CSCI 561 - Foundation for Artificial Intelligence

Discussion Section (Week 14) Sample Final Review

PROF WEI-MIN SHEN WMSHEN@USC.EDU

Probability

Your doctor performs a series of medical tests. You test positive for a serious, but very rare disease. The test is 99% accurate, meaning the probability that it is positive when you do have the disease is 0.99, and the probability that it is negative if you don’t is also 0.99. The disease strikes 1 in 10,000.

What is the probability that you have the disease if your test is positive?
Give both the formula and the value.

P(test|disease) = 0.99   P(¬test|¬disease) = 0.99

P(disease) = 0.0001

What the patient is concerned about is P(disease|test).

P(disease|test)

$$= \frac{P(test|disease)P(disease)}{P(test|disease)P(disease)+P(test|\neg disease)P(\neg disease)} \qquad \frac{= 0.99\times 0.0001 = .000099}{= 0.99\times 0.0001+0.01\times 0.9999 \\ =.010098}$$

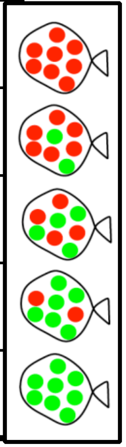
= .000099 / .010098 = .009804

Candy Example

$P(d_j|h_i)$ when d_j is a Lime

Suppose there are fives kinds of candy bags:

Hypothesis	Prior	Cherry	Lime
h_1	.1	100%	0%
h_2	.2	75%	25%
h_3	.4	50%	50%
h_4	.2	25%	75%
h_5	.1	0%	100%



$P(d_j|h_1)$

$P(d_j|h_2)$

$P(d_j|h_3)$

$P(d_j|h_4)$

$P(d_j|h_5)$

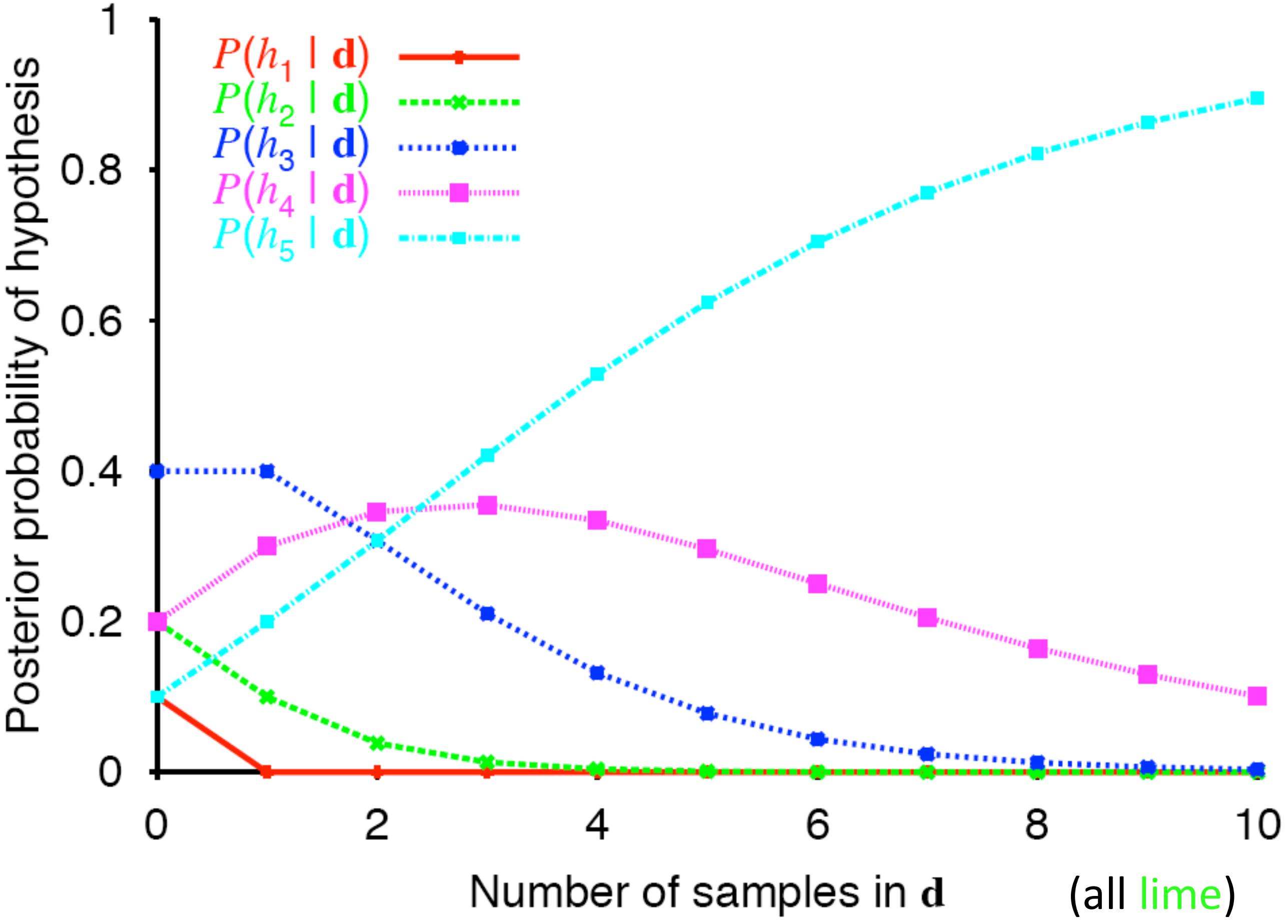
Then we observe candies: ● ● ● ● ● ● ● ● ● ●

What kind of bag is it? What flavor will be next?

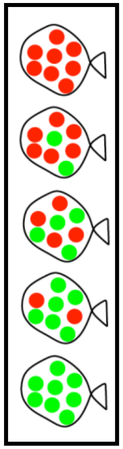
Assume big bag so that draws are i.i.d. $P(\mathbf{d} \mid h_i) = \Pi_j P(d_j \mid h_i)$

$P(\mathbf{d} \mid h_i) = P(\text{lime} \mid h_3)\, P(\text{lime} \mid h_3)\, P(\text{lime} \mid h_3)\, P(\text{lime} \mid h_3)\, P(\text{lime} \mid h_3)\, P(\text{lime} \mid h_3)\, P(\text{lime} \mid h_3)\, P(\text{lime} \mid h_3)\, P(\text{lime} \mid h_3)\, P(\text{lime} \mid h_3)$
$= .5 * .5 * .5 * .5 * .5 * .5 * .5 * .5 * .5 * .5$

Posterior Probability of H



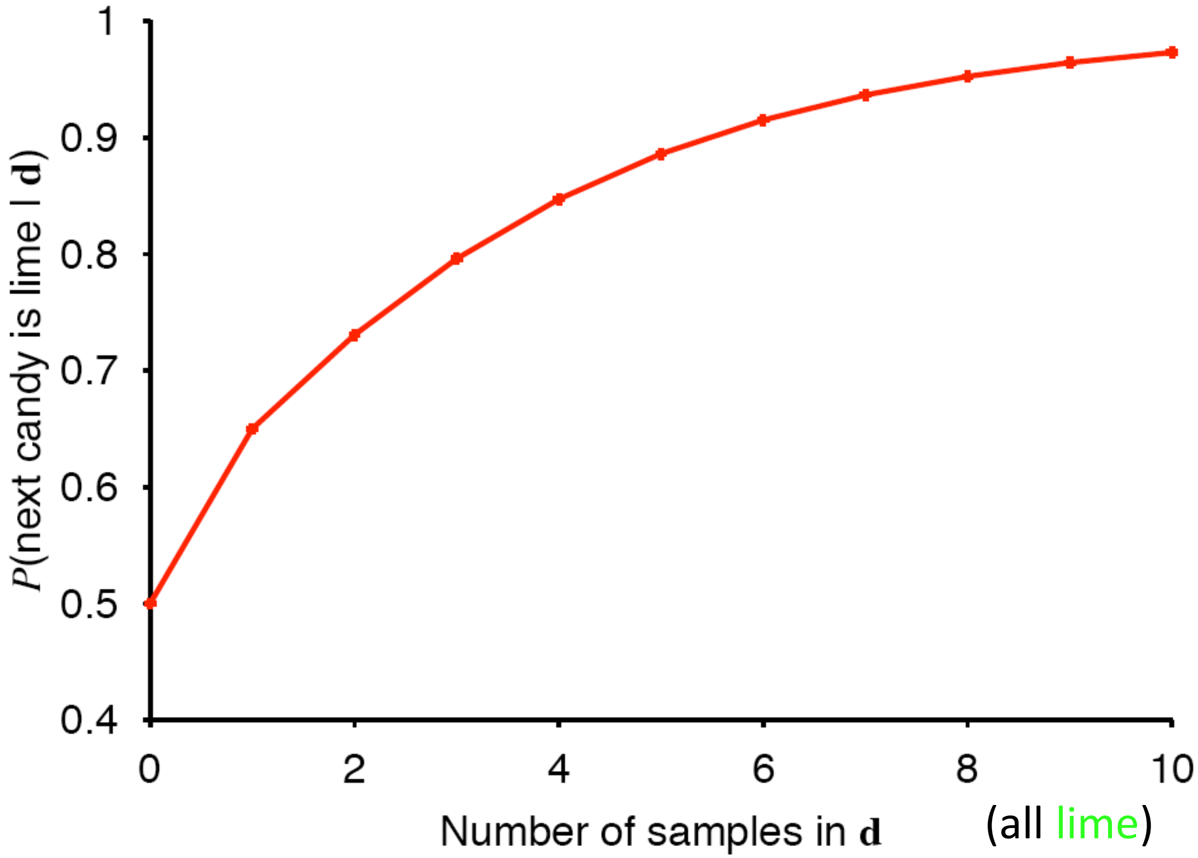
(all lime)



Computed as $P(h_i \mid d) = \alpha \Pi_j P(d_j \mid h_i) P(h_i)$

Ex: $P(h_3 \mid \mathbf{d}) = \alpha\pi_j (.5 * .4)$

Probability of Predicting Lime

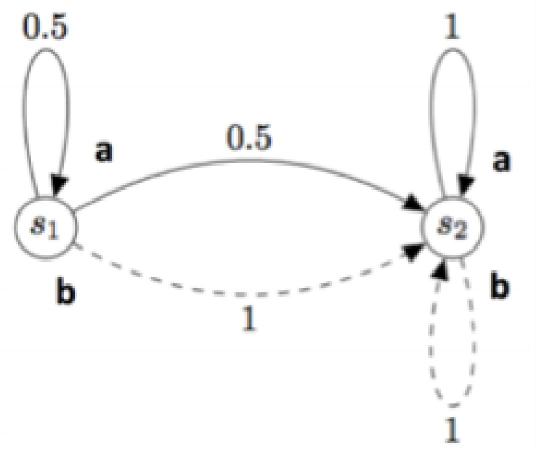


(all lime)

Computed as P(*lime* | d) = Σ_i P(*lime* | h_i)P(h_i | d)

MDP

Consider a simple MDP with two states, s1 and s2, and two actions, a (solid line) and b (dashed line); the numbers indicate transition probabilities. Rewards, which just depend on state and action (not the state resulting from the action), are shown in the table below.



$R(S_1\|a) = 8$	$R(S_2\|a) = -4$
$R(S_1\|b) = 16$	$R(S_2\|b) = -4$

Supposing that U_0 of both states is 0 and the discount factor, γ, is .5, fill in the four boxes (U_1 and U_2), Show all your work (including formulas) below.

~~Remember $U_{t+1}(s) = R(s) + \text{Max}_{a\varepsilon A} \{\gamma \Sigma_{s'\varepsilon S} P(s'|a,s) U_t(s')\}$~~

$U_0(s_1)=0$	$U_0(s_2)=0$
$U_1(s_1)=$	$U_1(s_2)=$
$U_2(s_1)=$	$U_2(s_2)=$

$U_{t+1}(s) = \text{Max}_{a\varepsilon A} \{ R(s,a) + \gamma\, \Sigma_{s'\varepsilon S} P(s'|a,s)\, Ut(s') \}$

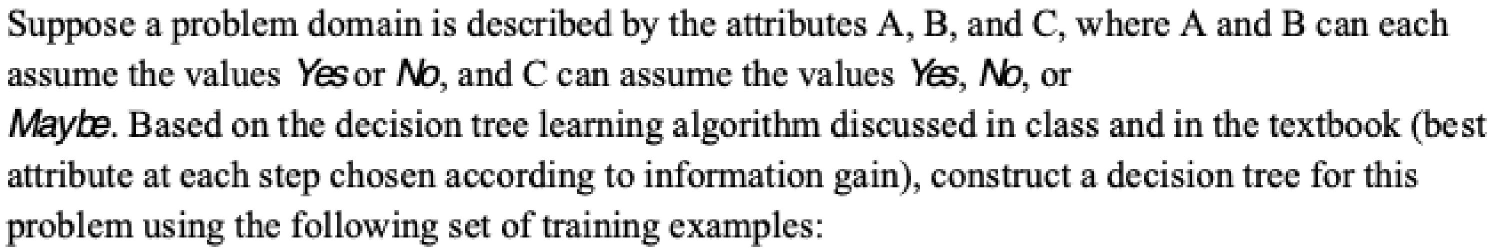
$U_1(S_1) = \text{Max} \{ (8 + 0.5*(0.5*0+0.5*0)) , (16 + 0.5*1*0) \} = \text{Max} (8,16) = 16$

$U_1(S_2) = \text{Max} ((-4 + 0.5*1*0) , (-4 + 0.5*1*0)) = \text{Max} (-4,-4) = -4$

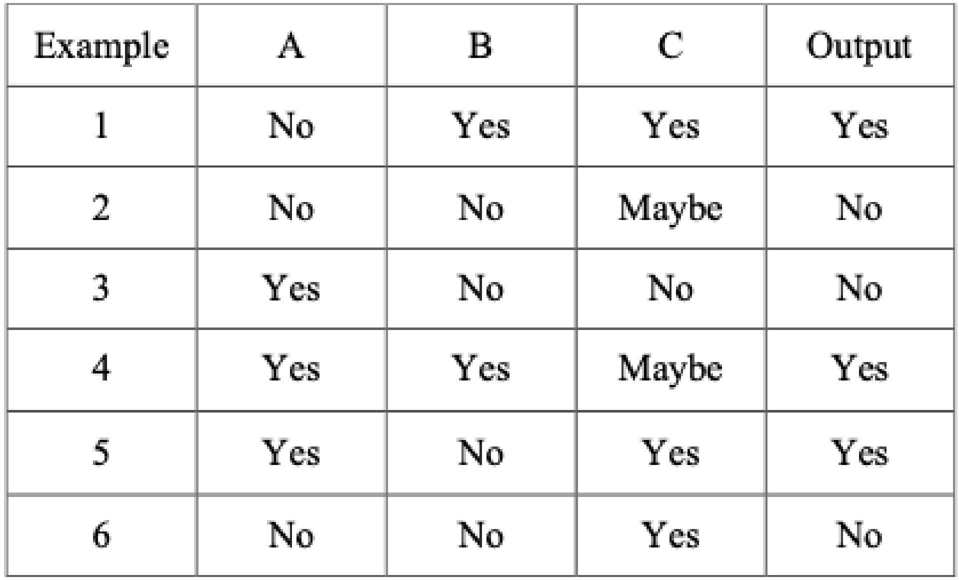
$U2(S1) = \text{Max} ((8 + 0.5*(0.5*16+0.5*-4)) , (16 + 0.5*1*-4)) = \text{Max} (11,14) = 14$

$U2(S2) = \text{Max} ((-4 + 0.5*1*-4) , (-4 + 0.5*1*-4)) = \text{Max} (-6,-6) = -6$

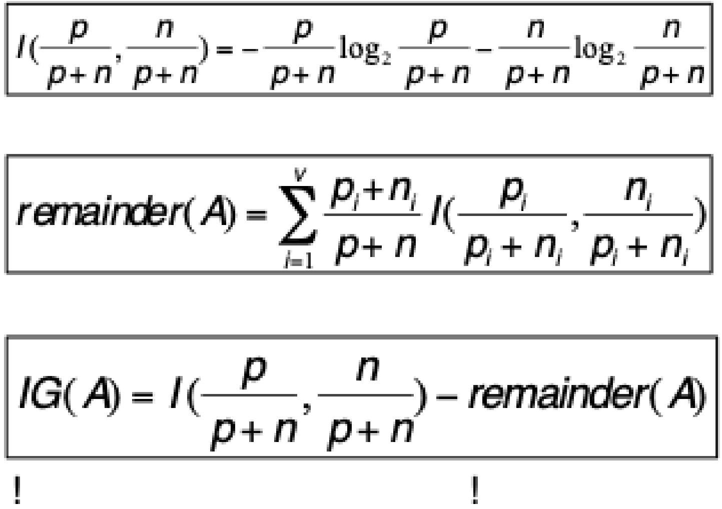
Decision Tree



Suppose a problem domain is described by the attributes A, B, and C, where A and B can each assume the values *Yes* or *No*, and C can assume the values *Yes*, *No*, or *Maybe*. Based on the decision tree learning algorithm discussed in class and in the textbook (best attribute at each step chosen according to information gain), construct a decision tree for this problem using the following set of training examples:



Example	A	B	C	Output
1	No	Yes	Yes	Yes
2	No	No	Maybe	No
3	Yes	No	No	No
4	Yes	Yes	Maybe	Yes
5	Yes	No	Yes	Yes
6	No	No	Yes	No



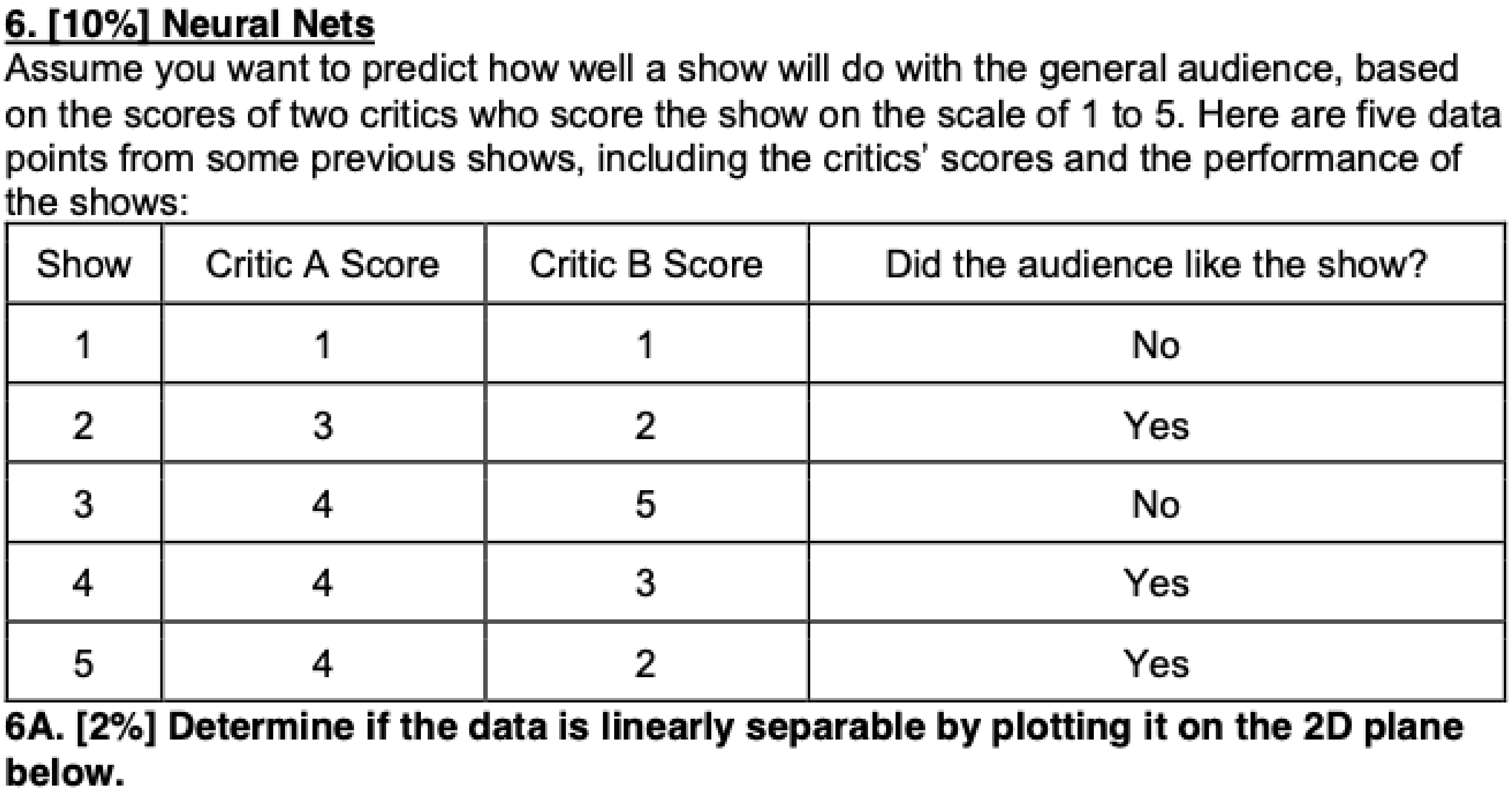
$$I(\frac{p}{p+n},\frac{n}{p+n}) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

$$remainder(A) = \sum_{i=1}^{v}\frac{p_i+n_i}{p+n} I(\frac{p_i}{p_i+n_i},\frac{n_i}{p_i+n_i})$$

$$IG(A) = I(\frac{p}{p+n},\frac{n}{p+n}) - remainder(A)$$

! !

At root: I(3/6 , 3/6) = 1
Remainder(A) = 3/6 I(2/3 ,1/3) + 3/6 I(2/3,1/3)
Remainder(B) = 2/6 I(2/2 , 0) + 4/6 I (1⁄4 , 3⁄4)
Remainder(C) = 3/6 I(2/3 , 1/3) + 1/6 I(1,0) + 2/6 I(1/2 , 1⁄2)
IG(B) is the largest so it will be the root. Similarly, next one is A , and then C.

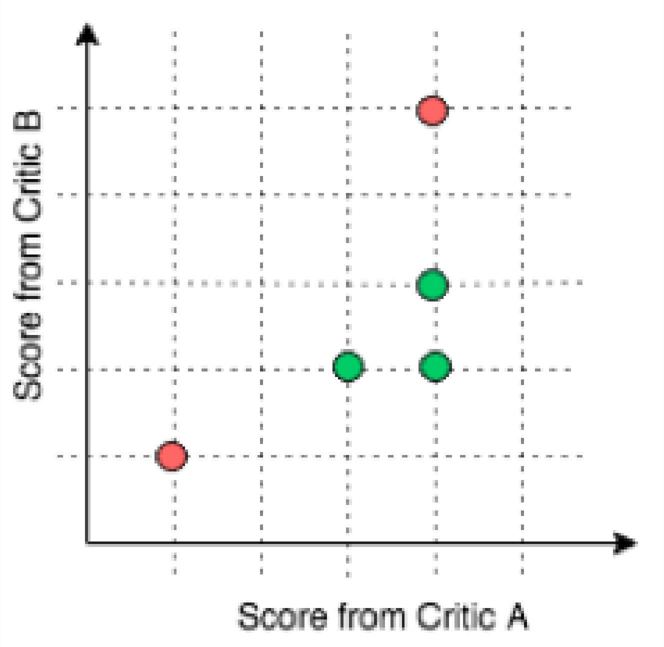


6. [10%] Neural Nets

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' scores and the performance of the shows:

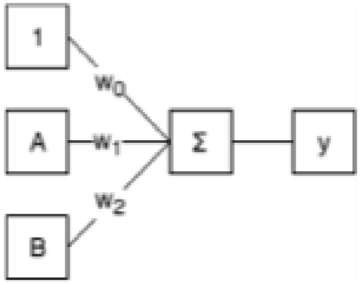
Show	Critic A Score	Critic B Score	Did the audience like the show?
1	1	1	No
2	3	2	Yes
3	4	5	No
4	4	3	Yes
5	4	2	Yes

6A. [2%] Determine if the data is linearly separable by plotting it on the 2D plane below.



[Green dots should be x's, red dots should be o's.]
The data is linearly separable.

6B. [8%] After establishing linear separability, you decide to use a perceptron to classify the data, using the scores as features. So the perceptron would be like this (If $w_0 + w_1*A + w_2*B > 0$, the audience will like the show ($y = 1$); 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.
```

So for example, 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).

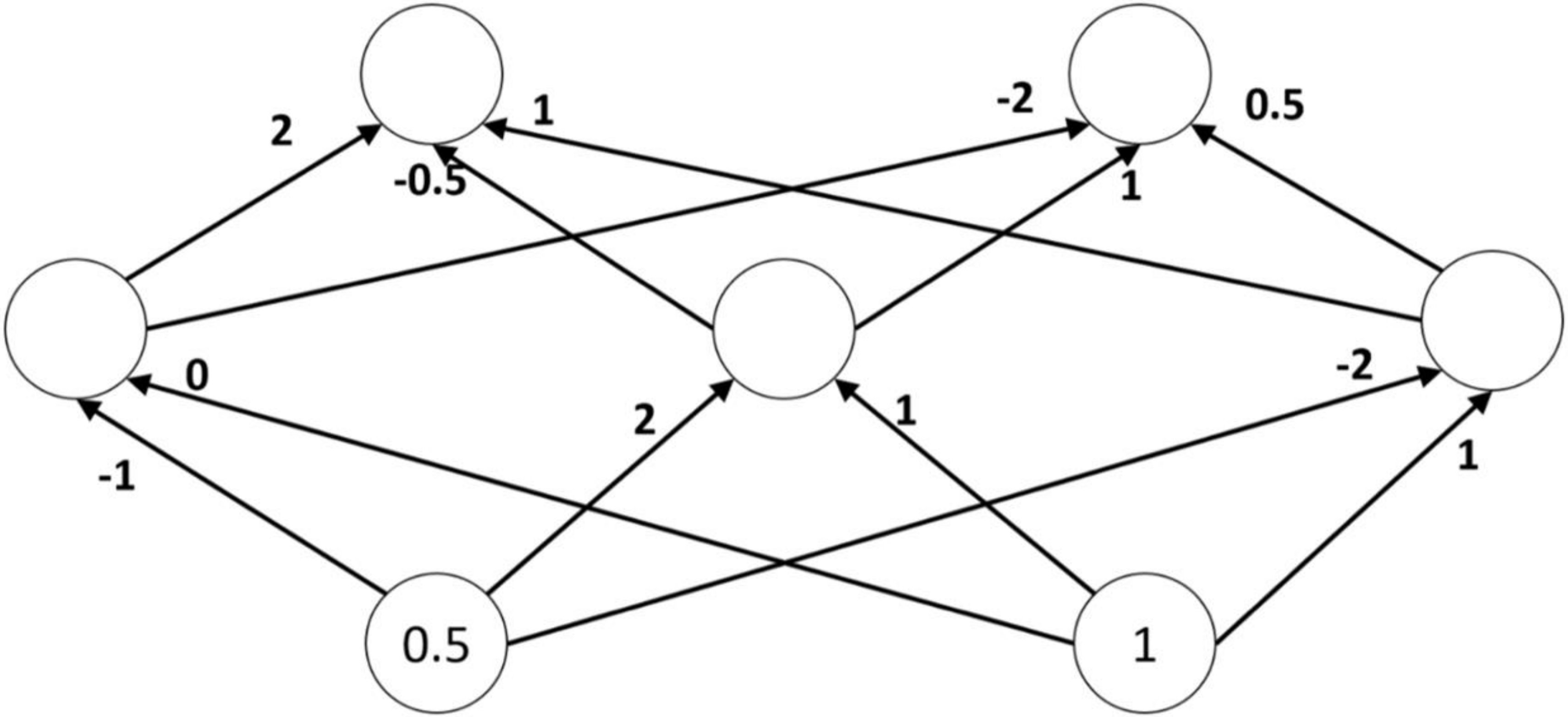
Assume we start with the weights {w_0: 0, w_1: 0, w_2: 0} (weight vector (0, 0, 0)). Determine the weights after two updates. Calculate the Accuracy of the perceptron on the data after these two updates.

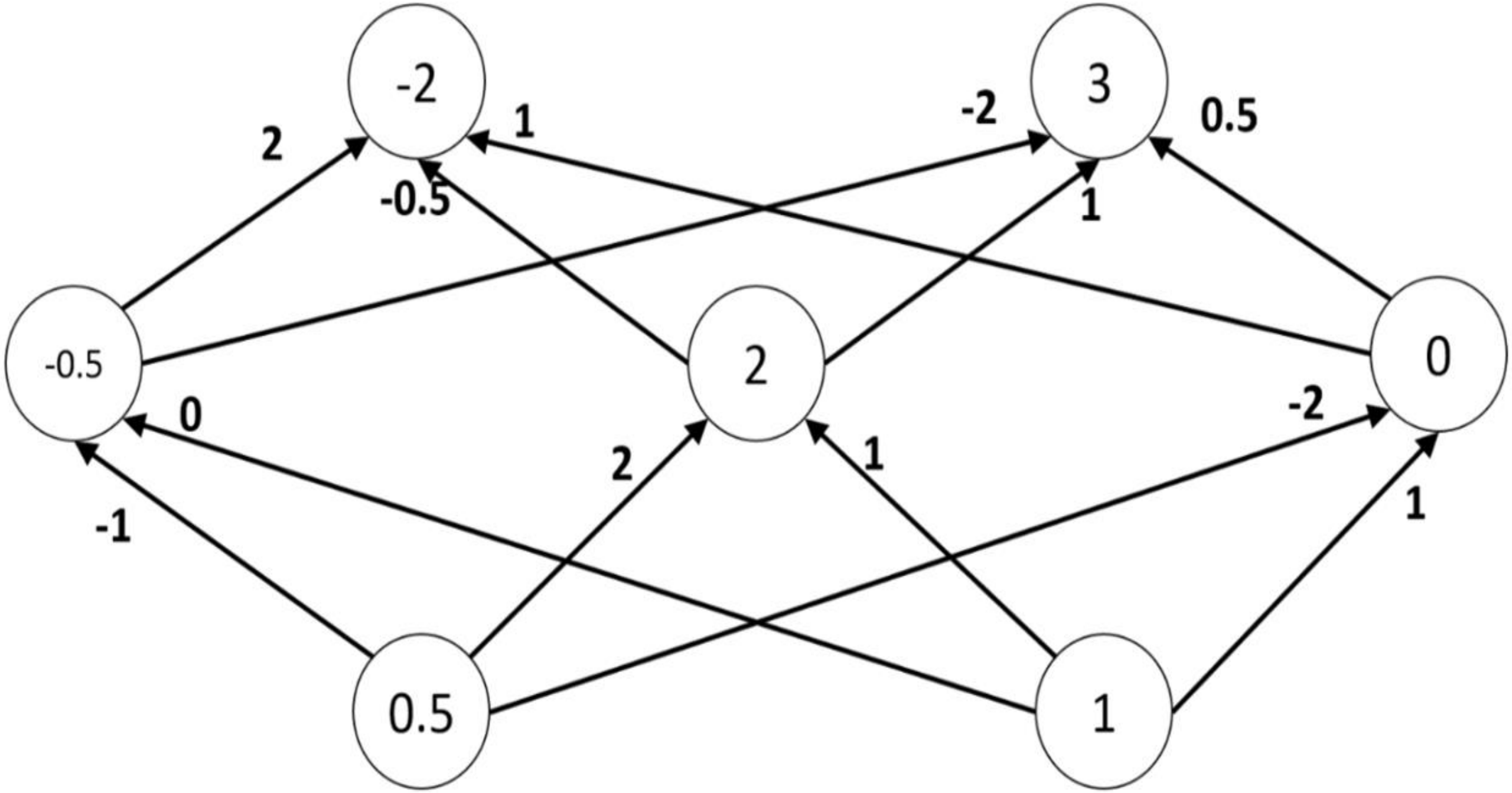
After first update: {w_0: 1, w_1: 3, w_2: 2}
After second update: {w_0: 0, w_1: -1, w_2: -3}
Accuracy: 40%

Neural Networks

The following is a network of linear neurons, that is, neurons whose output is identical to their net input. The numbers in the circles indicate the output of a neuron, and the numbers at connections indicate the value of the corresponding weight.

7.1. [5%] Compute the output of the hidden-layer and the output-layer neurons for the given input (0.5, 1) and enter those values into the corresponding circles.




-2
3
-2
0.5
1
2
-0.5
1
-0.5
2
0
0
-2
2
1
1
-1
0.5
1

Short Answer

1. List 3 reasons WHY machine learning is needed?

Unknown environments

Adaptability

Lazy

Autonomous

Short Answer

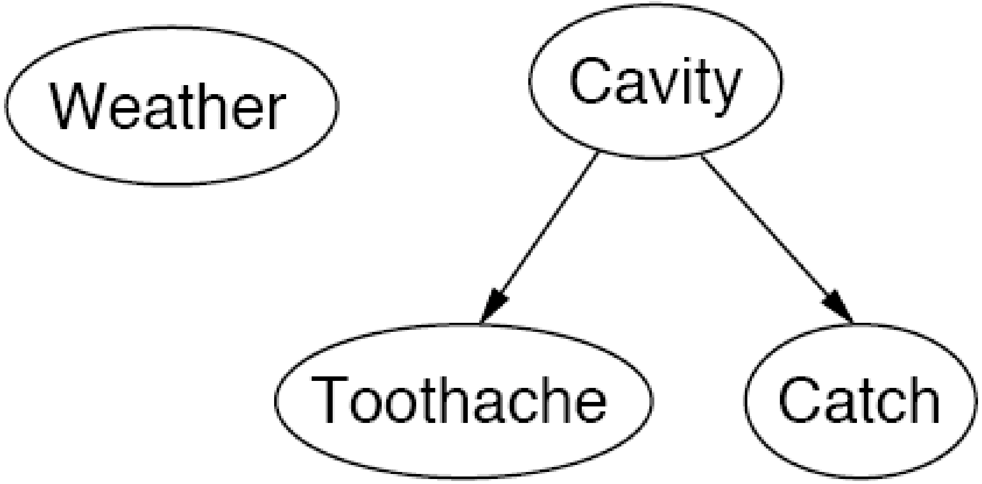
2. What is Ockham’s razor? How is it used in neural net learning?

Bias for simplest hypothesis

Prefer fewer hidden units

Independence in Bayesian Networks

Topology of network encodes (conditional) independence assertions:



Weather is independent of the other variables

Toothache and Catch are conditionally independent given Cavity

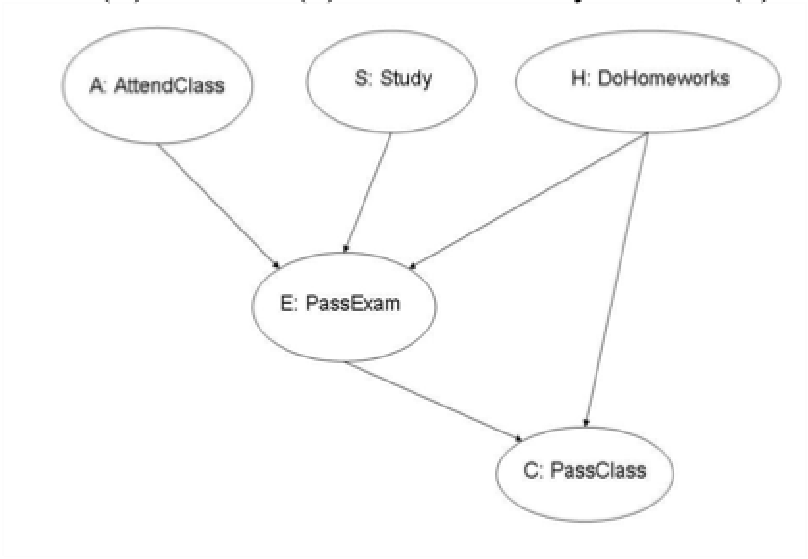
3. [15%] Bayes Network.

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).

P(A)	
+a	0.5
-a	0.5

P(S)	
+s	0.7
-s	0.3

P(H)	
+h	0.9
-h	0.1



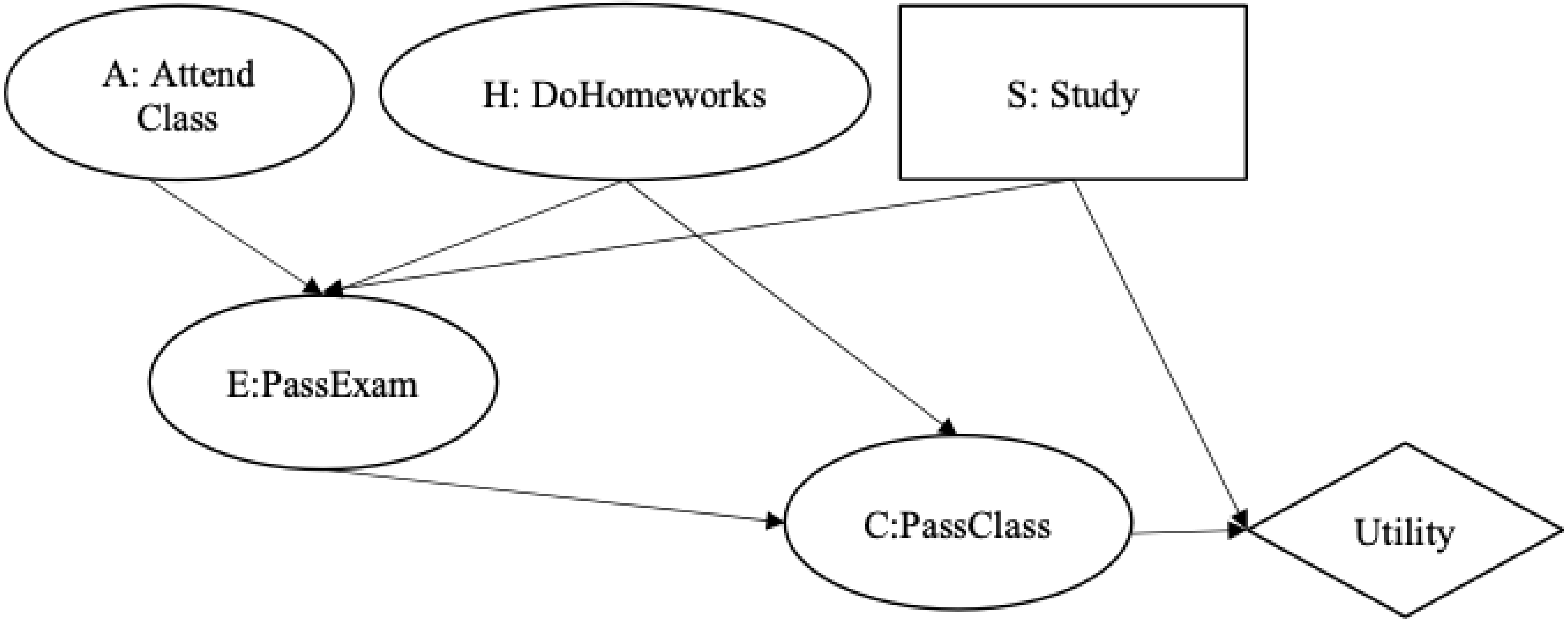
P(E|A,S,H)				
-a	-s	-h	+e	0.2
-a	-s	-h	-e	0.8
-a	-s	+h	+e	0.5
-a	-s	+h	-e	0.5
-a	+s	-h	+e	0.4
-a	+s	-h	-e	0.6
-a	+s	+h	+e	0.8
-a	+s	+h	-e	0.2
+a	-s	-h	+e	0.3
+a	-s	-h	-e	0.7
+a	-s	+h	+e	0.7
+a	-s	+h	-e	0.3
+a	+s	-h	+e	0.6
+a	+s	-h	-e	0.4
+a	+s	+h	+e	0.9
+a	+s	+h	-e	0.1

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

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

P(+a)* P(+s)* P(+h)*(+e|+a,+s,+h)*P(+c|+e,+h)

3D. [6%] Now, consider a student who has the choice to Study(S) or not Study(S) for passing Exam. We'll model this as a decision problem with one Boolean decision node, S, indicating whether the agent chooses to study. The rest 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.



When decision is+s, the probability to passclass is:

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

Where, $P(+s) = 1$

1pt

When decision is+s, the probability to not passclass is:

P(-c|+s)= P(-c,+s)= $\sum_{I ,J,K} P(a, +s, h, e, -c) = \sum_{I ,J,K} P(a) \times P(+s) \times P(h) \times P(e|a, +s, h) \times P(-c|e, h)$

where, $P(+s) = 1$

1pt

When decision is-s, the probability to passclass is:

P(+c|-s)= P(+c,-s)= $\sum_{I ,J,K} P(a, -s, h, e, +c) = \sum_{I ,J,K} P(a) \times P(-s) \times P(h) \times P(e|a, -s, h) \times P(+c|e, h)$

where, $P(-s) = 1$

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

where, $P(-s) = 1$

1pt

Expected Utility when study: P(+c|+s)*(100-50)+ P(-c|+s)*(-10-50)
1pt

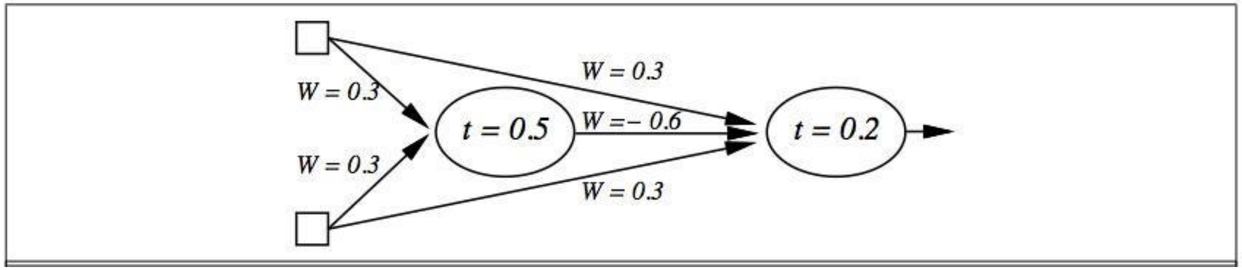
Expected Utility when not study: P(+c|-s)*(100+0)+ P(-c|-s)*(-10+0)

Neural Nets

Draw the neural network to solve the XOR function for two inputs.

Specify what type of units you are using.

(try it on paper now)



When input X = 0 and input Y = 0, what does the Unit A output? What does the Unit B output?

When input X = 0 and input Y = 1, what does the Unit A output? What does the Unit B output?

When input X = 1 and input Y = 0, what does the Unit A output? What does the Unit B output?

When input X = 1 and input Y = 1, what does the Unit A output? What does the Unit B output?

What Boolean function does this Neural Network compute?