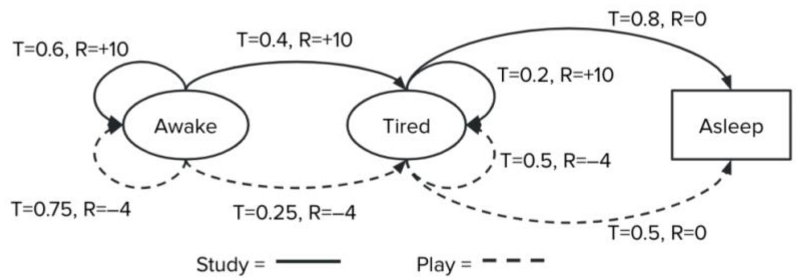
1. **Markov Decision Processes**

The Markov Decision Process below models three states – Awake, Tired, and Asleep – and two actions – Study (the solid arrows) and Play (the dashed arrows). The transition value (i.e. probability) and reward for each (s, a, s') triple is provided on the arc of each arrow. The Asleep state is a terminal state with no actions. (Alternatively, you could think of the Asleep state as having a single non-action arc looping back to Asleep with probability 1.0 and reward 0.)



Perform value iteration on this MDP up to $V_2(s)$ for states Awake and Tired (Sleep will always have value 0). Remember that $V_0(s)$ is initialized to be 0 and the update function is given by:

$$V_{k+1}(s) = \max_a \sum_{s'} T(s, a, s')[R(s, a, s') + \gamma V_k(s)]$$

To keep things simple, assume $\gamma = 1$.

I think we need to use the function to calculate:
Because we know the T and R
then, max(0.6*10   0.75*-4   0.4*10   0.25*-4)
for the awake it is 10

then max(0.2*10   0.5*-4   0.8*0   0.5*0)
for the tired it is 2

for the v2
because we get the 10 and 2
then max(0.6(10+10) 0.75(-4+10) 0.4(10+2) 0.25*(-4+2))
for the awake the max is 16.5

then max(0.2*(10+2) 0.5*(2-4) 0.8*10 0.5*10)
for the tired the max is 2.4

2. **Reinforcement Learning**

Consider the following gridworld with states $A, B, G1$, and $G2$.

Rewards

+10	+1

State names

A	B
G1	G2

From state A, possible actions are Right ($\rightarrow$) and Down ($\downarrow$). From state B, possible actions are Left ($\leftarrow$) and Down ($\downarrow$). For states $G1$ and $G2$, the only possible action is Exit. Upon exiting, we receive a reward (+10 and +1 respectively) and end at the end-of-game absorbing state X (which always has value $V(X) = 0$). The discount $\gamma = 1$ and there is no noise (i.e. all actions are deterministic).

Consider the following episodes. The following questions will use various sequences of these episodes as examples.

Episode 1 ($E1$)

s	a	s'	r
A	$\downarrow$	$G1$	0
$G1$	exit	X	10

Episode 2 ($E2$)

s	a	s'	r
B	$\downarrow$	$G2$	0
$G2$	exit	X	1

Episode 3 ($E3$)

s	a	s'	r
A	$\rightarrow$	B	0
B	$\downarrow$	$G2$	0
$G2$	exit	X	1

Episode 4 ($E4$)

s	a	s'	r
B	$\leftarrow$	A	0
A	$\downarrow$	$G1$	0
$G1$	exit	X	10

(a) Suppose the observed sequence is $\langle E1, E2, E3, E4 \rangle$. Determine the value $V(s)$ of each state using **direct evaluation** under policy π where $\pi(A) =$ Down and $\pi(B) =$ Down.

(b) Suppose the observed sequence is $\langle E3, E4, E1, E2 \rangle$. Determine the value $V(s)$ of each state using **temporal difference learning**. All values are initialized to zero and the learning rate is $\alpha = 0.5$. For reference, here is the update for each sample:

$$\text{sample} = R(s, \pi(s), s') + \gamma V^{\pi}(s')$$
$$V^{\pi}(s) = (1 - \alpha)V^{\pi}(s) + (\alpha)\text{sample}$$

(c) Consider using **Q-Learning** to learn the q-values of this gridworld. For which of the following sequences, if repeated an infinite number of times, would the q-values for *all* state-action pairs (s, a) converge to their optimal value $Q^*(s, a)$?

i. $\langle E1, E2, E1, E2, ... \rangle$
ii. $\langle E3, E4, E3, E4, ... \rangle$
iii. $\langle E1, E2, E3, E4, ... \rangle$

(a)For the E1,E2,E3,E4, G1=+10, G2=+1, the A have the possible action is right and down so A = +1, A have the possible action is left and down B=+10.
(b)For the E3,E4,E1,E2,
sample = R(s,pi(s),s’) + rVpi(s’), and Vpi(s) = (2-a) Vpi(s) + (a)sample
so we think the V(a) V(b) are 0, and put V(a) V(b) to sample function to get the V(s), every Episodes have two steps that are tow movement and exit. Then, we can get the Value about exit for every Episodes

(c)the q-value for all state-action pairs(s, a) converge to their optimal value Q*(s, a). The I follow sequences, if repeated an infinite number of times.

3. **Naive Bayes**

Consider the following labeled corpus of text messages, with punctuation and capitalization removed for simplicity:

(Spam) `you have a chance to win $100`
(Spam) `send your love with an exclusive offer`
(Spam) `you have an offer for free tickets`
(Ham) `you up`
(Ham) `i am here`
(Ham) `have you seen crazy stupid love`
(Ham) `yo terron canceled the quiz you wanna get drinks`

For the following problems, you need only represent probabilities with fractions, not computed decimal values.

(a) Compute the following probabilities:

Prior probability of spam: $P(Y = spam)$
Prior probability of ham: $P(Y = ham)$
Probability of the word `you` given spam: $P(W = you|Y = spam)$
Probability of the word `you` given ham: $P(W = you|Y = ham)$

(b) What is the probability of spam and the probability of ham given the following text: `love you`. That is, what are $P(Y = spam|X = \texttt{love you})$ and $P(Y = ham|X = \texttt{love you})$? You answer should be written as a product of probabilities (i.e. you do not have to compute the exact value of the product).

(c) Consider the text: `are you crazy`. What are the values of $P(Y = spam|X = \texttt{are you crazy})$ and $P(Y = ham|X = \texttt{are you crazy})$? (Hint: You shouldn't need to write out the entire product to determine this.) Why is this the case and how can we account for the issue observed here?

(a)
Naïve Bayes is P(Y|F1,...,Fn), and we have 3 spam and 4ham
Then P(Y=spam) = 3/(3+4) = 3/7
P(Y=ham) = 4/(3+4) = 4/7
Because we have 5 "you" in emails, and 2 are spam 3 are ham , then
P(W=you|Y=spam) = 2/3
P(W=you|Y=ham) = 3/4
(b)
Because P(Y|F1,...,Fn) = p(y)||p(f|y)
P(y=spam|x1…xn) and P(y=ham|x1…xn)
the "love you" in the spam is p(y=spam) $\prod$ p("love you" | y = spam)
the "love you" in the ham is p(y=ham) $\prod$ p("love you" | y = ham)
(c)
I think P(Y=spam | X = are you crazy)= 0.1, and P(Y=ham | X = are you crazy)=0.9 because the spam email is not sent this sentences to customer, but the friends often sent this sentences to people.

4. **Multiclass Perceptron**

Suppose for some multiclass classification problem (e.g. genres of music, categories for news articles, etc.) we have three labels, simplified here to the labels 1, 2, and 3. As such, we have three weight vectors, one per label. There are three features, so each vector contains three values. At some iteration during the update process, the weight vectors have the following values:

$$w_1 = [1, 2, -2], w_2 = [3, -2, -1], w_3 = [-1, 2, 4]$$

(a) On the next iteration, we sample a random data point x_i with label $y = 1$ and extract the feature vector $f(x_i) = [2, 2.5, -1]$. Perform the multiclass perceptron update for x_i and compute the values of any updated weight vectors.

(b) After performing the update in part (a), we sample another point x_j with label $y = 2$ and feature vector $f(x_j) = [1, -0.5, 3]$. Perform the multiclass perceptron update for x_j and compute the values of any updated weight vectors.

5. **Bias**

It is easy to think that machine learning algorithms are objectively correct because they are based on data. However, while the algorithm may optimally separate, cluster, or otherwise detect patterns in data, a key thing to keep in mind is how biased your data can be. As an extreme example, if a classifier is trained on a data set that only contains pictures of dogs, it will not be able to recognize pictures of cats.

For each of the following data sets, state how the data set may be biased for the task at hand:

(a) Data: A hand-selected set of student essays for an English class
Task: Autograding English essays

(b) Data: Collection of hand-drawn sketches by people who were told to draw a "shoe"
Task: Classifying a sketch as being a shoe or not

(c) Data: Former presidents of the United States
Task: Predicting who will be president next

4.

(a)w1 = [1,2,-2], w2=[3,-2,-1], w3=[-1,2,4]
F(x) = [2,2.5,-1] and y = 1, then f(x) = [1, 2.5,-1] is similar with w1

(b)w1 = [1,2,-2], w2=[3,-2,-1], w3=[-1,2,4]
F(x) = [1,0.5,3] and y = 2, then f(x) = [2, 1,6] is similar with w3

5.

(a)For the auto grading English essays, the system maybe will not be able to auto grade some “words” that probably never appear in the past.

(b)For the Classifying a sketch as being a shoe or not, the system might will not able to be classify. If system only has leather shoes in his data, it can't recognize the high heels shoe.

(c) For the Predicting who will be president next, the system might will not able to predicted, when the system do not know the Public opinion began to change and Political emergency.