XCS221   Assignment 2

JUAN RICARDO PEDRAZA ESCOBAR

Artificial Intelligence: Principles and Techniques

Stanford Center for Professional Development

November,2021

1.a

We can construct an example as:

tryingtounderstandsearchProblems

and we stablish the following cost set:

[try: 5, ying: 100, rying: 180 tryi: 100, to: 5, understand: 6, understands: 5, ing: 120, search: 5, searchP: 5, searc: 100, Problems: 5, Pro: 4, Problem: 5, Proble: 130, Probl: 10].

For this example, the greedy algorithm would pick *try* rather than *tryi* since the weights of try is only 5 and *tryi* is 100. Then the algorithm will pick *ying* rather than *rying* since the string one has lower weights. Therefore, we will have a false segmentation and we can say that the greedy algorithm is suboptional.

2.a

If we have the following example,

cp,s,dwn

with a bigram cost function as,

cost (-BEGIN-,cup)=1   cost(-BEGIN-,cap)=800

cost(cup,is)=9000 cost(cap,is)=1

cost(is,down)=1

The greedy algorithm would return *cup is down* for cost 9002, while the optimal path is *cap is down* for cost 802. It means that the algorithm is sub optional.

3.a

If $x_1, \dots, x_k$ are the input's characters, we can note each state as $s = (w', i, j)$ where we use the previous word, the current string position and the current position.

Then we have two possible actions, ending the word so creating a new word from i to j (END) or incrementing j and continue, if $j < k$ (CONT). Therefore, our costs for ich possible actions are:

$$Cost\left((w', i, j), (END)\right) = b(w', x_{i,j}) \quad , \quad Cost\left((w', i, j), (CONT)\right) = 0$$

Note that our final state is $s = (w', k, k)$ and the initial state is $s = (-BEGIN-, 0, 0)$.

3.c

if we had the following bigram cost function and we take account the next definition,

$$u_b(is) = \min b(w', w)$$

- where is 0.6
- when is 1.0
- who is 0.8
- it was 0.9
- which was 9.0

then,

$$u_b(is) = min((where, is), (when, is), (who, is)) = 0.6$$

In the same way,

$$u_b(was) = 0.9$$

It is clear that $Cost'(s, a) \leq Cost(s, a)$ for all states s and actions a. Also the states, actions, cost, start state, and end test have the same behavior of part *3a* of this assignment.

Now we can define $h(s) = FutureCost'(s)$.

3.d

Is UCS a special case of A*? Explain why or why not.

➤ Yes, it is a special case. Is the case when A* search has the function $h(n) = 0$.

Is BFS a special case of UCS? Explain why or why not.

➤ Yes, it is a special case. Is the case when all step costs are equal, so UCS reproduces BFS.