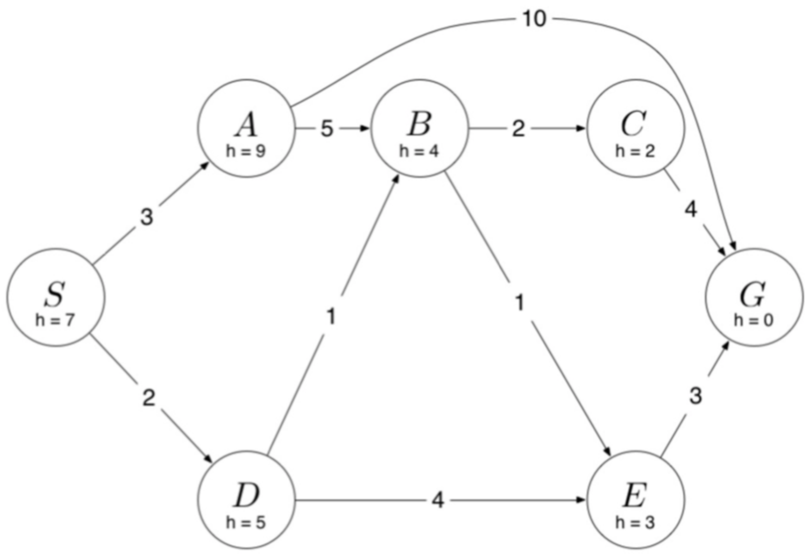
CS 480 Artificial Intelligence **Final Examination Solutions** **Spring 2020**

May 13, 2020, Duration: 11 AM to 2 PM

1. Shown below is a weighted directed graph with S as starting vertex and G as the goal. In all the search strategies, the algorithm terminates when the goal vertex is removed from the fringe. For BFS and DFS, alphabetical order is used for exploring the children. Similarly, DeleteMin is priority queue will use alphabetical order to break ties. Also the algorithm uses tree search (not graph search) which means the same node can enter/leave the open set (which could be a queue/stack/priority queue) multiple times. (Note: In (i) to (iv), the question is NOT asking the path from S to G found by the algorithm. It is asking you to list the order in which vertices leave the open set.)



(i) Write the sequence of states that leave the open set (fringe) if BFS is used. Answer the same question for DFS.

BFS: SADBG

DFS: SABCG

(ii) Write the sequence of states that leave the open set if greedy search is used.

Greedy: SDEG

(iii) Write the sequence of states that leave the open set if uniform-cost search is used. Uniform-cost-search: SDABECEG

(iv) Write the sequence of states that leave the open set if A^* search (with h given in the figure above) is used.

A* search: SDBCEG

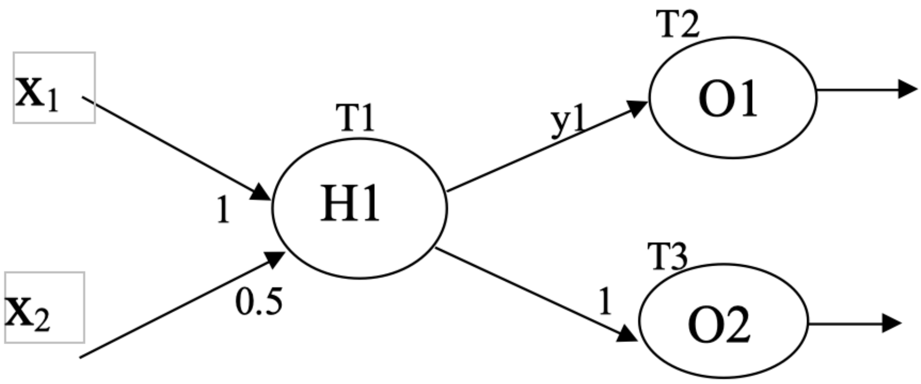
(v) Suggest a heuristic function h_1 that will result in the path $SDEG$ being output by the best first search algorithm using h_1. Your answer should assign a real number value $h_1(x)$ for each node x. Show your answer as a table.

$h_1(S) = h_1(D) = h_1(E) = h_1(G) = 0$ and $h_1(A) = h_1(B) = h_1(c) = 100$

(vi) Modify the value of h in as few nodes as possible so that it becomes non-monotonic, but still remains consistent.

Just one change is enough: $h(C) = 1$

2. Consider the neural network N shown below with one hidden node H_1 and two output nodes O_1 and O_2. In this problem, each neuron has a threshold T and it outputs 1 (or 0) if the dot product of its weight vector and the input vector is greater than or equal to 0 (less than 0). i.e., No sigmoid or RELU is used. The training data contains two inputs $(0, 1)$ and $(1, 0)$ with the following outputs: for the input $(1, 0)$, the target output is $(0, 1)$; for the input $(0, 1)$, the target output is $(1, 0)$.



(a) Provide a set of values for the weights and thresholds for the neurons H_1, O_1 and O_2 that would allow the network to behave in this manner. Show that the correct output is achieved for each of the two input vectors by exhibiting the weights and thresholds of the three neurons. Some of the weights are already given. Determine the values of T_1, T_2, T_3 and y_1.

One set of weightw is: $T_1 = 0.75$, $T_2 = -0.5$ $T_3 = 0.5$, $y_1 = -1$.

(b) Fix the weights as determined in (a). Now suppose the thresholding of output of a neuron is replaced by a RELU. So, instead of replacing positive output by 1 and negative by 0, each neuron sends the positive output unchanged, but negative output is replaced by 0. A RELU is applied to the output of all the three neurons H_1, O_1 and O_2. Also following the RELU of the output neurons, suppose softmax is applied. What are the outputs generated by O_1 and O_2 for the input $(0, 1)$? (Note that the answer should be two real numbers p_1, p_2 such that $0 \leq p_1, p_2 \leq 1$ and $p_1 + p_2 = 1$.)

For the weights above, $p_1 = \frac{1}{1+e^{0.5}}$, $p_2 = p_1 = \frac{e^{0.5}}{1+e^{0.5}}$

(c) For the weights you choose in (a), consider the test case input $(0, 0)$. Now consider the version of the network in which the threshold is replaced by sigmoid function in each neuron. What is the output pair (y_1, y_2) computed by the network. (Recall that the sigmoid function is defined as: $\sigma(x) = \frac{e^{-x}}{1+e^{-x}}$.)

$y_1 = \frac{e^{0.5}}{1+e^{0.5}}$, $y_2 = \frac{e^{-0.5}}{1+e^{-0.5}}$

3. Shown below is a small training set S with ten instances, two features and two class labels 0 and 1. S is presented as triples (x_1, x_2, t) where x_1 and x_2 are the feature values and t is the class label.

   $S = \{(1,2,1),(2,0,1),(2,3,1),(-1,1,1),(3,-1,1),(0,-1,0),(1,-2,0),(2,-3,0),(-1,-2,0),(-2,-3,0)\}$

   (i) Is S linearly separable? If so, provide a weight vector (w_1, w_2) and a threshold t such that for each $d = (x_1, x_2, y) \in S$, $y = 1$ if and only if $x_1w_1 + x_2w_2 \geq 0$ and $y = 0$ if and only if $x_1w_1 + x_2w_2 < 0$. Else, explain why the data set is not linearly separable.

   Yes, S is linearly separable with the linear separator $x + 2y = 0$.

   (ii) What is the predicted class label by k-NN (k-nearest neighbor) of the test instance $(0.5, -0.5)$ with $k = 1$?

   Class label is 0.

   (iii) What is the predicted class label by k-NN of the test instance $(0.5, -0.5)$ with $k = 3$?

   Class label is 0.

   (iv) Exhibit a test input, if it exists, for which k-NN outputs different labels when $k = 1$ and $k = 3$.

   One of them is $(3, -3)$.

   (v) Consider a decision tree to train S in which the nodes will have tests of the form 'Is $x \leq c$?' or 'Is $y \leq c$?' for different choices of c. Compute the information gain associated with the test Is $x < 0.5$? (Compute the entropy E before the test, and the entropies E_1 and E_2 associated with the left and right subtrees after the test, compute the weighted sum E_3 of E_1 and E_2. The information gain is $E_1 - E_3$.)

   Entropy of S is 1. The two subtrees generated are of size 4 and 6. The left subtree has 1 of class 0, and 3 of class 1; The right subtree has 4 of class 0 and 2 of class 1.

   The entropy of the left subtree is 0.8112781244591328 and that of the right subtree is 0.9182624971143585. Thus the information gain is $1 - 0.8754687480522683 = 0.12453125194773174$

   (vi) Do the same as in (v) for the query 'Is $y < 0.5$?'

   Information gain $= 1 - \frac{5.0}{7} \times (\frac{-5.0}{7} \times log_2(\frac{5.0}{7}) - \frac{2.0}{7} \times log_2(\frac{2.0}{7})) = 0.38348530816669213$

4. A fair six sided dice is rolled repeatedly and you observe outcomes sequentially. Formally, dice roll outcomes are independently and uniformly drawn from the set $\{1, 2, 3, 4, 5, 6\}$. After first or second roll, you can choose between two actions: (a) Stop: stop and receive a reward equal to the dollar amount of the number you got in the last roll, (b) Roll: roll again and receive no

immediate reward. If you did not stop after roll 1 or 2, you are forced to take the action Stop after roll 3, and you receive the dollar amount of the third role as your reward and the game ends.

(a) Consider the following strategy: Role the die once. If the outcome is 5 or more, stop, else role again. If the second outcome is 4 or more, stop. Else role a third time. What is the expected reward using this strategy?

Expected reward $= \frac{1}{3} \times 5.5 + (\frac{1}{2} \times 5 + \frac{1}{2} \times 3.5) = 4.67$

(b) If we decided to stop on getting 6 on the first role, and stopping on 5 or more on the second role, what is the expected reward?

Expected reward $= \frac{1}{6} \times 6 + (\frac{2}{3} \times 3.5 + \frac{1}{3} \times 5.5) = 4.47$

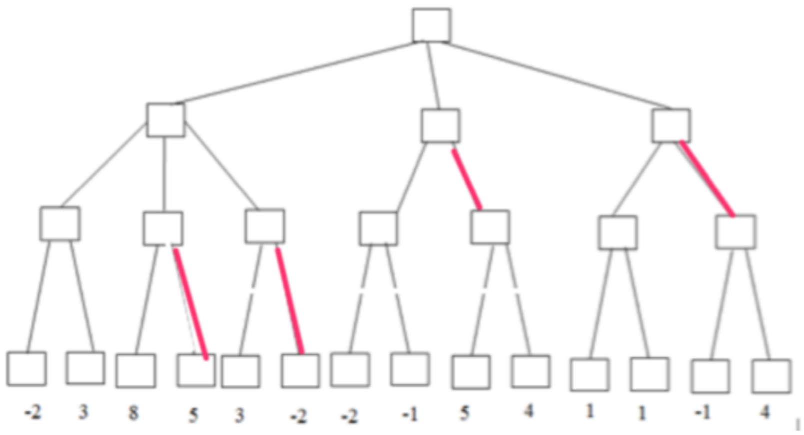
(c) Consider the variation of the above game in which we have to pay one dollar before each role. The rest of the rules are exactly the same as above. Now determine the expected reward of the strategies in (a) and (b) and compare them.

Using strategy of (a), expected reward $= \frac{1}{3} \times (5.5 - 1) + (\frac{1}{2} \times (5 - 2) + \frac{1}{2} \times (3.5 - 3) = 2.67$

Expected reward $= \frac{1}{6} \times (6 - 1) + (\frac{2}{3} \times (3.5 - 3) + \frac{1}{3} \times (5.5 - 3) = 2.083$

5. Suppose the alpha-beta pruning algorithm is applied to the game-tree shown below. As usual, the game involves players taking turns. The static evaluation given at the leaf nodes are from player 1's point of view (who, therefore, makes the move at the root of the tree). As usual, we will assume that both players have full knowledge of the tree and play optimally.

(a) Display the alpha and beta values at each node, and mark the branches that are pruned. (When a branch is pruned the entire subtree below that branch is known to be pruned, so you need not mark any other branch below a pruned branch.) One correction: Right child of the second node (from left) at level above the leaf nodes should also be marked as pruned.



The alpha and beta values at the nodes are shown below (top-to-bottom, and left-to-right at all the internal nodes):

$(3, \infty)$, $(-\infty, 3)$, $(3, 3)$, $(3, 3)$, $(3, \infty)$, $(8, 3)$, $(3, 3)$, $(3, infty)$, pruned, $(3, \infty)$, pruned.

(b) Determine the value of the root node and the optimal move for player 1 at the root.

Optimal value = 3, Optimal move = 1

(c) Suppose Player 2 makes a random choice among moves available to him/her and Player 1 knows this. Now, what is the optimal strategy for Player 1 ? What is the maximum (expected) reward for Player 1 in this case? Will alpha-beta pruning algorithm result in any pruned branches? If so, mark all the pruned branches.

Root value = 14/3. No pruning occurs.

6. The Naive Bayes model has been famously used for classifying *spam*. Each email has binary label Y which takes values in $\{spam,\ ham\}$ based on the probability of occurrence of some key words.

(a) A list of key words is chosen as feature set, and for each key word w, Prob$(w \mid Y)$ has been calculated by the table shown below:

W	note	to	self	become	perfect
$P(W \mid Y = \text{spam})$	1/6	1/8	1/4	1/4	1/8
$P(W \mid Y = \text{ham})$	1/8	1/3	1/4	1/12	1/12

You are given a new email to classify, with only two words 'perfect note'. What should be the smallest value for the prior probability for P$(Y = spam)$ that would allow this new mail to be classified as *spam*?

$P(Y = spam|w_1 = perfect,\ w2 = note) > P(Y = ham|w_1 = perfect,\ w_2 = note) \times P(w_1 = perfect|Y = s) \times P(w_2 = note|Y = s) \times P(Y = s) > P(w_1 = perfect|Y = h) \times P(w_2 = note|Y = h) \times P(Y = h)$

$1/8 \times 1/6 \times P(Y = spam) > 1/12 \times 1/8 \times (1?P(Y = spam)) \times 2/96 \times P(Y = spam) > 1/96?1/96 \times P(Y = spam) \times 3/96?P(Y = spam) > 1/96$

$P(Y = spam) > 1/3$ so the smallest value of $P(Y = spam) = 1/3$

(b) You are given only three emails as a training set:

(Spam) dear sir, I write to you in hope of recovering my gold watch.

(Ham) hey, lunch at 12?

(Ham) fine, watch it tomorrow night.

Find the values of P$(W = sir \mid Y = spam)$, P$(W = watch \mid Y = ham)$, P$(W = gauntlet \mid Y = ham)$ and P$(Y = ham)$

(c) You are training with the same emails as in the previous question, but now doing Laplace Smoothing. There are 50 words in the dictionary. Recalculate the values of P$(W = sir \mid Y = spam)$, P$(W = watch \mid Y = ham)$, P$(W = gauntlet \mid Y = ham)$ and P$(Y = ham)$.

$$P(W = \text{sir} \mid Y = \text{spam}) = \frac{c_w(W = \text{sir}, Y = \text{spam})/c_w(\text{total})}{c_w(Y = \text{spam})/c_w(\text{total})}$$
$$= \frac{c_w(W = \text{sir}, Y = \text{spam})}{c_w(Y = \text{spam}))} = \frac{1}{13}$$

Similarly, $P(W = \text{watch} \mid Y = \text{ham}) = \frac{1}{9}$.

The word "gauntlet" does not occur in our training emails, and since we're not smoothing, we estimate its conditional probability to be 0.

Estimating the prior probability $P(Y = \text{ham})$ only requires counting emails: $\frac{c_e(Y=\text{ham})}{c_e(\text{total})} = \frac{2}{3}$.

If V is the number of words in the dictionary then:

$P(W = \textbf{sir} \mid Y = \textbf{spam})$	"Sir" occurs once in the spam emails, and there are 13 total words in all spam emails. Smoothing with $k = 2$ gives $\frac{1+2}{13+2V}$.
$P(W = \textbf{watch} \mid Y = \textbf{ham})$	"Watch" occurs once in the ham emails, and there are nine total words in all ham emails. Smoothing with $k = 2$ gives $\frac{1+2}{9+2V}$.
$P(Y = \textbf{ham})$	Same as before: number of ham emails divided by the number of spam emails, which is 2/3. If you did Laplace smoothing on the priors as well, this would be 4/7, which was also accepted.

with $V = 50$, and $k = 2$, the answers are $3/113$, $3/109$ and $4/7$, respectively.

with $V = 50$, and $k = 1$, the answers are $2/63$, $2/59$ and $4/7$, respectively.

7. Consider the function $F(x, y) = \cos(x) + \cos(y) + \cos(2\pi - x - y)$. The goal of this problem is to find the maximum value of $F(x, y)$ over the region $0 \le x, y \le \pi$.

(a) Apply gradient ascent, starting with $P_0 = (x_0, y_0) = (0, 0)$ to find two successive improvements $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ using $\eta = 0.15$. Evaluate F at all three points and verify that F is increasing as we move from P_0 to P_1 to P_2.

$x_1 = y_1 = 0.1148257$ and $x_2 = y_2 = 0.13178585$. It is easy to check that $F_0 = 1.009942279905169$, $F_1 = 1.0130843518615023$ and $F_2 = 1.0171927032815233$. Clearly, $F_0 < F_1 < F_2$.

(b) Using the condition that $\frac{\partial F}{\partial x} = \frac{\partial F}{\partial y} = 0$ at the optimum value, find the maximum value (x^*, y^*) by solving the two equations (above) and find the value of F at this point. (When solving the equations, note that if $\sin(x) = \sin(y)$, you can conclude that $x = y$.)

It is easy to check that the optimum value $x^* = y^* = \pi/3$. Opt value of the function is $F^* = 1.5$

(c) To confirm that the solution you computed in (b) is, in fact, maximum (not minimum), the condition shown below should hold (at (x^*, y^*)). Check this condition: $(\frac{\partial^2 F}{\partial x^2})(\frac{\partial^2 F}{\partial y^2}) - (\frac{\partial^2 F}{\partial x \partial y})^2 < 0$

It is easy to check that the value of $(\frac{\partial^2 F}{\partial x^2})(\frac{\partial^2 F}{\partial y^2}) - (\frac{\partial^2 F}{\partial x \partial y})^2$ at (x^*, y^*) is $-1/4$.