CS5100 HOMEWORK 6

Sudharshan Subramaniam Janakiraman

08/01/2020

THEORY

1. .
   1. Pat is more likely to have a better car than chriss because of more sample. That being said patt is more likely to be sad for the same reason. More the sample more is the disappointment. From figure 16.3 from Artificial Intelligence : A modern Approach 3rd edition the person in said to be 0.85 time Standard deviation sad if k = 3 and 2 times standard deviation sad if k = 30 Here for pat k = 10. From the graph we can say that patt will be roughly 1.5 to 1.6 times standard deviation sad

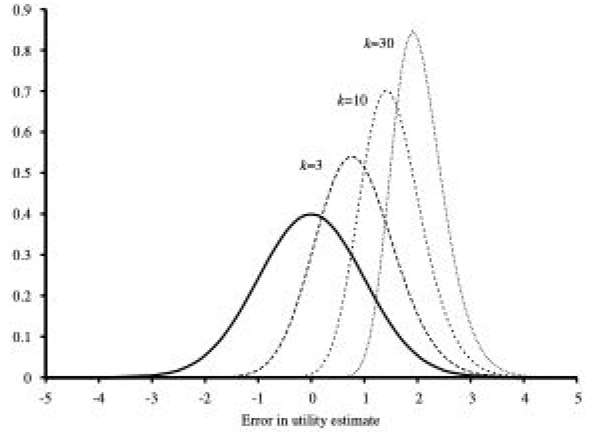
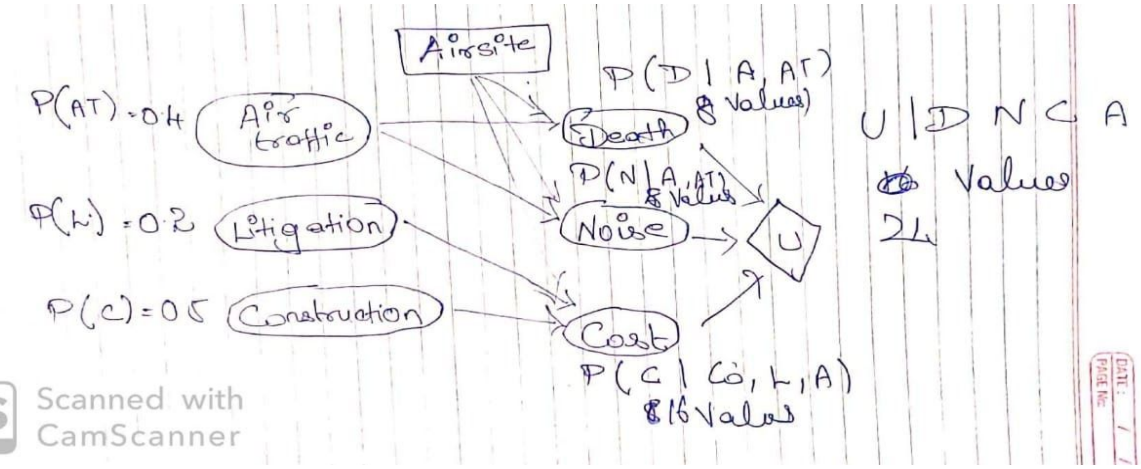


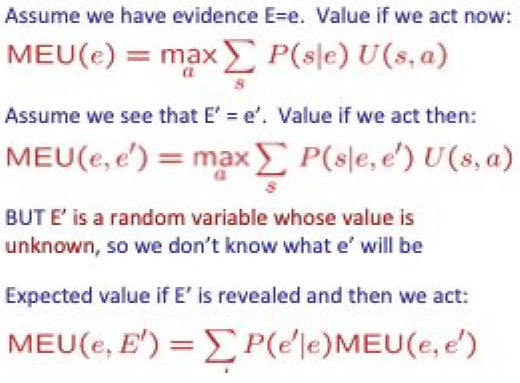
Figure 16.3 Plot of the error in each of k utility estimates and of the distribution of the maximum of k estimates for $k = 3$, 10, and 30.

2. Networks 2 and 3 correctly represents **P**(F lavor, W rapper, Shape) because in network 2 wrapper is connected to shape and flavour and shape is connected to flavour making it fully connected and in network 3 shape and wrapper are selected randomly for the flavour and shape is not dependent on wrapper and viceversa making all three variable independent

3. The decision node Airsite can take any one of the 3 sites as its decision A1, A2, A3
The chance nodes are affected by their conditions , Here the Cost, deaths and noise probabilities are affected by litigation, air traffic construction which can be considered as the hidden nodes and the Utility depends upon Cost, Death and Noise values. The below diagram represents the state along with the utility value



P (AT) ⇒ Probability of air traffic happening P(L) ⇒ probability of litigation
P(C0) ⇒ Probability of construction happening Airsite → Decision node which can take 3 values A1, A2, A3
P(D | A, AT) probability of death given Air Traffic and particular Airsite (6 values)
P(N | A, AT) probability of Noise given Air Traffic and particular Airsite (6 Values)
P(C | A,L,C0) probability if cost given a particular air traffic, construction and litigation (12 values)
U | D N C A : utility value given the Death . Noise, cost and Decision of airsite 24 values
With this data the Expected Utility and Maximum Expected Utility can be found using the following formula (referred from ucberkely notes) (LHS : Maximum EU : RHS : Maximum of all EUS)



Assume we have evidence E=e. Value if we act now:

$$\text{MEU}(e) = \max_a \sum_s P(s|e)\, U(s,a)$$

Assume we see that E' = e'. Value if we act then:

$$\text{MEU}(e,e') = \max_a \sum_s P(s|e,e')\, U(s,a)$$

BUT E' is a random variable whose value is unknown, so we don't know what e' will be

Expected value if E' is revealed and then we act:

$$\text{MEU}(e,E') = \sum_{e'} P(e'|e)\text{MEU}(e,e')$$

2. .
    1. CartPole is controlled by a controller which controls x(Location of cartpole) , theta(Angle of orientation), xdot(linear velocity), theta dot(Angular velocity) which can be considered as the game state.The agent will be given with more and more reward if it can keep the pole vertical and prevents it from falling down.Game Over when the pole falls down and agent is given with a negative reward when the game ends. The agent can move either left or right → actions of the agent Now for any particular state s there will be an action list A contationg all the possible actions that can be taken from s to go to next stateThis state action pair (s,a) can be assigned with reward and program it in such a way that the agent tries to maximize the reward ⇒ keep the pole from falling down. This means the agent has to learn the right set of actions at every instant. This can be done using q learning where the reward for taking an action a to move from state s to s' can be stored in Qmatrix Q(s,a). This Q value for every state action pair is initially zero and gets updates as and when the agent learns what is the best move from that particular state. This can be computed by the formula given below

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

        Ex: let us assume we are in state A and make a movement to state B by finding the best possible action from the Qmatrix. Now we compute Q values for B state based on possible values from B and use that value to update the Q value of A state. This method is known as Q Learning . it is an example of Active Reinforcement learning where the agent learns and observes simultaneously
    2. Passive Reinforcement learning is a kind of reinforcement learning where the agent interacts with the world and makes observation first and between states I.e., instead of updating the q matrix with the best possible value of its just updates the q value with the allowed possible action from that state. This means the q value is updated only if it's affected. This Q learning changes to Temporal Difference learning which gets computed by the below formula

        $$U^{\pi}(s) = U^{\pi}(s) + \alpha(R(s) + \gamma U^{\pi}(s') - U^{\pi}(s))$$

The difference between passive RL and active RL is that in active RL the agents learns and observes simultaneously where as in passive RL the agent observes first and learns in between transitions or states