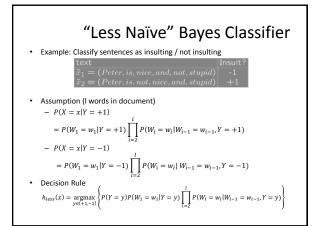
Modeling Sequence Data: Markov Models

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) (http://www.comp_leeds.ac.uk/roger/HiddenMarkovModels/html dev/main.html)



Markov Model

- Definition
 - Set of States: s₁,...,s_k
 - Start probabilities: P(S₁=s)
 - Transition probabilities: P(S_i=s | S_{i-1}=s')
- Random walk on graph
 - Start in state s with probability $P(S_1=s)$
 - Move to next state with probability $P(S_i=s \mid S_{i-1}=s')$
- Assumptions
 - Limited dependence: Next state depends only on previous state, but no other state (i.e. first order Markov model)
 - Stationary: $P(S_i=s \mid S_{i-1}=s')$ is the same for all i

Part-of-Speech Tagging Task

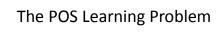
 Assign the correct part of speech (word class) to each word in a document

"The/DT planet/NN Jupiter/NNP and/CC its/PRP moons/NNS are/VBP in/IN effect/NN a/DT mini-solar/JJ system/NN ,/, and/CC Jupiter/NNP itself/PRP is/VBZ often/RB called/VBN a/DT star/NN that/IN never/RB caught/VBN fire/NN ./"

- Needed as an initial processing step for a number of language technology applications
 - Information extraction
 - Answer extraction in QA
 - Base step in identifying syntactic phrases for IR systems
 - Critical for word-sense disambiguation (WordNet apps)
 - ...

Why is POS Tagging Hard?

- Ambiguity
 - He will race/VB the car.
 - When will the race/NN end?
 - I bank/VB at CFCU.
 - Go to the bank/NN!
- Average of ~2 parts of speech for each word
 - The number of tags used by different systems varies a lot. Some systems use < 20 tags, while others use > 400.



Example

sentence $\bar{x}_1 = (I, bank, at, CFCU)$ $\bar{y}_1 = (PRP, V, PREP, N)$ $\bar{y}_2 = (Go, to, the, bank)$ $\bar{y}_2 = (V, PREP, DFT, N)$

Hidden Markov Model for POS Tagging

- States
 - Think about as nodes of a graph
 - One for each POS tag
 - special start state (and maybe end state)
- Transitions
 - Think about as directed edges in a graph
 - Edges have transition probabilities
- Output
 - Each state also produces a word of the sequence
 - Sentence is generated by a walk through the graph

Hidden Markov Model

- States: $y \in \{s_1, ..., s_k\}$
- Outputs symbols: $x \in \{o_1, ..., o_m\}$
- Starting probability P(Y₁ = y₁)
 Specifies where the sequence starts
- Transition probability P(Y_i = y_i | Y_{i-1} = y_{i-1})
 Probability that one states succeeds another
- Output/Emission probability P(X_i = x_i | Y_i = y_i)
 Probability that word is generated in this state
- => Every output+state sequence has a probability

$$P(x, y) = P(x_1, \dots, x_l, y_1, \dots, y_l)$$

= $P(y_1)P(x_1|y_1) \prod_{i=2}^{l} P(x_i|y_i)P(y_i|y_{i-1})$

Estimating the Probabilities

- Given: Fully observed data — Pairs of output sequence with their state sequence
- Estimating transition probabilities $P(Y_i | Y_{i-1})$ $P(Y_i = a | Y_{i-1} = b) = \frac{\# \text{ of times state a follows state b}}{\# \text{ of times state b occurs}}$
- Estimating emission probabilities $P(X_i | Y_i)$ $P(X_i = a|X_i = b) = \#$ of times output a is observed in state b
- Smoothing the estimates
 - Laplace smoothing -> uniform prior
 - See naïve Bayes for text classification
 - Partially observed data
 - Expectation Maximization (EM)