CS6784 - Spring 2010

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
$\mathbf{x}_1 = (The, bear, chased, the, cat)$	$\mathbf{y}_1 = (DET, N, V, DET, N)$
$\mathbf{x}_2 = (Students, bear, a, burden)$	$\mathbf{y}_2 = (N, V, DET, N)$

- **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

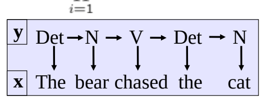
- **Bayes rule:** $h(x) = \operatorname{argmax}_{y \in Y} [P(X=x|Y=y)P(Y=y)]$
  - Generative model
- **Design decisions:**
  - Representation
    - Linear chain Hidden Markov Model
  - Prediction (i.e. inference)
    - Viterbi algorithm
  - Learning
    - Maximum likelihood

Representation: 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**

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

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



- **Prediction rule:**

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

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

Representation: Hidden Markov Model

- **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

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

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

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

Learning: Estimating HMM Probabilities

- **Maximum Likelihood: Given $(\mathbf{x}_1,\mathbf{y}_1),\dots,(\mathbf{x}_n,\mathbf{y}_n)$, find**

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

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

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

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

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

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

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

Prediction/Inference: Viterbi Algorithm

Prediction: Find most likely state sequence

- Given x and fully specified HMM:
  - $P(Y_c = y_a \mid Y_p = y_b)$ (transition probabilities)
  - $P(X_c = x_a \mid Y_c = y_b)$ (emission probabilities)
- 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) $(x_1,\ldots,x_l)$

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

- 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

Viterbi Example

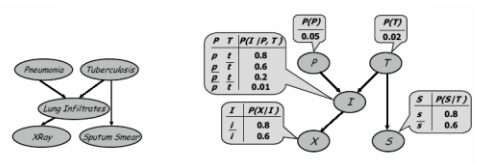
P(X=x|Y=y)	I	bank	at	CFCU	go	to	the
DET	0.01	0.01	0.01	0.01	0.01	0.01	0.94
PRP	0.94	0.01	0.01	0.01	0.01	0.01	0.01
N	0.01	0.4	0.01	0.4	0.16	0.01	0.01
PREP	0.01	0.01	0.48	0.01	0.01	0.47	0.01
V	0.01	0.4	0.01	0.01	0.55	0.01	0.01

$P(Y\|Y_{prev})$	DET	PRP	N	PREP	V
START	0.3	0.3	0.1	0.1	0.2
DET	0.01	0.01	0.96	0.01	0.01
PRP	0.01	0.01	0.01	0.2	0.77
N	0.01	0.2	0.3	0.3	0.19
PREP	0.3	0.2	0.3	0.19	0.01
V	0.2	0.19	0.3	0.3	0.01

Directed Graphical Models

- **Representation of joint distribution**
  - Exploit conditional independence between random variables
- **Example**
  - Joint distribution

$$P(P,T,I,X,S) = P(P)P(T)P(I \mid P,T)P(X \mid I)P(S \mid T)$$

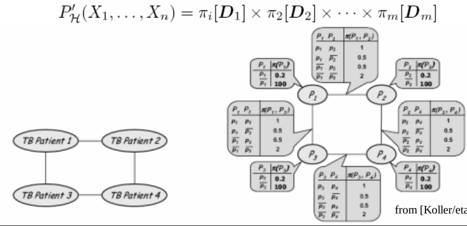


from [Koller/etal/07]

Undirected Graphical Models

- **Markov Networks / Markov Random Fields**
  - More flexible representation of joint distribution
- **Example**
  - Joint distribution $P_{\mathcal{H}}(X_1,\ldots,X_n) = \frac{1}{Z}P'(X_1,\ldots,X_n)$

$$P'_{\mathcal{H}}(X_1,\ldots,X_n) = \pi_i[D_1] \times \pi_2[D_2] \times \cdots \times \pi_m[D_m]$$



from [Koller/etal/07]