

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
$x_1 = (\text{The, bear, chased, the, cat})$	$y_1 = (\text{DET, N, V, DET, N})$
$x_2 = (\text{Students, bear, a, burden})$	$y_2 = (\text{N, V, DET, N})$

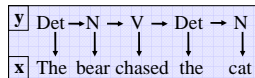
- Ambiguity
 - He will **race**/V the car.
 - When will the **race**/NOUN end?
 - I **bank**/V at CFCU.
 - Go to the **bank**/NOUN!
- Average of ~2 parts of speech for each word
- 20 – 400 different tags (i.e. word classes)

Predicting Sequences: Hidden Markov Model

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

$$P(Y = (y^{(1)}, \dots, y^{(l)})) = \prod_{i=1}^l P(y_i = y^{(i)} | y_{i-1} = y^{(i-1)})$$

$$P(X = (x^{(1)}, \dots, x^{(l)}) | Y = (y^{(1)}, \dots, y^{(l)})) = \prod_{i=1}^l P(x_i = x^{(i)} | y_i = y^{(i)})$$



- Prediction rule:

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

$$= \underset{(y^{(1)}, \dots, y^{(l)}) \in Y}{\operatorname{argmax}} \left[\prod_{i=1}^l P(y_i = y^{(i)} | y_{i-1} = y^{(i-1)}) P(x_i = x^{(i)} | y_i = y^{(i)}) \right]$$

Hidden Markov Model (HMM)

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

=> Every output + state sequence has a probability

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

$$= \left[\prod_{i=1}^l P(x_i = x^{(i)} | y_i = y^{(i)}) \right] \left[\prod_{i=1}^l P(y_i = y^{(i)} | y_{i-1} = y^{(i-1)}) \right]$$

$$= \left[\prod_{i=1}^l P(x_i = x^{(i)} | y_i = y^{(i)}) P(y_i = y^{(i)} | y_{i-1} = y^{(i-1)}) \right]$$

Estimating HMM Probabilities

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

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

$$= \underset{w \in \Omega}{\operatorname{argmax}} \left[\prod_{i=1}^n \prod_{j=1}^l P(y_j = y_i^{(j)} | y_{j-1} = y_i^{(j-1)}) P(x_i = x_i^{(i)} | y_i = y_i^{(i)}) \right]$$

- Closed-form solutions

- Estimating transition probabilities $P(Y_c = y_a | Y_p = y_b)$

$$P(Y_c = y_a | Y_p = y_b) = \frac{\# \text{ of Times State A Follows State B}}{\# \text{ of Times State B Occurs}}$$

- Estimating emission probabilities $P(X_c = x_a | Y_c = y_b)$

$$P(X_c = x_a | Y_c = y_b) = \frac{\# \text{ of Times Output A Observed In State B}}{\# \text{ of Times State B Occurs}}$$

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

HMM Prediction: Viterbi Algorithm

Prediction: Find most likely state sequence

- Given x and fully specified HMM:

- $P(Y_c = y_a | Y_p = y_b)$
- $P(X_c = x_a | Y_c = y_b)$

- Find the most likely state (i.e. tag) sequence (y_1, \dots, y_l) for a given sequence of observed output symbols (i.e. words) (x_1, \dots, x_l)

$$h(x) = \underset{(y^{(1)}, \dots, y^{(l)}) \in Y}{\operatorname{argmax}} \left[\prod_{i=1}^l P(y_i = y^{(i)} | y_{i-1} = y^{(i-1)}) P(x_i = x^{(i)} | y_i = y^{(i)}) \right]$$

- 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.**