CSCI 5832 Natural Language Processing

Jim Martin

Lecture 9

2/28/08 1

Today 2/19

- Review HMMs for POS tagging
- Entropy intuition
- Statistical Sequence classifiers
  - HMMs
  - MaxEnt
  - MEMMs

2/28/08 2

Statistical Sequence Classification

- Given an input sequence, assign a label (or tag) to each element of the tape
  - Or... Given an input tape, write a tag out to an output tape for each cell on the input tape
- Can be viewed as a classification task if we view
  - The individual cells on the input tape as things to be classified
  - The tags written on the output tape as the class labels

2/28/08 3

POS Tagging as Sequence Classification

- We are given a sentence (an “observation” or “sequence of observations”)
  - *Secretariat is expected to race tomorrow*
- What is the best sequence of tags which corresponds to this sequence of observations?
- Probabilistic view:
  - Consider all possible sequences of tags
  - Out of this universe of sequences, choose the tag sequence which is most probable given the observation sequence of n words w1…wn.

Statistical Sequence Classification

- We want, out of all sequences of n tags $t_1 \ldots t_n$ the single tag sequence such that $P(t_1 \ldots t_n | w_1 \ldots w_n)$ is highest.

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}}\, P(t_1^n | w_1^n)$$

- Hat ^ means “our estimate of the best one”
- Argmax_x f(x) means “the x such that f(x) is maximized”

Road to HMMs

- This equation is guaranteed to give us the best tag sequence

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}}\, P(t_1^n | w_1^n)$$

- But how to make it operational? How to compute this value?
- Intuition of Bayesian classification:
  - Use Bayes rule to transform into a set of other probabilities that are easier to compute

Using Bayes Rule

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} \frac{P(w_1^n|t_1^n)P(t_1^n)}{P(w_1^n)}$$

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(w_1^n|t_1^n)P(t_1^n)$$

2/28/08 7

Likelihood and Prior

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} \overbrace{P(w_1^n|t_1^n)}^{\text{likelihood}} \overbrace{P(t_1^n)}^{\text{prior}}$$

$$P(w_1^n|t_1^n) \approx \prod_{i=1}^{n} P(w_i|t_i)$$

$$P(t_1^n) \approx \prod_{i=1}^{n} P(t_i|t_{i-1})$$

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n|w_1^n) \approx \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1})$$

2/28/08 8

Transition Probabilities

- Tag transition probabilities $p(t_i|t_{i-1})$
  - Determiners likely to precede adjs and nouns
    - That/DT flight/NN
    - The/DT yellow/JJ hat/NN
    - So we expect P(NN|DT) and P(JJ|DT) to be high
  - Compute P(NN|DT) by counting in a labeled corpus:

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$$

$$P(NN|DT) = \frac{C(DT,NN)}{C(DT)} = \frac{56,509}{116,454} = .49$$

2/28/08 9

Observation Probabilities

- Word likelihood probabilities $p(w_i|t_i)$
  - VBZ (3sg Pres verb) likely to be "is"
  - Compute P(is|VBZ) by counting in a labeled corpus: $$P(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)}$$

$$P(is|VBZ) = \frac{C(VBZ, is)}{C(VBZ)} = \frac{10,073}{21,627} = .47$$

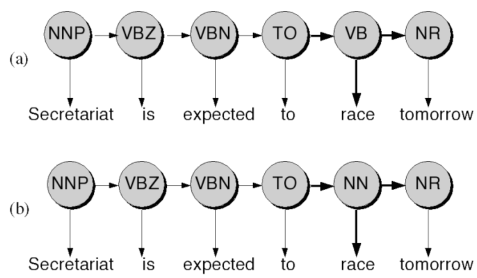
2/28/08 10

An Example: the verb "race"

- Secretariat/NNP is/VBZ expected/VBN to/TO **race**/VB tomorrow/NR
- People/NNS continue/VB to/TO inquire/VB the/DT reason/NN for/IN the/DT **race**/NN for/IN outer/JJ space/NN
- How do we pick the right tag?

2/28/08 11

Disambiguating "race"



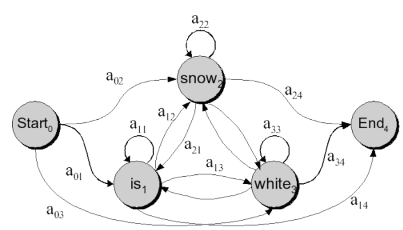
2/28/08 12

Example

- P(NN|TO) = .00047
- P(VB|TO) = .83
- P(race|NN) = .00057
- P(race|VB) = .00012
- P(NR|VB) = .0027
- P(NR|NN) = .0012
- P(VB|TO)P(NR|VB)P(race|VB) = .00000027
- P(NN|TO)P(NR|NN)P(race|NN)=.00000000032
- So we (correctly) choose the verb reading,

2/28/08 13

Markov chain for words



2/28/08 14

Markov chain = "First-order Observable Markov Model"

- A set of states
  - Q = $q_1, q_2 \ldots q_N$; the state at time t is q_t
- Transition probabilities:
  - a set of probabilities A = $a_{01}a_{02}\ldots a_{n1}\ldots a_{nn}$.
  - Each a_{ij} represents the probability of transitioning from state i to state j
  - The set of these is the transition probability matrix A
- Current state only depends on previous state

2/28/08 15

Hidden Markov Models

- States $Q = q_1, q_2 \ldots q_{N;}$
- Observations $O= o_1, o_2 \ldots o_{N;}$
  - Each observation is a symbol from a vocabulary $V = \{v_1,v_2,\ldots v_V\}$
- Transition probabilities
  - Transition probability matrix $A = \{a_{ij}\}$

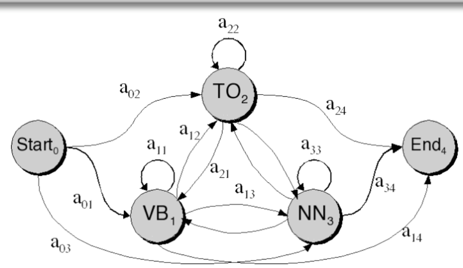
    [illegible]
- Observation likelihoods
  - Output probability matrix $B=\{b_i(k)\}$

    [illegible]
- Special initial probability vector π [illegible]

2/28/08

16

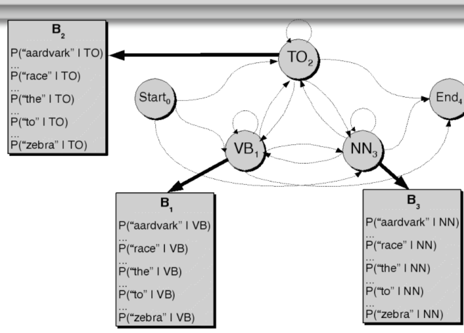
Transitions between the hidden states of HMM, showing A probs



2/28/08

17

B observation likelihoods for POS HMM



2/28/08

18

The A matrix for the POS HMM

	VB	TO	NN	PPSS
<s>	.019	.0043	.041	.067
VB	.0038	.035	.047	.0070
TO	.83	0	.00047	0
NN	.0040	.016	.087	.0045
PPSS	.23	.00079	.0012	.00014

Figure 4.15 Tag transition probabilities (the a array, $p(t_i|t_{i-1})$ computed from the 87-tag Brown corpus without smoothing. The rows are labeled with the conditioning event; thus $P(PPSS|VB)$ is .0070. The symbol <s> is the start-of-sentence symbol.

2/28/08 19

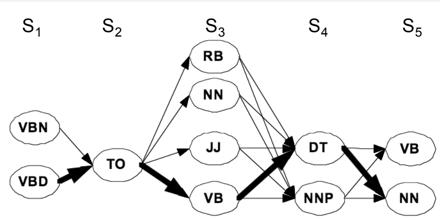
The B matrix for the POS HMM

	I	want	to	race
VB	0	.0093	0	.00012
TO	0	0	.99	0
NN	0	.000054	0	.00057
PPSS	.37	0	0	0

Figure 4.16 Observation likelihoods (the b array) computed from the 87-tag Brown corpus without smoothing.

2/28/08 20

Viterbi intuition: we are looking for the best 'path'



[redacted] [redacted] [redacted] [redacted] [redacted]

2/28/08 21

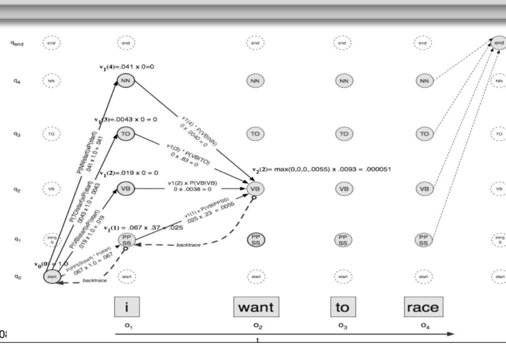
The Viterbi Algorithm

function VITERBI(*observations* of len T, *state-graph* of len N) **returns** *best-path*

create a path probability matrix *viterbi[N+2,T]*
for each state s **from** 1 **to** N **do** ;initialization step
$viterbi[s,1] \leftarrow a_{0,s} * b_s(o_1)$
$backpointer[s,1] \leftarrow 0$
for each time step t **from** 2 **to** T **do** ;recursion step
for each state s **from** 1 **to** N **do**
$viterbi[s,t] \leftarrow \max_{s'=1}^{N} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$
$backpointer[s,t] \leftarrow \operatorname{argmax}_{s'=1}^{N} viterbi[s',t-1] * a_{s',s}$
$viterbi[q_F,T] \leftarrow \max_{s=1}^{N} viterbi[s,T] * a_{s,q_F}$; termination step
$backpointer[q_F,T] \leftarrow \operatorname{argmax}_{s=1}^{N} viterbi[s,T] * a_{s,q_F}$; termination step
return the backtrace path by following backpointers to states back in time from $backpointer[q_F,T]$

2/28/08 22

Viterbi example



2/28/08 23

Information Theory

- Who is going to win the World Series next year?
- Well there are 30 teams. Each has a chance, so there’s a 1/30 chance for any team…? No.
  - Rockies? Big surprise, lots of information
  - Yankees? No surprise, not much information

2/28/08 24

Information Theory

- How much uncertainty is there when you don't know the outcome of some event (answer to some question)?
- How much information is to be gained by knowing the outcome of some event (answer to some question)?

2/28/08 25

Aside on logs

- Base doesn't matter. Unless I say otherwise, I mean base 2.
- Probabilities lie between 0 an 1. So log probabilities are negative and range from 0 (log 1) to –infinity (log 0).
- The – is a pain so at some point we'll make it go away by multiplying by -1.

2/28/08 26

Entropy

- Let's start with a simple case, the probability of word sequences with a unigram model
- Example
  - S = "One fish two fish red fish blue fish"
  - P(S) = P(One)P(fish)P(two)P(fish)P(red)P(fish)P(blue)P(fish)
  - Log P(S) = Log P(One)+Log P(fish)+…Log P(fish)

2/28/08 27

Entropy cont.

- In general that's [redacted]
- But note that
  - the order doesn't matter
  - that words can occur multiple times
  - and that they always contribute the same each time
  - so rearranging… [redacted]

2/28/08 28

Entropy cont.

- One fish two fish red fish blue fish
- Fish fish fish fish one two red blue

[redacted]

2/28/08 29

Entropy cont.

- Now let's divide both sides by N, the length of the sequence:

[redacted]

- That's basically an average of the logprobs

2/28/08 30

Entropy

- Now assume the sequence is really really long.
- Moving the N into the summation you get

[illegible]

- Rewriting and getting rid of the minus sign

[illegible]

2/28/08 31

Entropy

- Think about this in terms of uncertainty or surprise.
  - The more likely a sequence is, the lower the entropy. Why?

[illegible]

2/28/08 32

Model Evaluation

- Remember the name of the game is to come up with statistical models that capture something useful in some body of text or speech.
- There are precisely a gazzilion ways to do this
  - N-grams of various sizes
  - Smoothing
  - Backoff…

2/28/08 33

Model Evaluation

- Given a collection of text and a couple of models, how can we tell which model is best?
- Intuition… the model that assigns the highest probability to a set of withheld text
  - Withheld text? Text drawn from the same distribution (corpus), but not used in the creation of the model being evaluated.

2/28/08 34

Model Evaluation

- The more you're surprised at some event that actually happens, the worse your model was.
- We want models that minimize your surprise at observed outcomes.
- Given two models and some training data and some withheld test data… which is better?

2/28/08 35

Three HMM Problems

- Given a model and an observation sequence
  - Compute Argmax P(states | observation seq)
    - Viterbi
  - Compute P(observation seq | model)
    - Forward
  - Compute P(model | observation seq)
    - EM (magic)

2/28/08 36

Viterbi

- Given a model and an observation sequence, what is the most likely state sequence
  - The state sequence is the set of labels assigned
  - So using Viterbi with an HMM solves the sequence classification task

2/28/08 37

Forward

- Given an HMM model and an observed sequence, what is the probability of that sequence?
  - P(sequence | Model)
  - Sum of all the paths in the model that could have produced that sequence
  - So...
    - How do we change Viterbi to get Forward?

2/28/08 38

Who cares?

- Suppose I have two different HMM models extracted from some training data.
- And suppose I have a good-sized set of held-out data (not used to produce the above models).
- How can I tell which model is the better model?

2/28/08 39

Learning Models

- Now assume that you just have a single HMM model (pi, A, and B tables)
- How can I produce a second model from that model?
  - Rejigger the numbers... (in such a way that the tables still function correctly)
  - Now how can I tell if I've made things better?

2/28/08 40

EM

- Given an HMM structure and a sequence, we can learn the best parameters for the model without explicit training data.
  - In the case of POS tagging all you need is unlabelled text.
  - Huh? Magic. We'll come back to this.

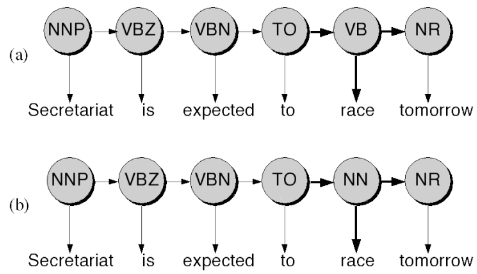
2/28/08 41

Generative vs. Discriminative Models

- For POS tagging we start with the question... P(tags | words) but we end up via Bayes at
  - P(words| tags)P(tags)
  - That's called a generative model
  - We're reasoning backwards from the models that could have produced such an output

2/28/08 42

Disambiguating "race"



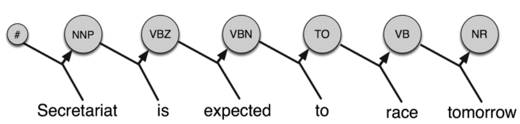
2/28/08 43

Discriminative Models

- What if we went back to the start to
  - Argmax P(tags|words) and didn't use Bayes?
  - Can we get a handle on this directly?
  - First let's generalize to P(tags|evidence)
    - Let's make some independence assumptions and consider the previous state and the current word as the evidence. How does that look as a graphical model?

2/28/08 44

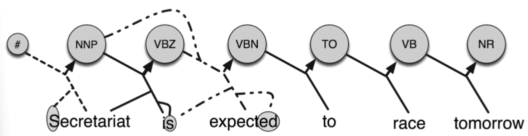
MaxEnt Tagging



2/28/08 45

MaxEnt Tagging

- This framework allows us to throw in a wide range of “features”. That is, evidence that can help with the tagging.



Statistical Sequence Classification