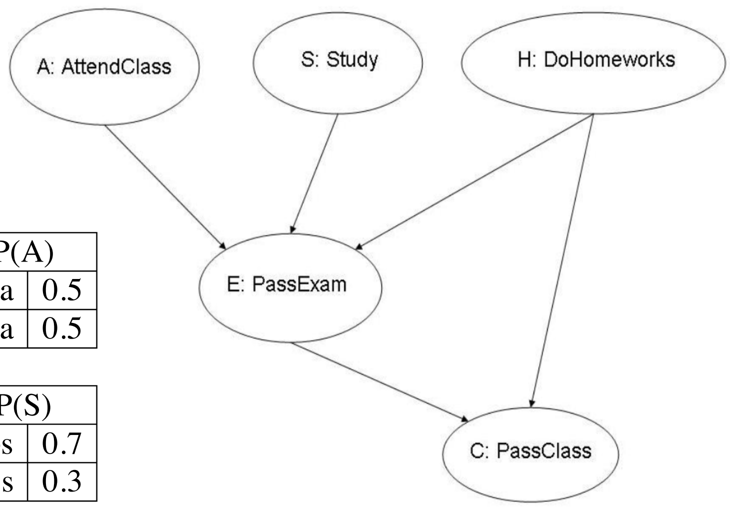
<u>3. [15%] Bayesian Network.</u>

Suppose that a student PassExam(E), could be caused by AttendClass(A), Study(S), DoHomeworks(H). PassClass(C) could be caused by PassExam(E) and DoHomeworks(H). As long as you have the formula correct, you will get full credit. There is no need to calculate the final numbers.



P(A)	
+a	0.5
-a	0.5

P(S)	
+s	0.7
-s	0.3

P(H)	
+h	0.9
-h	0.1

P(C|E,H)			
-e	-h	+c	0.1
-e	-h	-c	0.9
-e	+h	+c	0.4
-e	+h	-c	0.6
+e	-h	+c	0.3
+e	-h	-c	0.7
+e	+h	+c	0.9
-e	+h	-c	0.1

P(E|A,S,H)				
-a	-s	-h	+e	0.2
-a	-s	-h	-e	0.8
-a	-s	+h	+e	0.5
-a	-s	+h	-e	0.5
-a	+s	-h	+e	0.4
-a	+s	-h	-e	0.6
-a	+s	+h	+e	0.8
-a	+s	+h	-e	0.2
+a	-s	-h	+e	0.3
+a	-s	-h	-e	0.7
+a	-s	+h	+e	0.7
+a	-s	+h	-e	0.3
+a	+s	-h	+e	0.6
+a	+s	-h	-e	0.4
+a	+s	+h	+e	0.9
+a	+s	+h	-e	0.1

3A. [3%] Compute the following entry from joint distribution: P(+a, +s, +h, +e, +c).

$$P(+a, +s, +h, +e, +c)$$

$$= P(+a) \cdot P(+s) \cdot P(+h) \cdot P(+e \mid +a, +s, +h) \cdot P(+c \mid +e, +h)$$

+C

3B. [3%] What is the probability of passing the class, given that you attended class and studied, but didn't do the homework?

+a

+S -h

$$P(+c \mid +a, +s, -h) = \frac{P(+c, +a, +s, -h)}{P(+a, +s, -h)}$$

$$= \frac{\sum_e P(+c, +a, +s, e, -h)}{\sum_{c,e} P(c, +a, +s, e, -h)}$$

$$= \frac{\sum_e P(+a) \cdot P(+s) \cdot P(-h) \cdot P(e \mid +a, +s, -h) \cdot P(+c \mid e, -h)}{\sum_{e,c} P(+a) \cdot P(+s) \cdot P(-h) \cdot P(e \mid +a, +s, -h) \cdot P(c \mid e, -h)}$$

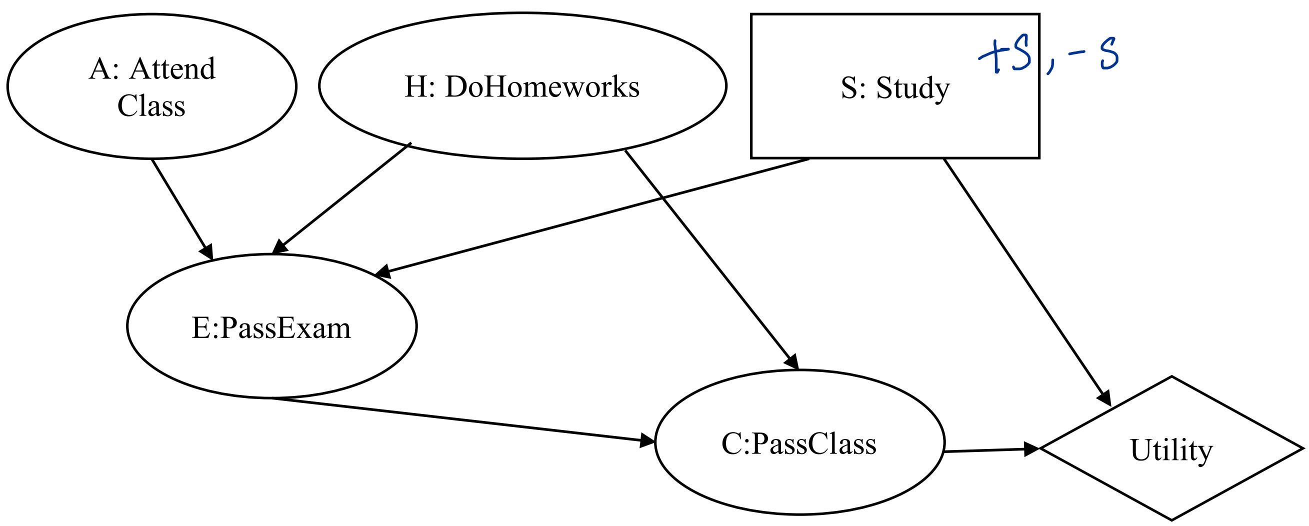
3C. [3%] Compute P(+a | +c, +h).

$$P(+a \mid +c, +h) = \frac{P(+a, +c, +h)}{P(+c, +h)}$$

$$= \frac{\sum_{e,s} P(+a, +c, e, s, +h)}{\sum_{a,e,s} P(a, +c, e, s, +h)}$$

$$= \frac{\sum_{e,s} P(+a) \cdot P(s) \cdot P(+h) \cdot P(e \mid +a, s, +h) \cdot P(+c \mid e, +h)}{\sum_{a,e,s} P(a) \cdot P(s) \cdot P(+h) \cdot P(e \mid a, s, +h) \cdot P(+c \mid e, +h)}$$

3D. [6%] Now, consider a student who has the choice to study (Study(S)) or not study for passing the exam. We will model this as a decision problem with one Boolean decision node, S, indicating whether the agent chooses to study. The remaining nodes are chance nodes. The probabilities are the same as above. There is also a utility node U. We have the following utility function: the utility for PassClass is 100 and the utility for Not PassClass is -10. The utility for Study is -50 and the utility for Not Study is 0.
Calculate the MEU for decision node S.



$P(+c|+s) = P(+c,+s) = \sum_{a,h,e} P(+c,+s,a,h,e)$

$= \sum_{a,h,e} P(a)\cdot P(+s)\cdot P(h)\cdot P(e|a,+s,h)\cdot P(+c|e,h)$

$P(-c|+s) = ----$

$P(+c|-s) = ----$

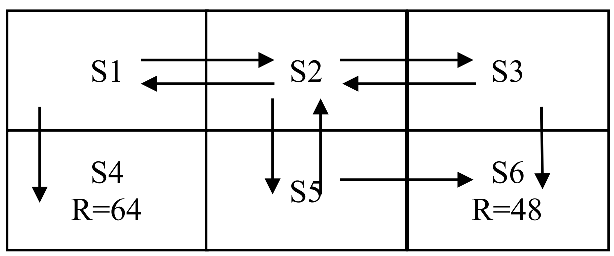
$P(-c|-s) = ----$

Expected Utility when study: $P(+c|+s)\cdot(100-50) + P(-c|+s)\times(-60)$

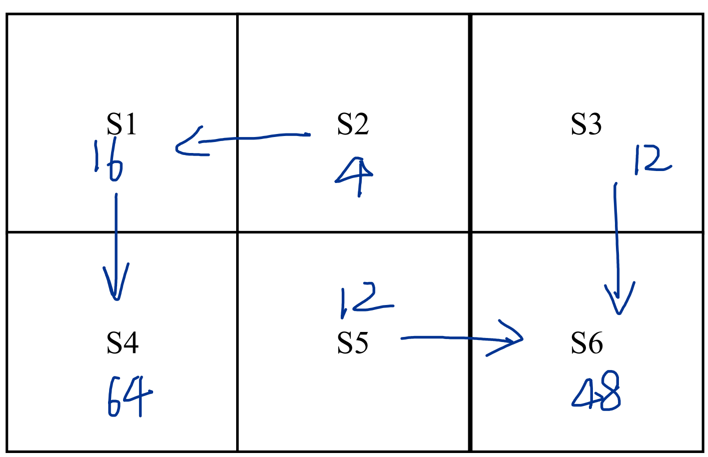
not study: $P(-c|-s)\cdot(-10) + P(+c|-s)\cdot(100)$

4. [20%] Markov Decision Process and Reinforcement Learning

Consider the 6-state Markov Decision process below. The goals with rewards are in state S4 and S6. At each state, the possible transitions are deterministic and indicated by the arrows. You get a reward of R_4=64 if you get to the goal S4 and a reward of R_6=48 if you get to the goal S6. In this problem, you have to get the final answer correct to get credits.



4A. [4%] Consider a discount factor of $\gamma = \frac{1}{4}$. On the figure below, show the optimal value V* for each state and the arrows corresponding to the set of the optimal actions.



4B. [5%] What values of γ would result in a different optimal action in S3? Indicate which policy action changes.

$$64 \cdot \gamma^3 \geq 48 \cdot \gamma$$

$$\gamma^2 \geq \frac{48}{64} = \frac{3}{4}$$

$$\gamma \geq \sqrt[2]{\frac{3}{4}}$$

4C. [8%] Suppose the action is no longer deterministic. The discount factor is still $\gamma = \frac{1}{4}$. Each action has a failure probability f=0.2, i.e. the action is now stochastic. Once an action is taken, with probability 0.2, it would stay in the original grid. On the figure below, show the optimal value V* for each state and the arrows corresponding to the set of the optimal actions.

new_value = $\gamma \cdot$ next_state $\times (1-f) + f \cdot$ this_state

S1 12.8	S2 2.56	S3 9.6
S4 64	S5 9.6	S6 48

4D. [3%] In this scenario, the discount factor is $\gamma = 1$ and the learning rate $\alpha = 0.5$. S6 has the reward 10 and S4 has the reward -10. All other states have reward 0. Now we no longer know the details of the transition probabilities ahead of time. We must instead use reinforcement learning to compute the necessary values. We have the following sequence of actions.

Remember that $Q(s, a)$ is initialized with 0 for all (s, a).

$Q(s,a) \leftarrow Q(s,a) + \alpha(R(s) + \gamma \max_{a'} Q(s',a') - Q(s,a))$

Example 1

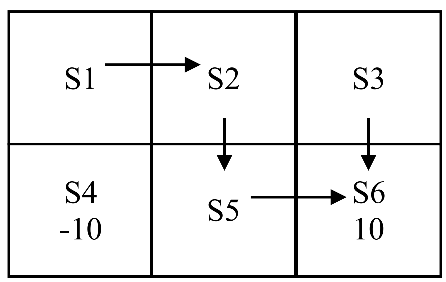
s	a	s'	r
S1	Right	S2	0
S2	Down	S5	0
S5	Right	S6	0
S6	Exit	Terminal	10

Example 2

s	a	s'	r
S1	Right	S2	0
S2	Down	S5	0
S5	Right	S6	0
S6	Exit	Terminal	10

Example 3

s	a	s'	r
S1	Right	S2	0
S2	Down	S5	0
S5	Right	S4	0
S4	Exit	Terminal	-10



After 3 example runs, what is the Q-value of (S5, Right)?

0

First, $Q(S5, Right) = 0 + 0.5 \cdot (0 + 1 \times (10 - 0)) = 5$

S_2.

3.

5. [15%] Decision Trees

Assume we want to train a machine to decide whether someone takes an Uber or just walks based on weather information and the length of the path. Our training data is provided below. The weather outlook can be either sunny (s), cloudy (c) or rainy (r). The humidity level can be either high (h) or normal (n). The destination is either near (n) or far (f).

Trip	Outlook	Humidity	Destination	TakeUber?
T1	s	h	n	y
T2	s	h	f	y
T3	c	h	n	n
T4	r	h	n	n
T5	r	n	n	n
T6	r	n	f	n
T7	c	n	f	n
T8	s	h	n	y
T9	s	n	n	n
T10	r	n	n	n
T11	s	n	f	n
T12	c	h	f	n
T13	c	n	n	n
T14	r	h	f	y
T15	r	n	n	n
T16	c	h	f	n

s: y:3, n:2
c: y:0, n:5
r: y:1, n:5

h: y:4, n:4
n: y:0, n:8

n: y:2 n:7
f: y:2 n:5

$\frac{1}{16} \times -\left[\log\left(\left(\frac{2}{9}\right)^2 \cdot \left(\frac{7}{9}\right)^7\right) + \log\left(\left(\frac{2}{7}\right)^2 \cdot \left(\frac{5}{7}\right)^5\right)\right]$

5A. [6%] Which attribute would be at the root of the tree? Give the reason for your answer in terms of entropy and information gain (no need for exact calculation of the expressions).

yes: 4 no: 12

$I(\text{takeUber}) = -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4}$

For Outlook:

$I(\text{sunny}) = -\frac{3}{5}\log\frac{3}{5} - \frac{2}{5}\log\frac{2}{5}$

$I(\text{cloudy}) = -0\log 0 - 1 \cdot \log_2 1 = 0$

$I(\text{rainy}) = -\frac{1}{6}\log\frac{1}{6} - \frac{5}{6}\log\frac{5}{6}$

$I(\text{Outlook}) = \frac{5}{16} \cdot I(\text{sunny}) + \frac{5}{16} I(\text{cloudy}) + \frac{6}{16} I(\text{rainy}) > \frac{1}{2}$

For Humidity:

$I(\text{high}) = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = 1$

$I(\text{normal}) = -0\log 0 - 1\log 1 = 0$ ⇒ least remainder

$I(\text{Humidity}) = \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 0 = \frac{1}{2}$ highest IG.

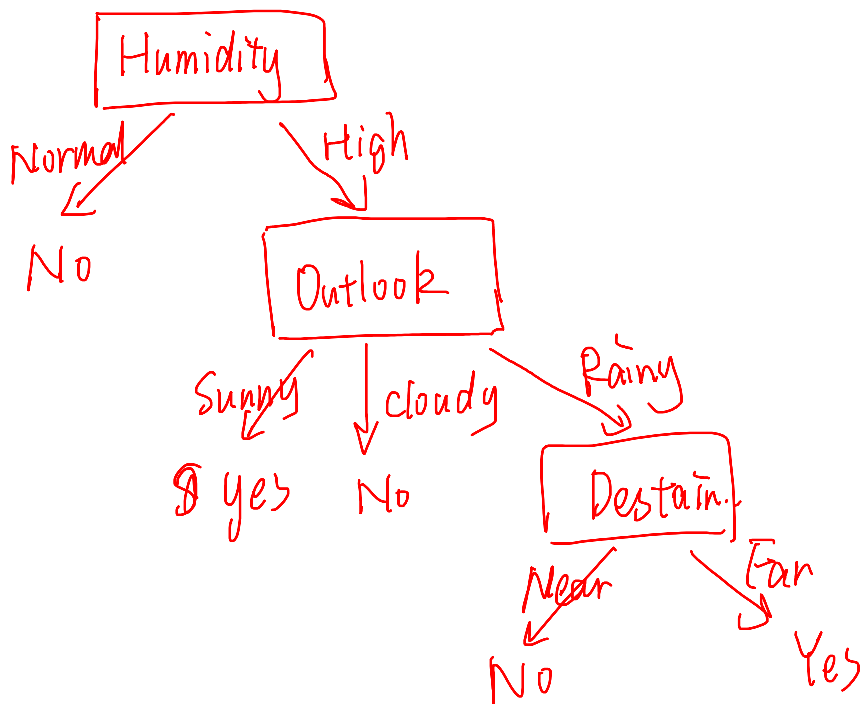
⇒ Root

For Des:

$\text{Remainder}(\text{Des}) = \frac{9}{16} \cdot \left(-\frac{2}{9}\log\frac{2}{9} - \frac{7}{9}\log\frac{7}{9}\right) + \frac{7}{16}\left(-\frac{2}{7}\log\frac{2}{7} - \frac{5}{7}\log\frac{5}{7}\right)$

$> \frac{1}{2}$

5B. [9%] Draw the learned decision tree and write down the learned Boolean expression for TakeUber as a disjunction of clauses.



TakeUber = (HumidityHigh $\wedge$ Sunny) $\vee$

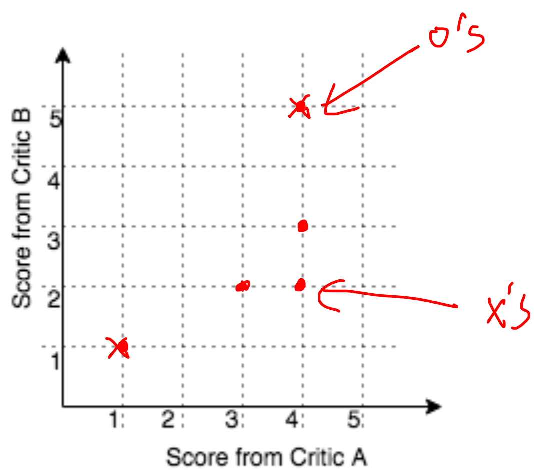
(HighHumidity $\wedge$ Rainy $\wedge$ Far).

6. [10%] Neural Nets

Assume you want to predict how well a show will do with the general audience, based on the scores of two critics who score the show on the scale of 1 to 5. Here are five data points from some previous shows, including the critics' score and the performance of the shows:

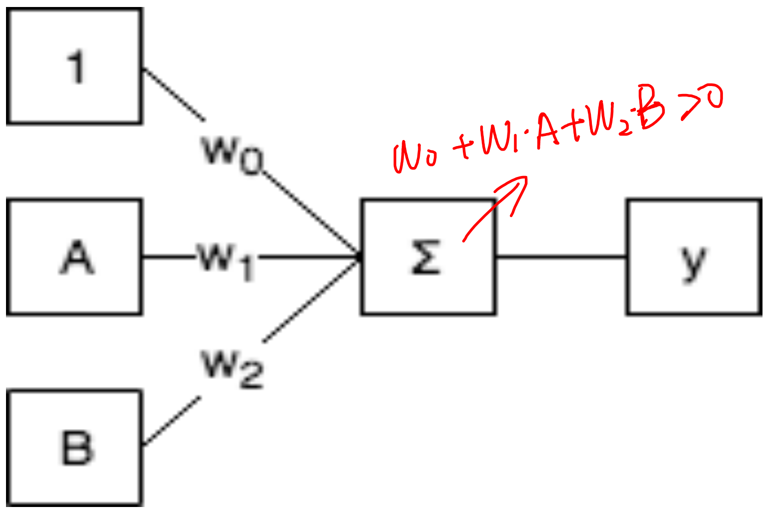
Show	Critic A Score	Critic B Score	Did the audience like the show?
1	1	1	No
2	3	2	Yes
3	4	5	No
4	4	3	Yes
5	4	2	Yes

6A. [2%] Determine if the data is linearly separable by plotting it on the 2D plane below. Represent positive data points with x's and negative data points with o's.



yes.

6B. [8%] After establishing linear separability, you decide to use a perceptron to classify the data, using the scores as features. In the figure below, A, B are inputs from each critic and constant input 1 is a bias input, y is output.
So, the perceptron would be like this (If $w_0 + w_1*A + w_2*B > 0$, the audience will like the show the output (y = 1); otherwise they won't (y = -1):



This is how perceptron update works:

Start with an initial vector of weights.
For each training instance (y is the output and y* is the expected result):
If correct (y = y*), no update is needed.
If wrong: update the weight vector by adding or subtracting the input vector. Subtract if y* is -1.

For example, for the input vector <1, A, B>, if at a training iteration the input is (1, 2, 1), the weight vector is (1, 2, 3), the output is 1, and the expected result is -1, the updated weight vector will be: (0, 0, 2).

<1, 3, 0>

Assume we start with the weights {w_0: 0, w_1: 0, w_2: 0} (weight vector (0, 0, 0)). Determine the weights after two updates. Calculate the accuracy of the perceptron on the data after these two updates.

Input <1, 3, 2>   Weight <0, 0, 0>

first: <1, 3, 2)

Input <1, 4, 5)   Weight <1, 3, 2>

second <0, -1, -3>   用这个去算5个Input的准确度

Accuracy: 40%.