HW10: written

Tuesday, February 5, 2019 8:07 PM

CS_188_Fall_2018_Written_HW10

CS 188 Fall 2018 | Introduction to Artificial Intelligence | **Written HW 10**

Due: Tuesday 11/13/2018 at 11:59pm (submit via Gradescope).

Leave self assessment boxes blank for this due date.

Self assessment due: Monday 11/26/2018 at 11:59pm (submit via Gradescope)

For the self assessment, **fill in the self assessment boxes in your original submission** (you can download a PDF copy of your submission from Gradescope – be sure to delete any extra title pages that Gradescope attaches). For each subpart where your original answer was correct, write "correct." Otherwise, write and explain the correct answer. **Do not leave any boxes empty.**

If you did not submit the homework (or skipped some questions) but wish to receive credit for the self-assessment, we ask that you first complete the homework without looking at the solutions, and then perform the self assessment afterwards.

Policy: Can be solved in groups (acknowledge collaborators) but must be written up individually

Submission: Your submission should be a PDF that matches this template. Each page of the PDF should align with the corresponding page of the template (page 1 has name/collaborators, question 1 begins on page 2, etc.). **Do not reorder, split, combine, or add extra pages.** The intention is that you print out the template, write on the page in pen/pencil, and then scan or take pictures of the pages to make your submission. You may also fill out this template digitally (e.g. using a tablet.)

First name	
Last name	
SID	
Collaborators	

Q1. The OMNIBUS

(a) Search

(i) [*true* or *false*] Uniform-cost search will never expand more nodes than A*-search. shitty, but valid heuristic

(ii) [*true* or *false*] Depth-first search will always expand more nodes than breadth-first search. lucky solution on first "branch" that w

(iii) [*true* or *false*] The heuristic $h(n) = 0$ is admissible for every search problem. optimism is a REQUIREMENT: h(n) <= delta(n, g)

(iv) [*true* or *false*] The heuristic $h(n) = 1$ is admissible for every search problem. false! where s has a 0 weight edge to n, and then

(v) [*true* or *false*] The heuristic $h(n) = c(n)$, where $c(n)$ is the true cheapest cost to get from the node n to a goal state, is admissible for every search problem. that's exactly what

Self assessment If correct, write "**correct**" in the box. Otherwise, **write and explain** the correct answer.

(b) CSPs

(i) [*true* or *false*] The most-constrained variable heuristic provides a way to select the next variable to assign in a backtracking search for solving a CSP. usually a secondary he

(ii) [*true* or *false*] By using the most-constrained variable heuristic and the least-constraining value heuristic we can solve every CSP in time linear in the number of variables. heuristics used aren't

Self assessment If correct, write "**correct**" in the box. Otherwise, **write and explain** the correct answer.

(c) Games

(i) [*true* or *false*] When using alpha-beta pruning, it is possible to get an incorrect value at the root node by choosing a bad ordering when expanding children. alpha beta pruning is s

(ii) [*true* or *false*] When using alpha-beta pruning, the computational savings are independent of the order in which children are expanded. we have mathematica False. Finding the supe

(iii) [*true* or *false*] When using expectimax to compute a policy, re-scaling the values of all the leaf nodes by multiplying them all with 10 can result in a different policy being optimal. expectimax decision making (priority) is unaffectec

Self assessment If correct, write "**correct**" in the box. Otherwise, **write and explain** the correct answer.

(d) MDPs For this question, assume that the MDP has a finite number of states.

(i) [*true* or *false*] For an MDP (S, A, T, γ, R) if we only change the reward function R the optimal policy is guaranteed to remain the same. maximizing cumulative reward is the whole point, this will DEFINTIELY change

(ii) [*true* or *false*] Value iteration is guaranteed to converge if the discount factor (γ) satisfies $0 < \gamma < 1$. gut

(iii) [*true* or *false*] Policies found by value iteration are superior to policies found by policy iteration.

Self assessment If correct, write "**correct**" in the box. Otherwise, **write and explain** the correct answer.
both converge to optimal policy

(e) Reinforcement Learning

(i) [*true* or *false*] Q-learning can learn the optimal Q-function Q^* without ever executing the optimal policy. but WHY?!

(ii) [*true* or *false*] If an MDP has a transition model T that assigns non-zero probability for all triples $T(s, a, s')$ then Q-learning will fail. not sure why it would, this is a perfectly valid MDP

Self assessment If correct, write "**correct**" in the box. Otherwise, **write and explain** the correct answer.

ve explore in depth

a 0 weight edge to g. if it had a 0.5 direct edge to g, we would explore it first and terminate
: we want!

euristic? I believe we like MRV better

stronge nough to guarantee no back-tracking (?)

safe-pruning.
l guarantees that pruned parts of tree will "not be chosen" by one of the two agents
er awesome leaves early lets you prune more agressively :)

d, linear scaling is monotonic.

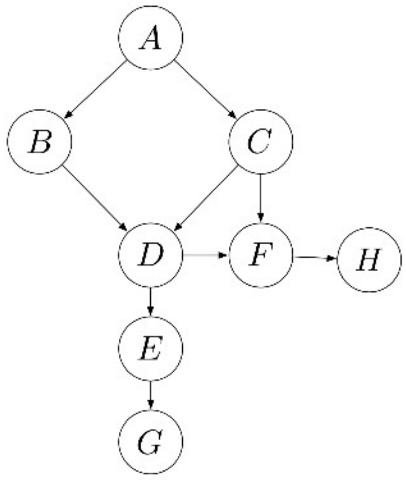
(e) Reinforcement Learning

(i) [*true* or *false*] Q-learning can learn the optimal Q-function Q^* without ever executing the optimal policy. but WHY?!

(ii) [*true* or *false*] If an MDP has a transition model T that assigns non-zero probability for all triples $T(s, a, s')$ then Q-learning will fail. not sure why it would, this is a perfectly valid MDP

Self assessment If correct, write "**correct**" in the box. Otherwise, **write and explain** the correct answer.

(f) Bayes' Nets For each of the conditional independence assertions given below, circle whether they are guaranteed to be true, guaranteed to be false, or cannot be determined for the given Bayes' net.



$B \perp\!\!\!\perp C$	Guaranteed true	Guaranteed false	Cannot be determined	shared common cause, variable
$B \perp\!\!\!\perp C \mid G$	Guaranteed true	Guaranteed false	Cannot be determined	shared cause
$B \perp\!\!\!\perp C \mid H$	Guaranteed true	Guaranteed false	Cannot be determined	shared cause
$A \perp\!\!\!\perp D \mid G$	Guaranteed true	Guaranteed false	Cannot be determined	chain of causality
$A \perp\!\!\!\perp D \mid H$	Guaranteed true	Guaranteed false	Cannot be determined	chain of causality
$B \perp\!\!\!\perp C \mid A, F$	Guaranteed true	Guaranteed false	Cannot be determined	d-sep moralizes on variable D, s
$F \perp\!\!\!\perp B \mid D, A$	Guaranteed true	Guaranteed false	Cannot be determined	d-sep
$F \perp\!\!\!\perp B \mid D, C$	Guaranteed true	Guaranteed false	Cannot be determined	

Self assessment If correct, write "**correct**" in the box. Otherwise, **write and explain** the correct answer.

e A. d-sep moralizes on A

shares common evidence when given F

Q2. Perceptron

(a) Suppose you have a binary perceptron in 2D with weight vector $\mathbf{w} = r\,[w_1, w_2]^T$. You are given w_1 and w_2, and are given that $r > 0$, but otherwise not told what r is. *Assume that ties are broken as positive.*
Can you determine the perceptron's classification of a new example x with known feature vector $f(x)$?

◯ Always ◯ Sometimes ◯ Never bias term needed if this includes the bias term, then we koo. hyp

(b) Now you are learning a multi-class perceptron between 4 classes. The weight vectors are currently $[1, 0]^T$, $[0, 1]^T$, $[-1, 0]^T$, $[0, -1]^T$ for the classes A, B, C, and D. The next training example x has a **label of A** and feature vector $f(x)$.
For the following questions, *do not make any assumptions about tie-breaking.* (Do not write down a solution that creates a tie.)

(i) Write down a feature vector in which no weight vectors will be updated.

$$f(x) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

◯ Not possible

(ii) Write down a feature vector in which **only** $\mathbf{w}_A$ will be updated by the perceptron.

$$f(x) = \begin{bmatrix} \quad \end{bmatrix}$$

◯ Not possible

(iii) Write down a feature vector in which **only** $\mathbf{w}_A$ and $\mathbf{w}_B$ will be updated by the perceptron.

$$f(x) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

◯ Not possible

(iv) Write down a feature vector in which **only** $\mathbf{w}_A$ and $\mathbf{w}_C$ will be updated by the perceptron.

$$f(x) = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

◯ Not possible

The weight vectors are the same as before, but now there is a bias feature with value of 1 for all x and the weight of this bias feature is $0, -2, 1, -1$ for classes A, B, C, and D respectively. As before, the next training example x has a **label of A** and a feature vector $f(x)$. The always "1" bias feature is the first entry in $f(x)$.

(v) Write down a feature vector in which **only** $\mathbf{w}_B$ and $\mathbf{w}_C$ will be updated by the perceptron.

$$f(x) = \begin{bmatrix} 1 \\ \\ \end{bmatrix}$$

◯ Not possible

(vi) Write down a feature vector in which **only** $\mathbf{w}_A$ and $\mathbf{w}_C$ will be updated by the perceptron.

$$f(x) = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$$

◯ Not possible

Self assessment If correct, write "**correct**" in the box. Otherwise, **write and explain** the correct answer.

erplane is scalable. if w1 is not corresponding to bias term, then no u foooked