# CS778 Fall 2006

## Primer on Hidden Markov Models

Thorsten Joachims

Cornell University
Department of Computer Science

## Part-of-Speech Tagging

- **Predict sequence of POS tags for sequence of words:**

| sentence | POS |
|---|---|
| $x_1 = (The, bear, chased, the, cat)$ | $y_1 = (DET, N, V, DET, N)$ |
| $x_2 = (Students, bear, a, burden)$ | $y_2 = (N, V, DET, N)$ |

- **Ambiguity**
  - He will race/V the car.
  - When will the race/NOUN end?
  - I bank/V at CFCU.
  - Go to the bank/NOUN!
- **Average of ~2 parts of speech for each word**
- **20 – 400 different tags (i.e. word classes)**

## Predicting Sequences: Hidden Markov Model

- **Bayes rule:** $h(x) = \operatorname{argmax}_{y \in Y} [P(X=x|Y=y)P(Y=y)]$
- **Independence assumptions for compact representation**

$$P(Y=(y^{(1)},..,y^{(l)})) = \prod_{i=1}^{l} P(Y_c=y^{(i)}|Y_p=y^{(i-1)})$$

$$P(X=(x^{(1)},..,x^{(l)})|Y=(y^{(1)},..,y^{(l)})) = \prod_{i=1}^{l} P(X_c=x^{(i)}|Y_c=y^{(i)})$$

| y | Det | → N | → V | → Det | → N |
|---|---|---|---|---|---|
| x | The | bear | chased | the | cat |

- **Prediction rule:**

$$h(x) = \operatorname{argmax}_{y \in Y} [P(X=x|Y=y)P(Y=y)]$$

$$= \operatorname{argmax}_{(y^{(1)},..,y^{(l)}) \in Y} \left[ \prod_{i=1}^{l} P(Y_c=y^{(i)}|Y_p=y^{(i-1)}) P(X_c=x^{(i)}|Y_c=y^{(i)}) \right]$$

## Hidden Markov Model (HMM)

- **States: $y \in \{s_1,\ldots,s_k\}$**
  - Special starting state $s_0$
- **Outputs symbols: $x \in \{o_1,\ldots,o_m\}$**
- **Transition probability $P(Y_c = y^{(i)} \mid Y_p = y^{(i-1)})$**
  - Probability that one states succeeds another
- **Output/Emission probability $P(X_c = x^{(i)} \mid Y_c = y^{(i)})$**
  - Probability that word is generated in this state

**=> Every output + state sequence has a probability**

$$P(X=x, Y=y) = P(X=x|Y=y)P(Y=y)$$

$$= \left[ \prod_{i=1}^{l} P(X_c=x^{(i)}|Y_c=y^{(i)}) \right] \left[ \prod_{i=1}^{l} P(Y_c=y^{(i)}|Y_p=y^{(i-1)}) \right]$$

$$= \left[ \prod_{i=1}^{l} P(X_c=x^{(i)}|Y_c=y^{(i)}) P(Y_c=y^{(i)}|Y_p=y^{(i-1)}) \right]$$

## Estimating HMM Probabilities

- **Maximum Likelihood: Given $(x_1, y_1), \ldots, (x_n, y_n)$, find**

$$\omega^* = \operatorname{argmax}_{\omega \in \Omega} \prod_{i=1}^{n} [P(Y=y_i, X=x_i|\omega)]$$

$$= \operatorname{argmax}_{\omega \in \Omega} \left[ \prod_{i=1,j=1}^{n} \prod^{l} P(Y_c=y_i^{(j)}|Y_p=y_i^{(j-1)}) P(X_c=x_i^{(j)}|Y_c=y_i^{(j)}) \right]$$

- **Closed-form solutions**
  - Estimating transition probabilities $P(Y_c = y_a \mid Y_p = y_b)$

$$P(Y_c = y_a \mid Y_p = y_b) = \frac{\#ofTimesStateAFollowsStateB}{\#ofTimesStateBOccurs}$$

  - Estimating mission probabilities $P(X_c = x_a \mid Y_c = y_b)$

$$P(X_c = x_a \mid Y_c = y_b) = \frac{\#ofTimesOutputAIsObservedInStateB}{\#ofTimesStateBOccurs}$$

- **Need for smoothing the estimates (e.g. Laplace)**

## HMM Prediction: Viterbi Algorithm

**Prediction: Find most likely state sequence**

- Given x and fully specified HMM:
  - $P(Y_c = y_a \mid Y_p = y_b)$
  - $P(X_c = x_a \mid Y_c = y_b)$
- Find the most likely state (i.e tag) sequence $(y_1,\ldots,y_l)$ for a given sequence of observed output symbols (i.e. words) $(x_1,\ldots,x_l)$

$$h(x) = \operatorname{argmax}_{(y^{(1)},..,y^{(l)}) \in Y} \left[ \prod_{i=1}^{l} P(Y_c=y^{(i)}|Y_p=y^{(i-1)}) P(X_c=x^{(i)}|Y_c=y^{(i)}) \right]$$

- Viterbi algorithm uses dynamic programming
  - Construct trellis graph for HMM
  - Shortest path in this graph is most likely state sequence
- Viterbi algorithm has runtime linear in length of sequence

## Experimental Results

| Tagger | Accuracy | Training time | Prediction time |
|---|---|---|---|
| **HMM** | **96.24%** | **20 sec** | **???** |
| **Tri-HMM** | **96.45%** | | **???** |
| **LexTri-HMM** | **96.80%** | | **18.000 words/s** |
| **TBL Rules** | **96.47%** | **9 days** | **750 words/s** |

- **Experiment**
  - WSJ Corpus
  - from [Pla and Molina, 2001]

## Reading

- **Leeds Online HMM Tutorial http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev/main.html**
- **C. Manning and H. Schuetze, Foundations of Statistical NLP, MIT Press, 1999.**