of prediction rules (i.e. hypotheses) \(H\)
  - Find \(h\) in \(H\) that best approximates the classifications made by
  \[
  h_{\text{bayes}}(x) = \arg\max_{y \in Y} P(Y = y|X = x)
  \]

- **Question**: Can we train HMM’s discriminately?
  - Later in semester: discriminative training of HMM and general structured prediction.