CSE 325/425 (Spring 2021) Homework 5

Due on 11:55pm, Apr 21, 2021

Grading: All questions have the same points (25 each). We will randomly grade some of the questions.

Submitting: Only electronic submissions on Coursesite are accepted. You can handwrite your answers on papers and then scan them to images. If you need to plot figures using a computer, the plotted files should be saved and included in the submitted pdf file. Submit a single pdf file named

<Your LIN>HW5.pdf

Other format will not be accepted.

Questions:

1. Assume the following PCFG:

```
S -> a (with probability 1/3)
S -> Sa (with probability 2/3)
```

where a is a terminal and S is the only non-terminal. Prove that the total probability of all sentences generated by the PCFG is 1.
(*Hints: you will need to find all sentences, each of which has a probability of being generated. Then sum up the probabilities and show that the sum is 1.*)

[[[The sentences generated by the PCFG are <==

- a (with probability $\frac{1}{3}$)
- aa (with probability $\frac{2}{3}\frac{1}{3}$)
- aaa (with probability $\left(\frac{2}{3}\right)^2\frac{1}{3}$)
- ...
- $(a)^k$ (with probability $\left(\frac{2}{3}\right)^{(k-1)}\frac{1}{3}$
- ...

The total probability is

$$\frac{1}{3}\left(1+\frac{2}{3}+\left(\frac{2}{3}\right)^2+\cdots+\left(\frac{2}{3}\right)^{(k-1)}+\dots\right)=\frac{1}{3}\left(1-\frac{2}{3}\right)^{-1}=1. \quad (1)$$

]]]

2. Prove that the two parsing trees will have the same probability under the same PCFG.

[[[The two trees use exactly the same set of rules from the given PCFG. According to the independence assumptions in PCFG when generating a parsing tree, the probability of a tree is the product of the probabilities of the rules used to generate the tree. The same PCFG will assign the same probabilities to these rules and therefore the two trees will have the same probability.]]] <==

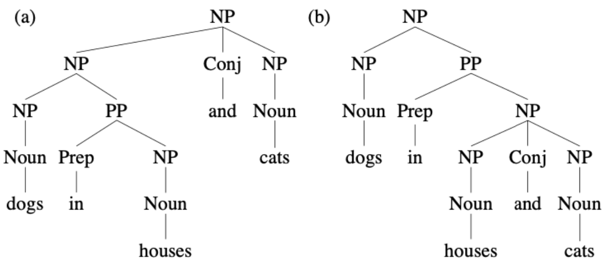

Figure 1: Two parsing trees for the input "*dogs in houses and cats*"

3. Analyze the time complexity of the CYK algorithm running on a sentence of length m with a CFG with N non-terminals. Use the Big-O notation. Answers on specific sentence or CFG will not be accepted. (*Hints: run the CYK algorithm on some simple sentence and CFG to get a sense of how much computation you need.*)

   [[[As all dividers for the range $[p, q]$ represented by the cell (p, q) need to be enumerated, this gives a $O(q - p)$ factor, for any $0 \leq p < q \leq m$. To be more detailed, <==

$$m + (m-1) + (m-2) * 2 + (m-3) * 3 + \cdots + (m - (m-1)) * (m-1) \tag{2}$$

   The first m arises when $p = q - 1$ for $0 \leq p < m$, covering single words.
   The $(m-1)$ arises when $p = q - 2$ for $0 \leq p < m - 1$, covering two consecutive words, with one option of the divider.
   $(m-2) * 2$ arises when $p = q - 3$ for $0 \leq p < m - 2$ with 2 possibilities of the divider of the range $[p, q]$, covering three consecutive words.
   And so on, until $(m - (m-1)) * (m-1)$ for $p = 0$ and $q = m$ covering the whole m words, with $m - 1$ possible dividers.
   The overall complexity is thus

$$\begin{aligned} m + \sum_{k=1}^{m-1} (m-k)k &= m + m \times (1 + 2 + \cdots + (m-1)) - (1^2 + 2^2 + \cdots + (m-1)^2) \qquad (3) \\ &= m + m \times \frac{m(m-1)}{2} - \frac{(m-1)(m)(2(m-1)+1)}{6} \qquad (4) \\ &= O(m^3) \qquad (5) \end{aligned}$$

   For each cell (p, q) in the matrix, except when $p = q - 1$, all possible non-terminals need to be tried for each divider, giving an $O(N)$ factor for each cell and each divider.

   Overall, the time complexity is $O(Nm^3)$.]]]

4. Explain in what order should the CYK matrix be filled out when calculating the outside probabilities using dynamic programming.

   [[[In the base case, the top-right corner $\alpha_j(1, m)$ will be filled first. Then by visiting rows from top to bottom, and on each row, from right to left, fill the cells. The reason is that, in the following equation <==

$$\begin{aligned} \alpha_j(p, q) &= \left[\sum_{f,g} \sum_{e=q+1}^{m} \alpha_f(p, e) \Pr(N^f \to N^j N^g) \beta_g(q+1, e) \right] \qquad (6) \\ &+ \left[\sum_{f,g} \sum_{e=1}^{p-1} \alpha_f(e, q) \Pr(N^f \to N^g N^j) \beta_g(e, p-1) \right] \qquad (7) \end{aligned}$$

   we need $\alpha_f(p, e)$, $e = q+1, \ldots, m$, which are on the right of the cell (p, q), and $\alpha_f(e, q)$, $e = 1, \ldots, p-1$, which are above the cell (p, q).]]]

5. In the algorithm that finds the most likely parse tree for a sentence, explain where and what information needs to be stored to reconstruct the tree.

   [[[In each cell (p, q) of the CYK matrix, we need to store the tuple (j, k, r) for each non-terminal N^i, where j and k are the non-terminals of the left and right children under the parent node N^i and r is the best position to divide the range $[p, q]$ into $[p, r]$ for the left subtree and $[r + 1, q]$ for the right subtree.]]] <==