CS6501: Natural Language Processing

Quiz 2

October 11th, 2016

This is a closed book exam. Everything you need in order to solve the problems is supplied in the body of this exam.

This exam booklet contains three problems.

The exam ends at 1:05 PM. You have 30 minutes to earn a total of 25 points.

Besides having the correct answer, being concise and clear is very important. For full credit, you must show your work and explain your answers.

Good Luck!

Name (or Computing ID): (1 Point)

<table>
<thead>
<tr>
<th>Short Questions</th>
<th>/8</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Viterbi Algorithm</td>
<td>/10</td>
</tr>
<tr>
<td>EM</td>
<td>/6</td>
</tr>
<tr>
<td>Total</td>
<td>/25</td>
</tr>
</tbody>
</table>
Short Questions [8 points]

(a) [2 points] Give a word with more than two potential POS tags (show the word and provide two sample sentences.)

i. The **back** door is open, in case if you want to leave (used as adjective).
ii. He promised to **back** me (used as verb).

(b) [3 points] Name an NLP task (besides POS tagging) that uses sequential tagging technique. Describe the input, the output and the tagset of the task.

**Named Entity Recognition (NER):** Identify all mentions of named entities (people, organizations, locations, dates)

**Input:** a sentence (e.g. Pierre Vinken, 61 years old, will join IBM’s board as a nonexecutive director on November 29.)

**Output:** all name entities (e.g., [PERS Pierre Vinken], 61 years old, will join [ORG IBM]’s board as a nonexecutive director on [DATE November 29]).

**Tagset:**
- B-PERS, B-DATE, ......: beginning of a mention of a person/date...
- I-PERS, I-DATE, ......: inside of a mention of a person/date...
- O: outside of any mention of a named entity

(c) [3 points] Provides three differences between generative and discriminative models.

i. Generative models estimate joint distribution but discriminative models estimate conditional distribution.

ii. In generative models, dependence assumption has to be specified for $P(w|t)$ and $P(t)$ but in discriminative models, arbitrary features can be incorporated for modeling $P(t|w)$.

iii. Generate models can be used in unsupervised learning but discriminative models require labeled data, suitable for (semi-) supervised learning.
The Viterbi Algorithm [10 points]

The following is a pseudocode of Viterbi algorithm, where \( \text{sent} \) is a word array, \( \text{model} \) is a structure contains three probability tables, and \( \text{trellis} \) is a temporary array. \( T \) and \( N \) are the numbers of tags and words in the sentence, respectively.

1: for \( t = 1 \ldots T \) do
2: \( \text{trellis}[1][t] = \text{model.init}[t] \times \text{model.emit}[t][\text{sent}[1]] \)
3: end for
4: for \( i = 2 \ldots N \) do
5: for \( t = 1 \ldots T \) do
6: \( \text{trellis}[i][t] = 0 \)
7: for \( s = 1 \ldots T \) do
8: \( \text{trellis}[i][t] = \max(\text{trellis}[i][t], \text{trellis}[i-1][s] \times \text{model.trans}[s][t]) \)
9: end for
10: \( \text{trellis}[i][t] = \text{trellis}[i][t] \times \text{model.emit}[t][\text{sent}[i]] \)
11: end for
12: end for
13: \( \text{output} = 0 \)
14: for \( t = 1 \ldots T \) do
15: \( \text{output} = \max(\text{output}, \text{trellis}[N][t]) \)
16: end for
17: return \( \text{output} \)

(a) [2 points] What is the output of the above algorithm? (explain in English)

Output of the above algorithm is the probability of the best tag sequence given the word array.

(b) [2 points] Please provide one reason to support the following statement:

“In practice, it is often better to compute likelihood and probability values in log space.”

Computing likelihood and probability values in log space solves the underflow problem. Moreover, when computing in log space, we can replace the multiplication operations by addition operations, which are usually faster than multiplication operations on a modern computer.
(c) [3 points] Let

\[
\text{model2.init}[t] = \log(\text{model.init}[t]),
\text{model2.emit}[t][i] = \log(\text{model.emit}[t][i]),
\text{model2.trans}[s][t] = \log(\text{model.trans}[s][t]).
\]

Fill in the blanks (in lines 2, 8, and 10) and complete the pseudocode of Viterbi algorithm that computes values in log space.

1: for \(t = 1 \ldots T\) do

2: \(\text{trellis}[1][t] = \text{model2.init}[t] + \text{model2.emit}[t][\text{sent}[1]]\)

3: end for

4: for \(i = 2 \ldots N\) do

5: for \(t = 1 \ldots T\) do

6: \(\text{trellis}[i][t] = 0\)

7: end for

8: \(\text{trellis}[i][t] = \max(\text{trellis}[i][t], \text{trellis}[i-1][s] + \text{model2.trans}[s][t])\)

9: end for

10: \(\text{trellis}[i][t] = \text{trellis}[i][t] + \text{model2.emit}[t][\text{sent}[i]]\)

11: end for

12: end for

13: \(\text{output} = 0\)

14: for \(t = 1 \ldots T\) do

15: \(\text{output} = \max(\text{output}, \text{trellis}[N][t])\)

16: end for

17: return \(\exp(\text{output})\)

(d) [3 points] If we know the POS tag of the second word in a sentence is always “noun” (you can assume the corresponding tag index is 1), describe how we can modify the above Viterbi algorithm to find the best tag sequence under this constraint. (You can describe the algorithm either in original space or in log space.)

Multiple solutions are possible for the scenario. One possible solution is to add the following code snippet after line 10:

10.1: if \(i == 2\) and \(t! = 1\):

10.2: \(\text{trellis}[i][t] = 0\) (or \(\text{trellis}[i][t] = -\infty\) in log space)
The EM Algorithm [6 points]

Prof. Chang likes to give students chocolates in class. Students observe that he gives different numbers of chocolates based on the levels of difficulty of lectures. (a psychologist investigated this and had no clue about the association). In Fall 2016, a student, named Henry Parker, decided to study this mystery. He sneaked into Prof. Chang’s office (it is not recommended to do so) and found a piece of paper shown below:

<table>
<thead>
<tr>
<th>Lec 1</th>
<th>Lec 2</th>
<th>Lec 3</th>
<th>lec 4</th>
<th>lec 5</th>
<th>lec 6</th>
<th>lec 7</th>
<th>lec 8</th>
<th>lec 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>E</td>
<td>H</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>H</td>
<td>E</td>
<td>E</td>
</tr>
</tbody>
</table>

All the lectures after lec 9 will be hard.

He suddenly realized that it shows the hardness of lectures in Prof. Chang’s NLP class, where “E” means the lecture is easy, and “H” means the lecture is hard. Unfortunately, Prof. Chang spilled his coffee on the paper, so the hardness of lectures 4-6 is unknown.

Henry also wrote down the numbers of chocolates Prof. Chang gave at each lecture until lecture 9.

<table>
<thead>
<tr>
<th>Lec 1</th>
<th>Lec 2</th>
<th>Lec 3</th>
<th>lec 4</th>
<th>lec 5</th>
<th>lec 6</th>
<th>lec 7</th>
<th>lec 8</th>
<th>lec 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>7</td>
<td>11</td>
<td>11</td>
<td>7</td>
<td>11</td>
<td>11</td>
<td>7</td>
<td>11</td>
</tr>
</tbody>
</table>

Apparently, Prof. Chang loves prime numbers. Based on these numbers, Henry assumes Prof. Chang gives either 7 or 11 chocolates each class. Then, Henry modeled Prof. Chang’s behavior using an HMM model.

(a) [2 points] What are the model parameters that Henry has to figure out? (Hint: there are three types of model parameters. Name them and briefly describe what they mean.)

- Transition probabilities: $P(E|H), P(E|E), P(H|E), P(H|H)$
- Emission probabilities: $P(7|E), P(11|E), P(7|H), P(11|H)$
- Initial probabilities: $P(E|<s>), P(H|<s>)$ where $<s>$ represents the start symbol.
Henry decided to use hard EM to evaluate these model parameters. This algorithm is simple and he doesn’t need to use forward and backward algorithms to compute the expected counts.

The first step in the hard EM is to guess the missing labels. Because Henry thinks NLP is hard, he guessed Lecture 4-6 are hard.

(b) [2 points] Estimate the model parameters of HMM based on the observations in lectures 1-9. When estimating the transition probability, you can assume the last transition (from lec 9 to lec 10) is E-H.

- Transition probabilities:  \( P(E|H) = \frac{1}{5}, \ P(E|E) = \frac{2}{4}, \ P(H|E) = \frac{2}{4}, \ P(H|H) = \frac{4}{5} \)
- Emission probabilities:  \( P(7|E) = \frac{3}{4}, \ P(11|E) = \frac{1}{4}, \ P(7|H) = \frac{1}{5}, \ P(11|H) = \frac{4}{5} \)
- Initial probabilities:  \( P(E|<s>) = 1.0, \ P(H|<s>) = 0.0 \)

(c) [2 points] Based on the parameters you obtained in (b), what are the most likely difficulty levels of Lecture 4-6? Are they consistent with Henry’s guess?

The most likely difficulty levels of lecture 4-6 is H-H-H and yes it is consistent with Henry’s guess.

(Some students performed the Viterbi algorithm over lecture 1-9 to find the most likely difficulty levels. Full credit will be given if the answer and the calculations are correct.)