

CSE 842 Natural Language Processing

Lecture 6: Hidden Markov Model (1)

2/2/2009

CSE842, Spring 2009, MSU

1

Back to POS Tagging

Suppose we have

- A set of tags (tagset)
- A training data set where each sentence is annotated with a sequence of correct tags

*The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN
of/IN other/JJ topics/NNS ./.*

The goal: automatically recognize the best sequence of POS for any new unseen sentence.

2/2/2009

CSE842, Spring 2009, MSU

2

Viterbi Algorithm

Initialization Step

For $i = 1$ to M do
 $VSCORE(i, 1) = P(w_i | t_i) * P(t_i | \emptyset);$
 $BACKPTR(i, 1) = 0$

Iteration Step

For $j = 2$ to N
for $i = 1$ to M
 $VSCORE(i, j) = \max_{k=1}^M (VSCORE(k, j-1) * P(t_i | t_k)) P(w_i | t_i)$
 $BACKPTR(i, j) = \text{index of } k \text{ that gave the max above}$

Sequence Identification Step

$C(N) = i$ that maximizes $VSCORE(i, N)$
For $j = N - 1$ to 1 do
 $C(j) = BACKPTR(C(j+1), j+1)$

2/2/2009

CSE842, Spring 2009, MSU

3

Viterbi Algorithm for POS

- The basic operation is to sweep methodically through a two dimensional array filling in the columns one at a time
- For each entry, the highest probability path through the array to that entry is computed and stored. (a pointer to remember where it came from is also stored)

2/2/2009

CSE842, Spring 2009, MSU

4

Dynamic Programming

- Dynamic programming approaches operate by solving small problems once and remember the answers in a table so that they can be used to solve the overall problem.
 - Need best solutions to sub-problems
 - Need optimization criteria to indicate a given solution is the best solution to a sub-problem.
 - Need to only remember the best solution to that problem, not all the solutions.

2/2/2009

CSE842, Spring 2009, MSU

5

Dynamic Programming

- Dynamic programming approaches operate by solving small problems once and remember the answers in a table so that they can be used to solve the overall problem.
 - Need best solutions to sub-problems
 - Need optimization criteria to indicate a given solution is the best solution to a sub-problem.
 - Need to only remember the best solution to that problem, not all the solutions.

2/2/2009

CSE842, Spring 2009, MSU

6

Dynamic Programming

- Minimum Editing distance
 - Viterbi
- All are bottom up table filters. Each element of the table is computed from previously entered elements. Each entered element represents the best solution to the sub problem represented by that table element

2/2/2009

CSE842, Spring 2009, MSU

7

Transformation-based Tagging

- The rule based approach is too expensive, slow, and tedious, etc.
- Brill's Transformation based learning (TBL)
- The idea is to do a poor job first and then use the learned rules to improve things

2/2/2009

CSE842, Spring 2009, MSU

8

Brill's Tagger

Example:

Step 1: Tag all uses of "race" as nouns (based on unigrams, use the most frequently used tag)

Secretariat/NNP is/VBZ expected/VBN to/TO **race/NN** Tomorrow/NN

Step 2: Based on training corpus, refine the rule

Rule: change NN to VB if the previous tag is TO

2/2/2009

CSE842, Spring 2009, MSU

9

Brill's Tagger

Example:

Rule: change NN to VB if the previous tag is TO

This works fine for

is expected to race tomorrow

Leave the following alone

the race for outer space

And screws this one up

drawing the district line according to race

2/2/2009

CSE842, Spring 2009, MSU

10

Brill's Tagger

Assume some tagged training corpus

1. Tag the corpus with the most likely tag for each word (unigram model)
2. Choose a transformation that deterministically replaces an existing tag with a new tag such that the resulting tagged training corpus has the lowest error rate out of all transformations
3. Apply that transformation to the training set
4. Iterate 2 and 3
5. Stop based on some criterion, and return an ordered list of transformations

2/2/2009

CSE842, Spring 2009, MSU

11

Brill's Tagger

Brill's template: change tag a to tag b when:

The preceding (following) word is tagged z

The word two before (after) is tagged z

One of the two preceding (following) words is tagged z

One of the three preceding (following) words is tagged z

The preceding word is tagged z and the following word is tagged w .

The preceding word (following) word is tagged z and the word two before (after) is tagged w .

2/2/2009

CSE842, Spring 2009, MSU

12

Evaluation

- Compare them with a human labeled Gold Standard test set, based on percent correct.
- Human ceiling: when using a human Gold Standard to evaluate a classification algorithm, check the agreement rate of humans on the standard
 - In POS, 96-97%
- Baseline: check with the performance that any dumb approach could achieve.
 - In POS, unigram

2/2/2009

CSE842, Spring 2009, MSU

13

Error Analysis

- Look at a confusion matrix

	IN	JJ	NN	NNP	RB	VBD	VBN
IN	—	.2			.7		
JJ	.2	—	3.3	2.1	1.7	.2	2.7
NN		8.7	—				.2
NNP	.2	3.3	4.1	—	.2		
RB	2.2	2.0	.5		—		
VBD	.3	.5				—	4.4
VBN		2.8				2.6	—

- See what errors are causing problems
 - Noun (NN) vs ProperNoun (NNP) vs Adj (JJ)
 - Preterite (VBD) vs Participle (VBN) vs Adjective (JJ)

2/2/2009

CSE842, Spring 2009, MSU

14

About Homework 1

- Updated programming assignment is posted.
- Case should not matter in this assignment. Words in different case should be treated the same.
- The testing file is just to show you an example of the input file that may be used to test your program. (note that “UNKA” is given). But your program should be automatically identify unknown words. Change the token “UNKNOWN” to “UNKA”
- To evaluate your program, you should use the truth file. You should remove the POS tags and run your program on the original sentences and then use the provided POS tags to do the evaluation.

2/2/2009

CSE842, Spring 2009, MSU

15

Review: First-order Observable Markov Model

- a set of states $Q = 1, 2, \dots, N$, the state at time t is q_t
- Current state only depends on previous state

$$P(q_t | q_1 \dots q_{t-1}) = P(q_t | q_{t-1})$$

- Transition probability matrix A

$$a_{ij} = P(q_t = j | q_{t-1} = i) \quad 1 \leq i, j \leq N$$

- Special initial probability vector π

$$\pi_i = P(q_1 = i) \quad 1 \leq i \leq N$$

- Constraints:

$$\sum_{j=1}^N a_{ij} = 1; \quad 1 \leq i \leq N \quad \sum_{j=1}^N \pi_j = 1$$

2/2/2009

CSE842, Spring 2009, MSU

16

Markov Model for Dow Jones

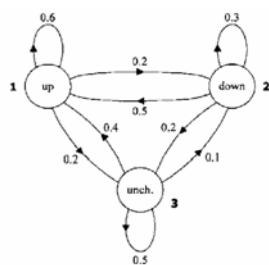


Figure from Huang et al.

Initial state probability matrix

$$\pi = (\pi_i) = \begin{pmatrix} 0.5 \\ 0.2 \\ 0.3 \end{pmatrix}$$

State-transition probability matrix

$$A = \{a_{ij}\} = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.5 & 0.3 & 0.2 \\ 0.4 & 0.1 & 0.5 \end{bmatrix}$$

2/2/2009

CSE842, Spring 2009, MSU

17

Markov Model for Dow Jones

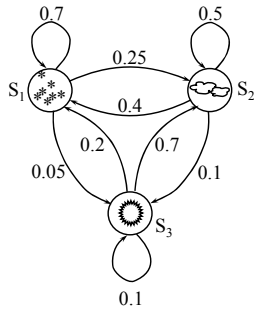
- What is the probability of 5 consecutive up days?
- Sequence is up-up-up-up-up
 - i.e., state sequence is 1 1 1 1 1
 - $P(1,1,1,1,1) =$
 - $\pi_1 a_{11} a_{11} a_{11} a_{11} = 0.5 \times (0.6)^4 = 0.0648$

2/2/2009

CSE842, Spring 2009, MSU

18

Markov Model for Winter Weather in EL



2/2/2009

CSE842, Spring 2009, MSU

19

Markov Model for Winter Weather in EL

$S_1 = \text{event}_1 = \text{snow}$
 $S_2 = \text{event}_2 = \text{clouds}$
 $S_3 = \text{event}_3 = \text{sun}$

$$A = \{a_{ij}\} = \begin{pmatrix} .70 & .25 & .05 \\ .40 & .50 & .10 \\ .20 & .70 & .10 \end{pmatrix} \quad \begin{matrix} \pi_1 = 0.5 \\ \pi_2 = 0.4 \\ \pi_3 = 0.1 \end{matrix}$$

• what is probability of {snow, snow, snow, clouds, sun, clouds, snow}?

2/2/2009

CSE842, Spring 2009, MSU

20

Markov Model for Winter Weather in EL

$S_1 = \text{event}_1 = \text{snow}$
 $S_2 = \text{event}_2 = \text{clouds}$
 $S_3 = \text{event}_3 = \text{sun}$

$$A = \{a_{ij}\} = \begin{pmatrix} .70 & .25 & .05 \\ .40 & .50 & .10 \\ .20 & .70 & .10 \end{pmatrix} \quad \begin{matrix} \pi_1 = 0.5 \\ \pi_2 = 0.4 \\ \pi_3 = 0.1 \end{matrix}$$

• what is probability of {snow, snow, snow, clouds, sun, clouds, snow}?

Obs. = {sn, sn, sn, c, sun, c, sn}
 S = {S₁, S₁, S₁, S₂, S₃, S₂, S₁}
 time = {1, 2, 3, 4, 5, 6, 7} (days)

$$\begin{aligned}
 &= P[S_1] P[S_1|S_1] P[S_1|S_1] P[S_2|S_1] P[S_3|S_2] P[S_2|S_2] P[S_1|S_2] \\
 &= 0.5 \cdot 0.7 \cdot 0.7 \cdot 0.25 \cdot 0.1 \cdot 0.7 \cdot 0.4 \\
 &= 0.001715
 \end{aligned}$$

2/2/2009

CSE842, Spring 2009, MSU

21

Summary of Markov Model

- *Conditional Independence* is assumed when computing probability of sequence of events (e.g., for first-order model).
- Each state associated with only one event (output).
- Given list of observations, it can determine exact state sequence. \Rightarrow state sequence not *hidden*.
- Computing probability of a given observation is straightforward.
- Given multiple Markov Models and an observation sequence, it's easy to determine the M.M. most likely to have generated the data.

2/2/2009

CSE842, Spring 2009, MSU

22

Review: Hidden Markov Model

- *more than 1* event are associated with each state.
- all events have some *probability* of emitting at each state.
- given a sequence of observations, we can't determine exactly the state sequence.
- We can compute the *probabilities* of different state sequences given an event/observation sequence.
- Doubly stochastic (probabilities of both emitting events and transitioning between states); exact state sequence is "hidden."

2/2/2009

CSE842, Spring 2009, MSU

23

Elements of Hidden Markov Model

- Time $t = \{1, 2, 3, \dots, T\}$
- N states $Q = \{1, 2, 3, \dots, N\}$
- M events/observations $E = \{e_1, e_2, e_3, \dots, e_M\}$
- initial probabilities $\pi_j = P[q_1 = j] \quad 1 \leq j \leq N$
- transition probabilities $a_{ij} = P[q_t = j | q_{t-1} = i] \quad 1 \leq i, j \leq N$
- Observation probabilities (*Emission probabilities*) $b_j(k) = P[o_t = e_k | q_t = j] \quad 1 \leq k \leq M$
 $b_j(o_t) = P[o_t = e_k | q_t = j] \quad 1 \leq k \leq M$
- $A =$ matrix of a_{ij} values, $B =$ set of observation probabilities, $\pi =$ vector of π_j values.

Entire Model: $\Phi = (A, B, \pi)$

2/2/2009

CSE842, Spring 2009, MSU

24

Hidden Markov Models

- a set of states $Q = 1, 2, \dots, N$, the state at time t is q_t
- Transition probability matrix $A = \{a_{ij}\}$

$$a_{ij} = P(q_t = j | q_{t-1} = i) \quad 1 \leq i, j \leq N$$

- Output probability matrix $B = \{b_i(k)\}$

$$b_i(k) = P(X_t = o_k | q_t = i) \quad 1 \leq k \leq M$$

- Special initial probability vector π

$$\pi_i = P(q_1 = i) \quad 1 \leq i \leq N$$

- Constraints:

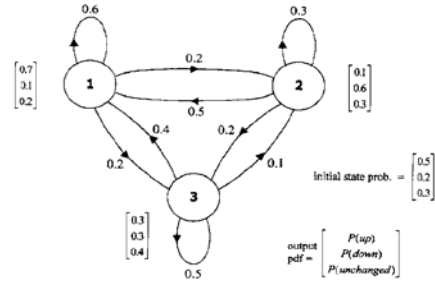
$$\sum_{j=1}^N a_{ij} = 1; \quad 1 \leq i \leq N \quad \sum_{k=1}^M b_i(k) = 1 \quad \sum_{j=1}^N \pi_j = 1$$

2/2/2009

CSE842, Spring 2009, MSU

25

HMM for Dow Jones



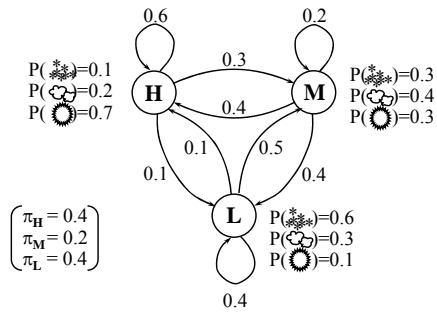
2/2/2009

CSE842, Spring 2009, MSU

From Huang et al.

26

HMM for Weather and Atmospheric Pressure



2/2/2009

CSE842, Spring 2009, MSU

27

The Three Basic Problems for HMMs

- L. R. Rabiner. 1989. A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proc IEEE 77(2), 257-286. (most relevant to our discussion: P. 257-266)

2/2/2009

CSE842, Spring 2009, MSU

28

The Three Basic Problems for HMMs

- Problem 1 (Evaluation):** Given the observation sequence $O=(o_1, o_2, \dots, o_T)$, and an HMM model $\Phi=(A, B, \pi)$, **how do we efficiently compute $P(O|\Phi)$** , the probability of the observation sequence, given the model
- Problem 2 (Decoding):** Given the observation sequence $O=(o_1, o_2, \dots, o_T)$, and an HMM model $\Phi=(A, B, \pi)$, **how do we choose a corresponding state sequence $Q=(q_1, q_2, \dots, q_T)$** that is optimal in some sense (i.e., best explains the observations)
- Problem 3 (Learning):** **How do we adjust the model parameters $\Phi=(A, B, \pi)$** to maximize $P(O|\Phi)$?

2/2/2009

CSE842, Spring 2009, MSU

From Rabiner

29

The Evaluation Problem

- Given an observation sequence O and HMM Φ , compute $P(O|\Phi)$
- Why is this hard? Sum over all possible sequences of states!

$$P(O|\Phi) = \sum_{\text{all } Q} P(O, Q|\Phi) = \sum_{\text{all } Q} P(Q|\Phi)P(O|Q, \Phi)$$

2/2/2009

CSE842, Spring 2009, MSU

30

The Evaluation Problem

With an observation sequence $O=(o_1 o_2 \dots o_T)$, state sequence $Q=(q_1 q_2 \dots q_T)$, and model Φ :

Probability of O , given state sequence Q and model Φ , is:

$$P(O|Q, \Phi) = \prod_{t=1}^T P(o_t | q_t, \Phi)$$

assuming conditional independence between observations. This expands:

$$P(O|Q, \Phi) = b_{q_1}(o_1) \cdot b_{q_2}(o_2) \dots b_{q_T}(o_T)$$

The probability of the state sequence Q can be written:

$$P(Q|\Phi) = \pi_{q_1} \cdot a_{q_1 q_2} \cdot a_{q_2 q_3} \dots a_{q_{T-1} q_T}$$

2/2/2009

CSE842, Spring 2009, MSU

31

The Evaluation Problem

- Given an observation sequence O and HMM Φ , compute $P(O|\Phi)$
- Why is this hard? Sum over all possible sequences of states!

$$P(O|\Phi) = \sum_{\text{all } Q} P(O, Q|\Phi) = \sum_{\text{all } Q} P(Q|\Phi) P(O|Q, \Phi)$$

$$= \sum_{q_1 q_2 \dots q_T} \pi_{q_1} \cdot b_{q_1}(o_1) \cdot a_{q_1 q_2} \cdot b_{q_2}(o_2) \cdot a_{q_2 q_3} \dots a_{q_{T-1} q_T} \cdot b_{q_T}(o_T)$$

- We cannot do an explicit sum over all paths because it's intractable: $O(N^T)$
- What we do instead: the **Forward Algorithm**. $O(N^2T)$

2/2/2009

CSE842, Spring 2009, MSU

32

The Forward Algorithm

- Define variable α which has meaning of "the probability of observations o_1 through o_t , and being in state i at time t , given our HMM Φ "

$$\alpha_t(i) = P(o_1 o_2 \dots o_t, q_t = i | \Phi)$$

Compute α and $P(O|\Phi)$ with the following procedure:

$$\alpha_t(i) = \pi_i \cdot b_i(o_t) \quad 1 \leq i \leq N$$

Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) \cdot a_{ij} \right] \cdot b_j(o_{t+1}) \quad \begin{matrix} 1 \leq t \leq T-1 \\ 1 \leq j \leq N \end{matrix}$$

Termination:

$$P(O|\Phi) = \sum_{i=1}^N \alpha_T(i)$$

2/2/2009

CSE842, Spring 2009, MSU

33

$\alpha_t(i)$: easy to define recursively

$$\alpha_1(i) = P(o_1 \wedge q_1 = S_i)$$

$$= P(q_1 = S_i) P(o_1 | q_1 = S_i)$$

$$= \pi_i b_i(o_1)$$

$$\alpha_{t+1}(j) = P(o_1 o_2 \dots o_t o_{t+1} \wedge q_{t+1} = S_j)$$

$$= \sum_{i=1}^N P(o_1 o_2 \dots o_t \wedge q_t = S_i \wedge o_{t+1} \wedge q_{t+1} = S_j)$$

$$= \sum_{i=1}^N P(o_{t+1}, q_{t+1} = S_j | o_1 o_2 \dots o_t \wedge q_t = S_i) P(o_1 o_2 \dots o_t \wedge q_t = S_i)$$

$$= \sum_{i=1}^N P(o_{t+1}, q_{t+1} = S_j | q_t = S_i) \alpha_t(i)$$

$$= \sum_{i=1}^N P(q_{t+1} = S_j | q_t = S_i) P(o_{t+1} | q_{t+1} = S_j) \alpha_t(i)$$

$$= \sum_{i=1}^N a_{ij} b_j(o_{t+1}) \alpha_t(i)$$

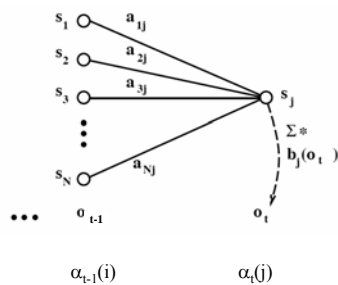
2/2/2009

CSE842, Spring 2009, MSU

34

The Inductive Step

- Computation of $\alpha_t(j)$ by summing all previous values $\alpha_{t-1}(i)$ for all i



2/2/2009

CSE842, Spring 2009, MSU

35

Easy Questions

We can efficiently compute

$$\alpha_T(i) = P(o_1 o_2 \dots o_T \wedge q_T = i)$$

How to efficiently compute

$$P(o_1 o_2 \dots o_T) ?$$

How to efficiently compute

$$P(q_t = i | o_1 o_2 \dots o_T)$$

2/2/2009

CSE842, Spring 2009, MSU

36

Easy Questions

We can efficiently compute

$$\alpha_T(i) = P(O_1 O_2 \dots O_T \wedge q_T = i)$$

How to efficiently compute

$$P(O_1 O_2 \dots O_T) \quad ? \quad \sum_{i=1}^N \alpha_T(i)$$

How to efficiently compute

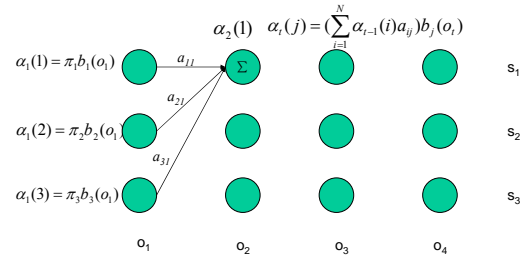
$$P(q_T = i | O_1 O_2 \dots O_T) = \frac{\alpha_T(i)}{\sum_{j=1}^N \alpha_T(j)}$$

2/2/2009

CSE842, Spring 2009, MSU

37

Forward Algorithm



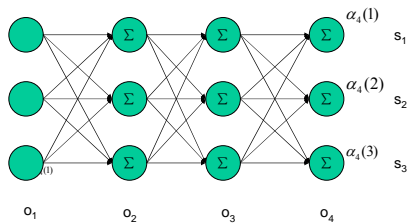
2/2/2009

CSE842, Spring 2009, MSU

38

Forward Algorithm

$$P(o_1 o_2 o_3 o_4 | \Phi) = \alpha_4(1) + \alpha_4(2) + \alpha_4(3)$$



2/2/2009

CSE842, Spring 2009, MSU

39