5512, Spring-2019
ASSIGNMENT 4:
Assigned: 04/06/19 Due: ~~04/17/19~~ 04/14/19 at 11:55 PM (submit via Canvas, you may scan or take a picture of your paper answers) Submit only pdf or txt files (for non-code part), separate submission for code files
Show as much work as possible for all problems!

Problem 0. (5 points)
Go on Canvas and fill out "are you going to do a project?".

Problem 1. (30 points)
(1) Assume job A will pay N(0,1) money and job B will pay N(0.5, 0.5) money. Does job B stochastically dominate job A?

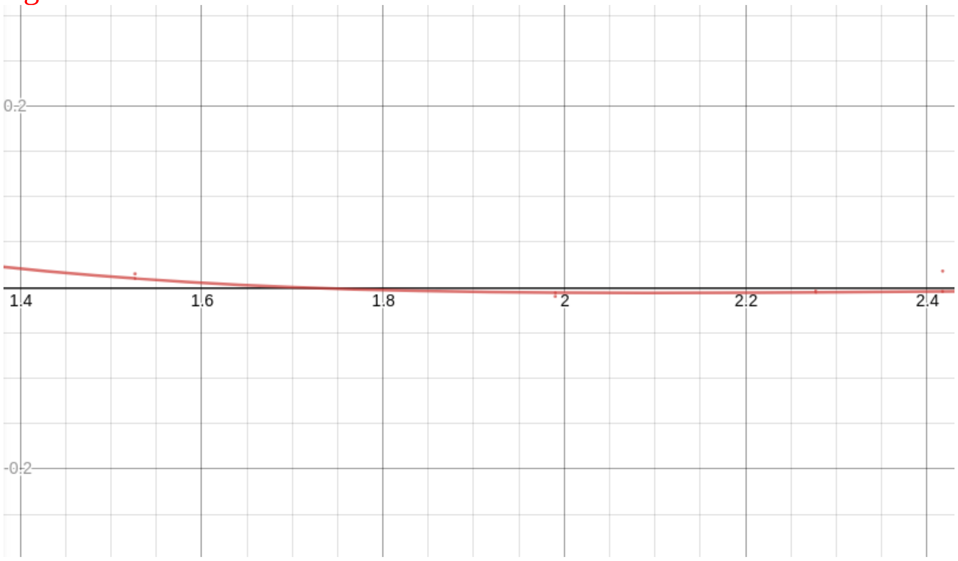
(2) If two jobs play $N(\mu_x, \delta_x^2)$ and $N(\mu_y, \delta_y^2)$ for jobs X and Y. Under what conditions does job Y stochastically dominate job X?

(3) Is it possible for a uniform distribution to stochastically dominate a normal distribution? Is it possible for a normal distribution to stochastically dominate a uniform distribution? Justify.

Solution:
(1 & 2)
Since the we want P(var1 < x) >= P(var2 < x), where P(var < x) = integral{-infinity to x} 1/sqrt(2*pi*sigma^2) * e^(-1/2 * (x-mu)^2/sigma^2).

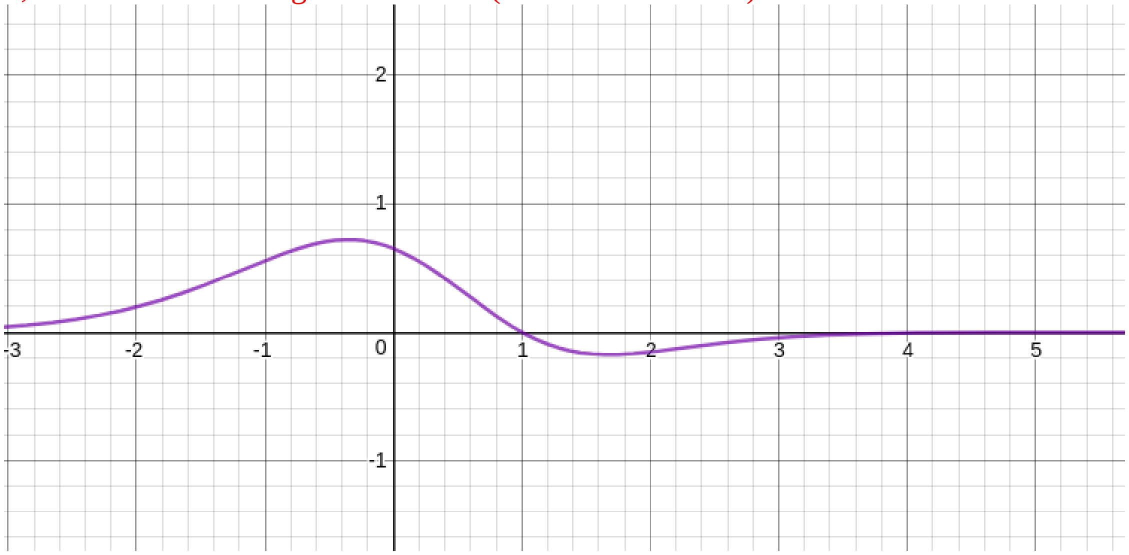
So for (1), is:
1/sqrt(2*pi) integral{-infinity to x} e^(-1/2 *x^2) – 1/sqrt(0.5)*e^((x-0.5)^2) >= 0

Plotting this we get:



Sure enough, if we plug in 2, we have P(N(0,1) < 2) = 0.9772, P(N(0.5,0.5) < 2) = 0.9831, so job B does not dominate.

(2) Overall, the general graph looks like this (N(0,2) vs N(0.5, 0.5)), where for it to be stochastically dominant, it cannot have a "negative" section (i.e. cross the x-axis).



Generalizing this to a range of parameters (you can assume 0 is always one mean, as only the difference between means matters), we want to inspect: N(0,σ_1^2) vs. N(μ,σ_2^2).

For it to be stochastically dominant, it must be true that $\mu>0$ (obviously the mean needs to be to the right), but beyond this it is hard to make any generalizations. You can get stochastic dominance both when $\sigma_1^2>\sigma_2^2$ or $\sigma_2^2>\sigma_1^2$.

The for it to be stochastically dominant you need approximately $|\sigma_1^2-\sigma_2^2| < \mu/2$. (Found by sampling the 3-parameter space and analyzing.)

(3) No, neither of these cases are possible. Assume the range of the uniformly distributed variable is [a,b]. Since the normal distribution goes from -∞ to ∞... P(normal<a-1)>0 but P(uniform<a-1) = 0. On the other hand, P(normal < b+1) < 1 but P(uniform < b+1) = 1. So these lines must cross somewhere (as normal starts larger, but ends smaller). Since they cross, they do not dominate.

Problem 2. (15 points)
Suppose you go to a casino and there are three slot machines. The three slot machines have rewards as follows (random variables):
Slot X = [(10%, \$20), (30%, \$5), (40%, \$1), (20%, \$0)]
Slot Y = [(5%, \$40), (25%, \$4), (30%, \$2), (40%, \$0)]
Slot Z = [(25%, \$10), (25%, \$5), (25%, \$2), (25%, \$0)]

It costs \$4 to play any one of the slot machines and you are only allowed to play a single time (between all slot machines, not once per slot machine). You can see the three slot machines, but you have no idea which machine corresponds to which random variable. If you had the option to choose one of these slot machines and identify it (i.e. be able to determine if it is "Slot X", "Slot Y" or "Slot Z"), how much would you pay for to have this identification done?

Solution:
E[X] = 3.9
E[Y] = 3.6
E[Z] = 4.25

So if we play without any information, we would choose one of these machines at random to get on average (expected value):
1/3*3.9 + 1/3*3.6 + 1/3*4.25 = 3.916667. This means we should not play and keep our $4

If we got to choose a machine, there is a 1/3 chance that we would point to “X” and identify it:
Playing just X: average reward = 3.9
Playing Y or Z randomly: average reward = 1/2*3.6+1/2*4.25 = 3.925
... So we would play randomly Y or Z and get 3.925 in this case (found “X”)

If we identified Y:
Play just Y: average = 3.6
Play X or Z: average = 1/2*3.9 + 1/2*4.25 = 4.075
So best for this case is 4.075 average

If we identify Z:
Play just Z: average = 4.25
Play X or Y: average = 1/2*3.9 + 1/2*3.6 = 3.75
Best is 4.25

Since the identification happens at random, we have a random variable:
[(1/3, 3.925), (1/3, 4.075), (1/3, 4.25)], which has expected value: 1/3*3.925 + 1/3*4.075 + 1/3*4.25 = 4.0833333

So in this case we should play and we are willing to pay 4.08333 – 4 (best option of just keeping money with no info) = 0.083333 dollars for this identification (8 cents). This is the “value of perfect information” equation.

Problem 3 & 4 use the following Markov Decision Process (MDP) with rewards as shown:

Assume that when moving there is a 70% chance to end up where you want to go and a 15% chance to end up 90 degrees left/right of where you want to go.

So for example, if you intend to go “up”: there is a 70% chance you go up, 15% chance you go right and 15% chance you go left

You may assume that when you hit the 50 or -50, that you cannot move anymore and just get that reward then stop the “game”.

■	50	■
■	0	-3
-50	-1	-10
■	-3	-2

Problem 3. (20 points)
For all parts of this problem, use the MDP given above and assume γ=0.8.

(1) Run value iteration until convergence and report the utilities for every state.

(2) If your initial guesses for the utilities are all zero, what is the least amount of iterations to find best policy?

(3) On the iteration you found in part (2), what is largest difference the estimated utility vs. actual utility (i.e. between parts (1) and (2)). How does this compare to the theoretical bound for the utility?

Solution:
(1) After convergence you should get the following values/actions:

34.3793	18.7819	^ <
12.5281	2.29684	^ ^
4.70433	1.03414	^ <

(2) These actions happen after 4 iterations:

33.6892	16.7274	^ <
11.0047	-1.38432	^ ^
1.47738	-2.97299	^ <

(3) The absolute difference between the answers to part (1) and (2) are:
0.690148 2.05454 1.52336 3.68116 3.20863 3.87278

... which is a maximum of 3.87278
Our initial guesses (all 0) had a difference of 34.3793 from the correct utility, which should shrink by 0.8 each iteration. Thus after 4 iterations, the difference should less than 34.3793*0.8^4 = 14.08176128.
Sure enough our difference was less: 3.87278 < 14.08176128

Problem 4. (15 points)
Use policy iteration to solve the MDP shown above. Start by assuming all actions are "Up" and γ=0.8.

Solution:
If the six boxes with unknown utility are called:

a	b
c	d
e	f

... Then with up arrows we get:
a = 0 + 0.8*(0.7*50 + 0.15*a + 0.15*b)
b = -3 + 0.8*(0.7*b + 0.15*a + 0.15*b)
c = -1 + 0.8*(0.7*a + 0.15*-50 + 0.15*d)
d = -10 + 0.8*(0.7*b + 0.15*c + 0.15*d)
e = -3 + 0.8*(0.7*c + 0.15*e + 0.15*f)
f = -2 + 0.8*(0.7*d + 0.15*e + 0.15*f)

Solving this system gives the following utilities:
a = 32.185629
b = 2.694611
c = 10.030217
d = -8.281127
e = 1.982104
f = -7.272249

.. which has the following best actions:
a = ^
b = <
c = ^
d = <
e = ^
f = <

... which gives the following system of equations:
a = 0 + 0.8*(0.7*50 + 0.15*a + 0.15*b)
b = -3 + 0.8*(0.7*a + 0.15*b + 0.15*d)
c = -1 + 0.8*(0.7*a + 0.15*-50 + 0.15*d)
d = -10 + 0.8*(0.7*c + 0.15*b + 0.15*f)
e = -3 + 0.8*(0.7*c + 0.15*e + 0.15*f)
f = -2 + 0.8*(0.7*e + 0.15*f + 0.15*d)

... which give the following utilities:
a = 34.312444
b = 18.291254
c = 12.096304
d = -0.988872
e = 4.336565
f = 0.352059

... which give the following best actions:
a = ^
b = <
c = ^
d = ^
e = ^
f = <

These are technically the best actions, but if we hadn’t run value iteration before this we wouldn’t know this fact. So to properly follow the algorithm we should do this one more iteration to confirm that the actions don’t change.

So we get a system of equations (only “d” changes from last time):
a = 0 + 0.8*(0.7*50 + 0.15*a + 0.15*b)
b = -3 + 0.8*(0.7*a + 0.15*b + 0.15*d)
c = -1 + 0.8*(0.7*a + 0.15*-50 + 0.15*d)
d = -10 + 0.8*(0.7*b + 0.15*c + 0.15*d)

e = -3 + 0.8*(0.7*c + 0.15*e + 0.15*f)
f = -2 + 0.8*(0.7*e + 0.15*f + 0.15*d)

... which give the following utilities (same actual utilities from previous problem):
a = 34.379347
b = 18.781881
c = 12.528055
d = 2.296841
e = 4.704327
f = 1.034141

... which have the following best actions:
a = ^
b = <
c = ^
d = ^
e = ^
f = <

Problem 5. (20 points)
Assume you have the following POMDP with rewards as shown below. Assume your initial guess of where you are in this POMDP is 20% in the top-left, 30% in the top-right and 50% in the bottom-right (also shown below). Assume the movement is the same as in Problems 3 & 4, but the only actions are moving “left” or “down”. There is a boolean evidence variable, e, and P(e|s) is shown for all possible states in the third picture.

Find all possible belief states that result from taking two actions and their associated likelihoods. Which sequence of actions is best if you only take two actions?

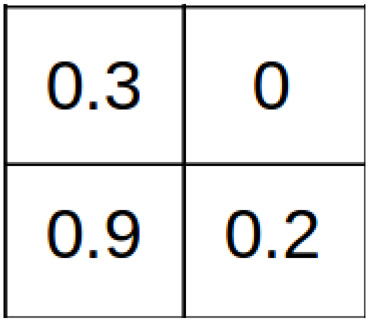
Rewards:

-1	-4
2	-2

Initial guesses for states:

20%	30%
0%	50%

P(e|s):



0.3	0
0.9	0.2

Solution:

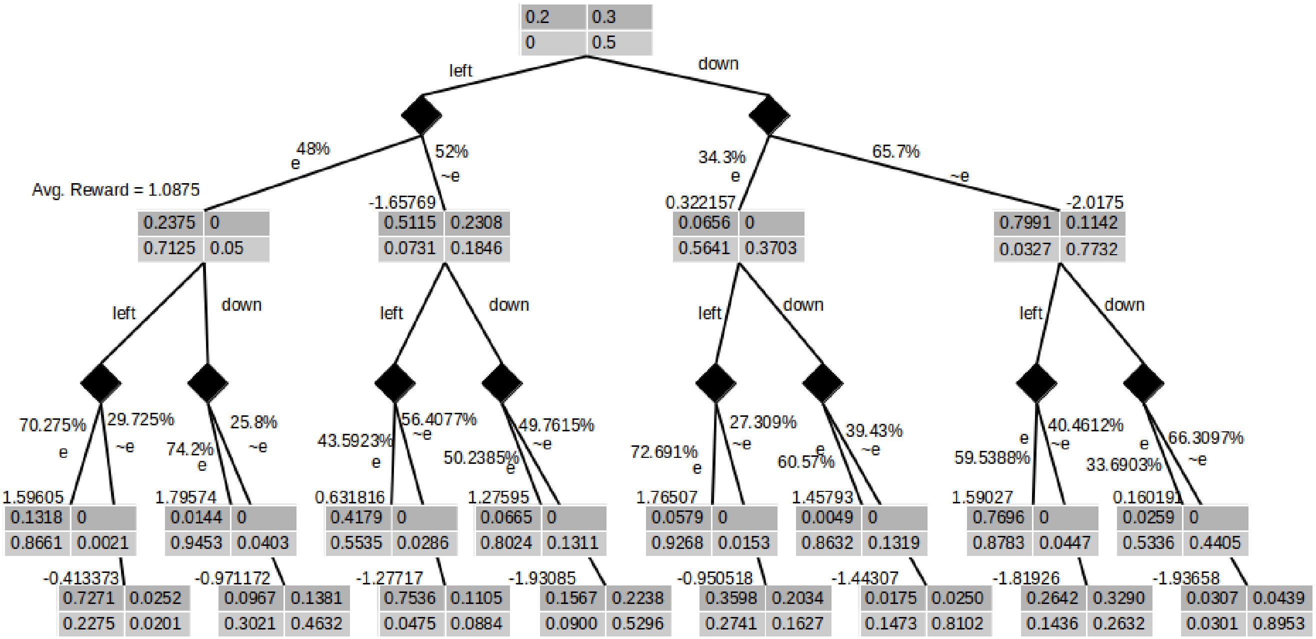
For going left, you get probability of landing in ([initial prob] * [transition prob]):
Top left (TL) = P(TL|l) = .2*.85 + .3*.7 + 0*.15 + .5*0 = 0.38
Top right (TR) = P(TR|l) = 0.2*0 + 0.3*0.15 + 0*0 + 0.5*0.15 = 0.12
Bottom left (BL) = P(BL|l) = 0.2*0.15 + 0.3*0 + 0*.85 + .5*.7 = 0.38
Bottom right (BR) = P(BR|l) 0.2*0 + 0.3*0.15 + 0*0 + 0.5*.15 = 0.12

.. We then weight by the evidence:
P(TL|l,e) = 0.38 * 0.3 = 0.114
P(TR|l,e) = 0.12 * 0 = 0
P(BL|l,e) = 0.38*0.9 = 0.342
P(BR|l,e) = 0.12*0.2 = 0.024

The probability to going to this belief state (call it b_{LE}) is the sum over the states:
P(b_{LE}) = 0.114+0+0.342+0.024 = 0.48

The actual probabilities for b_{LE} just need to be normalized (so the four states add up to 100%):
TL in b_{LE} = 0.114/0.48 = 0.2375
TR in b_{LE} = 0/0.48 = 0
BL in b_{LE} = 0.342/0.48 = 0.7125
BR in b_{LE} = 0.024/0.48 = 0.05

We repeat this process for belief state -> action -> evidence to get the following depth 4 tree:



Average reward b_{LELE} is computed as “sum_states P(state)*R(state)”:
0.1318*-1 + 0*-4 + 0.8661*2 + 0.0021*-2 = 1.5962 (non-rounded is 1.59605). This computation is shown for all states.

We start at the bottom of the tree and see that the probability and rewards of b_{LELE} and $b_{LEL\sim E}$ are: [(0.70276, 1.59605), (0.29725, -0.413373)]
So we find the expected value of b_{LEL} to be = 0.70276*1.59605 + 0.29725*-0.413373 = 0.99875
For E[b_{LED}]= 0.742*1.79574 + 0.258*-0.971172 = 1.08187
From this we conclude that from b_{LE}, we would prefer “down” on average.

For the expected values of the other 2nd actions are:
E[$b_{L\sim EL}$] = -0.445
E[$b_{L\sim ED}$] = -0.319808 (preferred)

E[b_{DEL}] = 1.02347 (preferred)
E[b_{DED}] = 0.314067

E[$b_{D\sim EL}$] = 0.210731 (preferred)
E[$b_{D\sim ED}$] = -1.23018

Now we can tell that if we go “left” and see “e” that we would prefer “down” for an average value of 1.08187. The value of b_{LE} is 1.0875 without considering future actions, so we the value of 1st action with 2nd action: 1.08187+ 1.0875 = 2.16937 (which happens 48% of the time).

If we go “left” and see “~e” we would again go “down” for an average of -0. 319808. Just “left” and “~e” has an average reward on the first action of -1.65769, with the 2nd action (“down”) worth -0.319808, so total is: -1.977498. (happens 52% of time)

This gives us another random variable for just going “left”:
[(0.48, 2.16937), (0.52, -1.977498)], which has expected value of 0.48*2.16937+0.52*-

1.977498=0.01299864.

Next we do a similar situation for the 1st “down” action:
Total for “down” and “e” (34.3%) is 0.322157+ 1.02347 = 1.345627
Total for “down” and “~e” (65.7%) is -2.0175+ 0.210731 = -1.806769

So on average down is:
E[(0.343, 1.345627), (0.657, -1.806769)] = 0.343*1. 345627 + 0.657*-1.806769 = -0.725497172

This choice of “down” is on average worse, so the best action sequence is “left” then “down”.