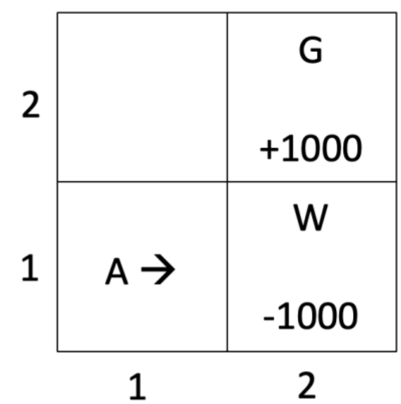
Washington State University
School of Electrical Engineering and Computer Science
Fall 2019

CptS 440/540 Artificial Intelligence
Homework 11 - Solution
Due: November 21, 2019 (11:59pm)

General Instructions: Put your answers to the following problems into a PDF document and submit as an attachment under Content → Homework 11 for the course CptS 440 Pullman (all sections of CptS 440 and 540 are merged under the CptS 440 Pullman section) on the Blackboard Learn system by the above deadline. Note that you may submit multiple times, but we will only grade the most recent entry submitted before the above deadline.



1. Consider the 2x2 Wumpus world on the right, where the agent starts in (1,1) facing Right, the Wumpus is in (2,1), and the gold is in (2,2). We will represent the agent's state as [*x*,*y*,*o*], where (*x*,*y*) is the agent's location, and *o* is the agent's orientation (Up, Down, Left, Right). The agent has three possible actions: TurnLeft, TurnRight and GoForward. GoForward always works (i.e., moves the agent to the location it is facing, or bumps into a wall and stays in the same location). However, TurnLeft and TurnRight only work 80% of the time; the other 20% of the time the agent's orientation does not change. If the agent enters terminal state [2,2,*], it receives reward +1000. If the agent enters terminal state [2,1,*], it receives reward –1000. The agent receives a reward of –1 for all other states.

   a. Given the following policy, compute the utilities of each non-terminal state, using the equation on slide 58 of the lecture notes, where $\gamma = 0.9$. Show your work.

State	**Action**
[1,1,Right]	TurnLeft
[1,1,Up]	GoForward
[1,1,Left]	TurnRight
[1,1,Down]	TurnRight
[1,2,Right]	GoForward
[1,2,Up]	TurnRight
[1,2,Left]	TurnRight
[1,2,Down]	TurnLeft

   b. Using temporal difference Q-learning (equation on slide 64 of lecture notes), compute the Q values for Q([1,1,Right],TurnLeft), Q([1,1,Up],GoForward), Q([1,2,Up],TurnRight), Q([1,2,Right],GoForward), after each of five executions of the action sequence: TurnLeft, GoForward, TurnRight, GoForward (starting from [1,1,Right] for each sequence). You may assume $\alpha = 1$, $\gamma = 0.9$, and all Q values for non-terminal states are initially zero. Show your work.

Solution:

a. Utility equations:

U([1,1,Right]) = R([1,1,Right]) + (0.9)*{ (0.8)*U([1,1,Up]) + (0.2)*U([1,1,Right]) }
U([1,1,Up]) = R([1,1,Up]) + (0.9)*{ (1.0)*U([1,2,Up]) }
U([1,1,Left]) = R([1,1,Left]) + (0.9)*{ (0.8)*U([1,1,Up]) + (0.2)*U([1,1,Left]) }
U([1,1,Down]) = R([1,1,Down]) + (0.9)*{ (0.8)*U([1,1,Left]) + (0.2)*U([1,1,Down]) }
U([1,2,Right]) = R([1,2,Right]) + (0.9)*{ (1.0)*U([2,2,Right]) }
U([1,2,Up]) = R([1,2,Up]) + (0.9)*{ (0.8)*U([1,2,Right]) + (0.2)*U([1,2,Up]) }
U([1,2,Left]) = R([1,2,Left]) + (0.9)*{ (0.8)*U([1,2,Up]) + (0.2)*U([1,2,Left]) }
U([1,2,Down]) = R([1,2,Down]) + (0.9)*{ (0.8)*U([1,2,Right]) + (0.2)*U([1,2,Down]) }

Solving analytically:

U([1,2,Right]) = (-1) + (0.9)*{ (1.0)*1000 } = 899
U([1,2,Up]) = (-1) + (0.9)*{ (0.8)*(899) + (0.2)*U([1,2,Up]) }
= (-1) + 647.28 + (0.18)*U([1,2,Up])
= 646.28 / (1 – 0.18) = 788
U([1,2,Down]) = (-1) + (0.9)*{ (0.8)*(899) + (0.2)*U([1,2,Down]) }
= (-1) + 647.28 + (0.18)*U([1,2,Down])
= 646.28 / (1 – 0.18) = 788
U([1,2,Left]) = (-1) + (0.9)*{ (0.8)*(788) + (0.2)*U([1,2,Left]) }
= (-1) + 567.36 + (0.18)*U([1,2,Left])
= 566.36 / (1 – 0.18) = 691
U([1,1,Up]) = (-1) + (0.9)*{ (1.0)*(788) } = 708
U([1,1,Right]) = (-1) + (0.9)*{ (0.8)*(708) + (0.2)*U([1,1,Right]) }
= (-1) + 509.76 + (0.18)*U([1,1,Right])
= 508.76 / (1 – 0.18) = 620
U([1,1,Left]) = (-1) + (0.9)*{ (0.8)*(708) + (0.2)*U([1,1,Left]) }
= (-1) + 509.76 + (0.18)*U([1,1,Left])
= 508.76 / (1 – 0.18) = 620
U([1,1,Down]) = (-1) + (0.9)*{ (0.8)*(620) + (0.2)*U([1,1,Down]) }
= (-1) + 446.4 + (0.18)*U([1,1,Left])
= 445.4 / (1 – 0.18) = 543

Solving iteratively (utilities start at 0; updated only after all equations calculated, i.e., batch learning):

Iteration	U([1,1,Right])	U([1,1,Up])	U([1,1,Left])	U([1,1,Down])	U([1,2,Right])	U([1,2,Up])	U([1,2,Left])	U([1,2,Down])
1	-1	-1	-1	-1	899	-1	-1	-1
2	-2	-2	-2	-2	899	646	-2	646
3	-3	580	-3	-3	899	763	464	763
4	416	686	416	-4	899	784	632	784
5	568	705	568	298	899	787	677	787
6	609	707	609	462	899	788	688	788
7	618	708	618	521	899	788	690	788
8	620	708	620	538	899	788	691	788
9	620	708	620	542	899	788	691	788
10	620	708	620	543	899	788	691	788
11	620	708	620	543	899	788	691	788

b. Q-learning equations (note: when α=1, the Q(s,a)'s in the equation cancel out):

After observing [1,1,Right] $\rightarrow$ TurnLeft $\rightarrow$ [1,1,Up]:

Q([1,1,Right],TurnLeft) = R([1,1,Right]) + (0.9) $\max_{a'}$ Q(([1,1,Up], a')

= (-1) + (0.9) max{Q([1,1,Up],TurnLeft),Q([1,1,Up],TurnRight),Q([1,1,Up],GoForward)}

After observing [1,1,Up] $\rightarrow$ GoForward $\rightarrow$ [1,2,Up]:

Q([1,1,Up],GoForward) = R([1,1,Up]) + (0.9) $\max_{a'}$ Q(([1,2,Up], a')

= (-1) + (0.9) max{Q([1,2,Up],TurnLeft),Q([1,2,Up],TurnRight),Q([1,2,Up],GoForward)}

After observing [1,2,Up] $\rightarrow$ TurnRight $\rightarrow$ [1,2,Right]:

Q([1,2,Up],TurnRight) = R([1,2,Up]) + (0.9) $\max_{a'}$ Q(([1,2,Right], a')

= (-1) + (0.9) max{Q([1,2,Right],TurnLeft),Q([1,2,Right],TurnRight),Q([1,2,Right],GoForward)}

After observing [1,2,Right] $\rightarrow$ GoForward $\rightarrow$ [2,2,Right]:

Q([1,2,Right],GoForward) = R([1,2,Right]) + (0.9) $\max_{a'}$ Q(([2,2,Right], a')

= (-1) + (0.9) (1000) = 899

Iteration	Q([1,1,Right],TurnLeft)	Q([1,1,Up],GoForward)	Q([1,2,Up],TurnRight)	Q([1,2,Right],GoForward)
1	-1	-1	-1	899
2	-1	-1	808	899
3	-1	726	808	899
4	653	726	808	899
5	653	726	808	899

2. Given the following bigram model, compute the probability of the sentence "the agent ate the wumpus". Show your work.

Word 1	**Word 2**	**Frequency**
the	wumpus	1,000
wumpus	ate	500
ate	the	10,000
the	agent	5,000
agent	ate	100
agent	shot	500

Solution:

P(sentence) = P(agent | the) * P(ate | agent) * P(the | ate) * P(wumpus | the)

P(agent | the) = P(the agent) / P(the) = (5000/17100) / (6000/17100) = 5/6 = 0.833

P(ate | agent) = P(agent ate) / P(agent) = (100/17100) / (600/17100) = 1/6 = 0.167

P(the | ate) = P(ate the) / P(ate) = (10000/17100) / (10000/17100) = 1.0

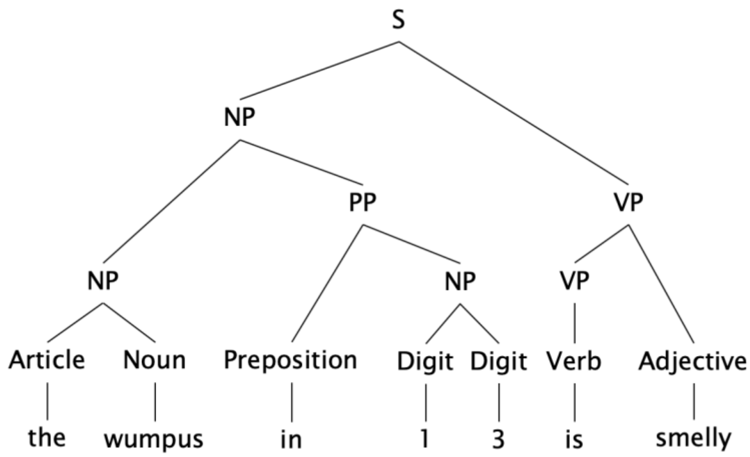
P(wumpus | the) = P(the wumpus) / P(the) = (1000/17100) / (6000/17100) = 1/6 = 0.167

P(sentence) = (5/6)*(1/6)*(1)*(1/6) = <u>0.023</u>

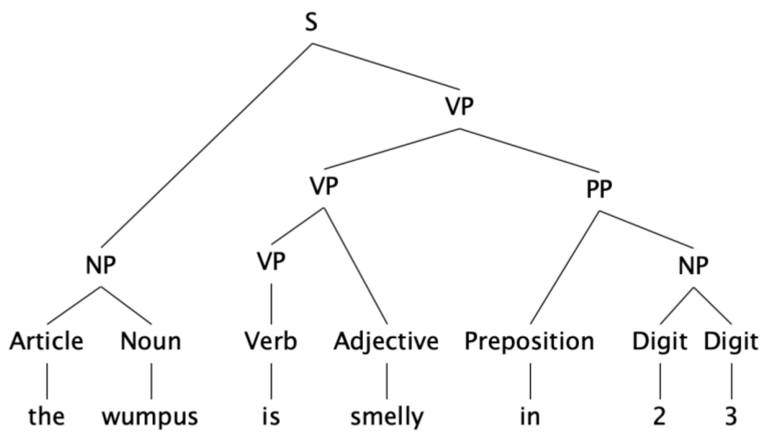
3. Given the lexicon on slide 23 of the lecture notes and the grammar on slide 24 of the lecture notes, show all possible parse trees of each of the following sentences. If there is no parse, then just state "No parse".
   a. "the wumpus in 1 3 is smelly"
   b. "the wumpus is smelly in 2 3"
   c. "the wumpus and the agent eat"

Solution:

a. Parse tree below.



b. Parse tree below.



c. No parse.

4. Given the HMM for the [m] phoneme on slide 40 of the lecture notes, compute the probability of each possible path through the HMM for the sequence of frame features C_1,C_3,C_4,C_6. Show your work.

Solution: There are three possible paths through the HMM:

Onset, Onset, Mid, End, Final

P(path) = P(C_1|Onset)*P(Onset|Onset)*P(C_3|Onset)*P(Mid|Onset)*P(C_4|Mid)*P(End|Mid)*P(C_6|End)*P(Final|End)
= (0.5)(0.3)(0.3)(0.7)(0.7)(0.1)(0.5)(0.6) = 0.0006615

Onset, Mid, Mid, End, Final

P(path) = P(C_1|Onset)*P(Mid|Onset)*P(C_3|Mid)*P(Mid|Mid)*P(C_4|Mid)*P(End|Mid)*P(C_6|End)*P(Final|End)
= (0.5)(0.7)(0.2)(0.9)(0.7)(0.1)(0.5)(0.6) = 0.001323

Onset, Mid, End, End, Final

P(path) = P(C_1|Onset)*P(Mid|Onset)*P(C_3|Mid)*P(End|Mid)*P(C_4|End)*P(End|Mid)*P(C_6|End)*P(Final|End)
= (0.5)(0.7)(0.2)(0.1)(0.1)(0.4)(0.5)(0.6) = 0.000084

5. *CptS 540 Students Only*. The Stanford Parser is available at nlp.stanford.edu:8080/parser/. For each of the sentences in problem 3, show the parse tree obtained by the Stanford Parser. You may just copy-and-paste the Parse result into your homework submission; no need to draw the parse tree. But make sure the indentation is preserved.

Solution:

a. Parse tree below.

```
(ROOT
  (S
    (NP
      (NP (DT the) (NNS wumpus))
      (PP (IN in)
        (NP
          (QP (CD 1) (CD 3)))))
    (VP (VBZ is)
      (ADJP (JJ smelly)))))
```

b. Parse tree below.

```
((ROOT
  (S
    (NP (DT the) (NNS wumpus))
    (VP (VBZ is)
      (ADJP (JJ smelly))
      (PP (IN in)
        (NP
          (QP (CD 2) (CD 3)))))))
```

c. Parse tree below.

```
(ROOT
  (S
    (NP
      (NP (DT the) (NNS wumpus))
      (CC and)
      (NP (DT the) (NN agent)))
    (VP (VBP eat))))
```