CS 181 Artificial Intelligence (Fall 2020), Final Exam

Instructions

- Time: 15:20 – 17:00 (100 minutes)
- This exam is closed-book, but you may bring one A4-size cheat sheet. Put all the study materials and electronic devices (except a calculator) into your bag and put your bag in the front, back, or sides of the classroom.
- Two blank pieces of paper are attached, which you can use as scratch paper. Raise your hand if you need more paper.

1 Multiple choices (10 pt)

Each question has one or more correct answers. Select all the correct answers. For each question, you get 1 point if your select all the correct answers and nothing else, 0 point if you select one or more wrong answers, and 0.5 point if you select a non-empty proper subset of the correct answers.

1	2	3	4	5

6	7	8	9	10

1. Which of the following statements about Markov process and the hidden Markov model (HMM) is/are correct?

   A. The transition model for a second-order Markov process is the conditional distribution $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-2},\mathbf{X}_{t-1})$

   B. Both the time complexity and the space complexity of the Viterbi algorithm are linear in the length of the sequence.

   C. Both the time complexity and the space complexity of the filtering algorithm are linear in the length of the sequence.

   D. Every hidden Markov model can be represented as a dynamic Bayesian network with a single state variable and a single evidence variable.

   E. None of the above.

2. Which of the following statements about Markov Logic is/are correct?

   A. Markov logic only models hard constraints.

   B. A Markov logic network is a set of pairs (F, w), where F is a formula in first-order logic and w is a real number.

   C. If all the weights in a Markov logic network are the same, then it becomes first-order logic.

   D. Markov logic does not allow contradictions between formulas.

E. None of the above.

3. Which of the following statements are true for an MDP?

   A. If the only difference between two MDPs is the value of the discount factor then they must have the same optimal policy.

   B. For an infinite horizon MDP with a finite number of states and actions and with a discount factor γ that satisfies $0 < \gamma < 1$, value iteration is guaranteed to converge.

   C. When running value iteration, if the policy (the greedy policy with respect to the values) has converged, the values must have converged as well.

   D. MDP does not satisfy the memoryless property.

   E. None of the above.

4. Bob notices value iteration converges more quickly with smaller γ and rather than using the true discount factor γ, he decides to use a discount factor of $\alpha\gamma$ with $0 < \alpha < 1$ when running value iteration. Mark each of the following that are guaranteed to be true:

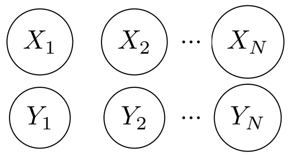
   A. If the MDP's transition model is deterministic and the MDP has zero rewards everywhere, except for a single transition at the goal with a positive reward, then Bob will still find the optimal policy.

   B. If the MDP's transition model is deterministic, then Bob will still find the optimal policy.

   C. Bob's policy will tend to more heavily favor short-term rewards over long-term rewards compared to the optimal policy.

   D. None of the above.

5. Let $X_1, \cdots, X_N$ represent words of a length-N sentence and $Y_1, \cdots, Y_N$ represent the words' corresponding labels. Which of the following statements about sequence labeling models is/are correct?



   A. Recall the simplest method for sequence labeling: for each word, predict its most frequent label. For this method, we have $Y_i \perp\!\!\!\perp X_j$ for $i \neq j, 1 \leq i \leq N, 1 \leq j \leq N$.

   B. For HMM, we have: $X_i \perp\!\!\!\perp X_j \mid Y_i$ for $i \neq j, 1 \leq i \leq N, 1 \leq j \leq N$.

   C. For MEMM, we have: $X_i \not\!\perp\!\!\!\perp X_j \mid Y_i$ for $1 \leq i < j \leq N$.

   D. MEMM prefers labels with higher numbers of transitions to other labels.

   E. Unlike MEMM, CRF is based on global normalization.

6. Which of the following statements about overfitting is/are correct?

   A. Overfitting occurs when the model fits the training data very closely but fits the test data poorly.

   B. Overfitting occurs when the model always performs bad on the held-out data.

   C. Overfitting could happen when there are too few training examples or there are too few attributes.

   D. Making the model more expressive can avoid overfitting.

   E. None of the above.

7. Which of the following statements about supervised learning is/are correct?

   A. The naïve Bayes assumption takes all features to be independent given the class label.

   B. We can use smoothing or regularization to improve model's generalization.

C. A neural network with a sufficient number of neurons and a complex enough architecture can approximate any continuous function to any desired accuracy.

D. In Convolutional Neural Networks, different hidden units organized into the same “feature map” share weight parameters.

E. None of the above.

8. Which of the following statements about unsupervised learning is/are correct?

   A. K-means algorithm can always find the global optimal.

   B. EM algorithm can always find the global optimal.

   C. If we assume that all the Gaussians of a Gaussian mixture model are spherical and have identical weights and covariances, EM algorithm will degrade to k-means.

   D. In E-step of the EM algorithm, we compute new parameter values to maximize expected log likelihood based on distributions over hidden variables.

   E. None of the above.

9. In general, which of the following is/are necessary or Q-Learning to converge to the optimal Q-values?

   A. Every state-action pair is visited infinitely often.

   B. The learning rate α (weight given to new samples) is decreased to 0 over time.

   C. The discount γ is less than 0.5.

   D. Actions get chosen according to $\arg\max_a Q(s, a)$.

10. In which of the following Reinforecement Learning method(s), the policy is not given?

    A. Direct Evaluation

    B. Temporal difference learning

    C. Q-learning

    D. None of the above.

Solution:

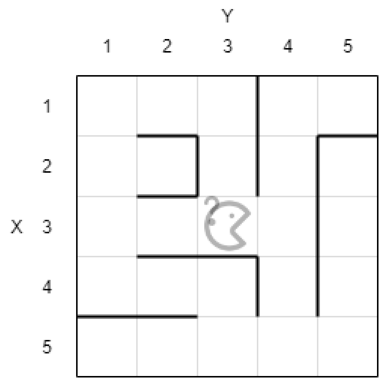
1. ABD
2. B
3. B
4. AC
5. ABE
6. A
7. ABCD
8. E
9. AB
10. C

2 (Temporal models) Pacman and Casper II (10 pt)

"Pacman? Pacman!"

Casper hasn't seen pacman for a long time. After mid-term, CS181 staff equipped the pacman with a ghost detector, so that pacman can easily hide from the ghosts. Casper always helps pacman in the game, but pacman doesn't know this. For months, casper's just wandering alone in the maze, missing those happy days playing with pacman. In this problem, you'll help casper to find pacman.

For simplification, you will only work on a simple 5×5 grid map, which is shown below:

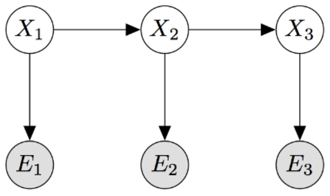


2.1 HMM (6 pt)

Casper does not have a detector, but he has a wireless noisy monitor of the maze. The monitor will display the position of pacman with a probability θ, and display a random position with a probability $1-\theta$. For example, if pacman is at position $(3,3)$ and $\theta = 0.25$, then the monitor would display $(3,3)$ with probability $0.25 + (1-0.25) \div 25 = 0.28$ and display any other position (such as $(1,3)$) with probability $(1-0.25) \div 25 = 0.03$.

He also knows that pacman will wander in the maze if no ghost appears (you can assume that there is no ghost now, so pacman is now wandering.) i.e. pacman will choose to move in any possible direction or stay still with the same probability. For example, if pacman is at $(4,3)$ at $T = t$, then at $T = t+1$ he would appear at $(4,2)$, $(4,3)$, $(5,3)$ with the same probability $\frac{1}{3}$.

Consider the typical Hidden Markov Model (HMM) graph structure shown below.



Here, X_i represents the position of pacman at time $t = i$. E_i represents the monitor result at time $t = i$. Recall the Forward algorithm:

Elapse Time: $P(X_t|e_{1:t-1}) = \sum_{x_{t-1}} P(X_t|x_{t-1})P(x_{t-1}|e_{1:t-1})$

Observe: $P(X_t|e_{1:t}) = \frac{P(e_t|X_t)P(X_t|e_{1:t-1})}{\sum_{x_t} P(e_t|x_t)P(x_t|e_{1:t-1})}$

Assume $\theta = \mathbf{0.25}$. Initially we assume that pacman position is uniformly distributed. That means $\forall x, y \in \{1,2,3,4,5\}, P(X_1 = (x,y)) = 0.04$. Now, time to do some simple calculation!

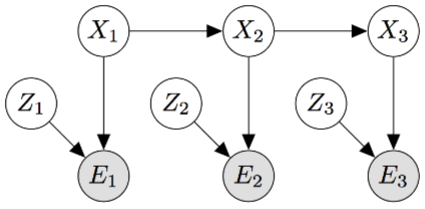
Please write down the decimal number with 4 decimal places such as 0.2500 in the box below. Since auto-grading will be applied, any ambiguous or out-of-region answer will be treated as 0 points.

(1) (1 pt) $P(X_2 = (2,2) | X_1 = (2,2)) =$ $\boxed{}$

(2) (1 pt) $P(X_1 = (2,2) | E_1 = (2,2)) =$ $\boxed{}$

(3) (1 pt) $P(X_2 = (2,2) | E_1 = (2,2)) =$ $\boxed{}$

However, due to the poor network in ShanghaiTech, the monitor is unstable. If the monitor is connected to channel A (Administration Center channel), the signal is normal (θ is good); if the monitor is connected to channel T (Teaching Center channel), the signal is noisy (θ is small). Suppose Z_t is the channel that monitor is connected to at time t, but we do not know what Z_t is. Now the graph may change a little bit:



Remember that Z_t is not observed. Now, casper wants to know whether the update function will change.

Please choose the correct answer by filling up the circle like ●, not ✓. Since auto-grading will be applied, any ambiguous, out-of-region answer, or answers not following the instruction will be treated as 0 points.

(4) (1.5 pt) Elapse Time:

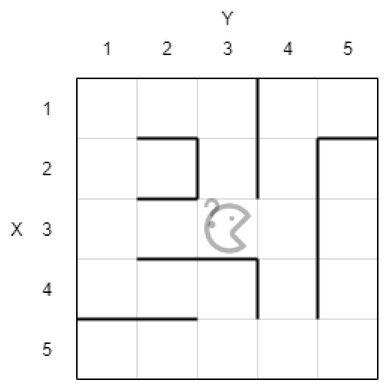
- ○ It will not change. i.e. still $P(X_t | e_{1:t-1}) = \sum_{x_{t-1}} P(X_t | x_{t-1}) P(x_{t-1} | e_{1:t-1})$.
- ○ It will change.

(5) (1.5 pt) Observe:

- ○ It will not change. i.e. still $P(X_t | e_{1:t}) = \frac{P(e_t | X_t) P(X_t | e_{1:t-1})}{\sum_{x_t} P(e_t | x_t) P(x_t | e_{1:t-1})}$.
- ○ It will change.

2.2 Particle Filtering (4 pt)

The real maze is much bigger than this 5×5 maze, so casper decides to use particle filtering to accelerate the process. For this problem, we still take the 5×5 maze as an example. Only consider the typical HMM model (same as the first one in 2.1, without considering Z_t). The maze is repeated here for your convenience.



Recall the main steps of Particle Filtering:

1. Propagate forward. Each particle is moved by sampling its next position from the transition model: $x_{t+1} \sim P(X_{t+1}|x_t)$.
2. Observe. Weight samples based on the evidence: $w = P(e_t|x_t)$.
3. Resample and repeat.

The following problems may have more than 1 correct answer. Select all correct answers. You will get half of the points if you choose a non-empty subset of the correct answers.

Please choose the correct answer by filling up the box like ■, not ✓. Since auto-grading will be applied, any ambiguous, out-of-region answer, or answers not follow the instruction will be treated as 0 points.

(1) (2 pt) Assume we observe the monitor displaying position $(2, 2)$ with $\theta = 0.25$. For a particle at position $(3, 2)$, what could the position and weight of the next particle be? Select all possible choices.

- ☐ Position $(2, 2)$, weight 0.28.
- ☐ Position $(2, 3)$, weight 0.03.
- ☐ Position $(3, 2)$, weight 0.28.
- ☐ Position $(3, 3)$, weight 0.03.
- ☐ Position $(4, 2)$, weight 0.28.
- ☐ Position $(3, 1)$, weight 0.03.
- ☐ None of the above.

(2) (2 pt) Assume a particle at position $(3, 2)$ moves to position $(3, 3)$ with weight 0.04. what could observed position from monitor and θ be? Select all possible choices.

- ☐ Position $(4, 3)$, $\theta = 0$.
- ☐ Position $(3, 2)$, $\theta = 1$.
- ☐ Position $(3, 3)$, $\theta = 0$.
- ☐ Position $(3, 4)$, $\theta = 1$.
- ☐ Position $(3, 3)$, $\theta = 0.52$.
- ☐ Position $(3, 2)$, $\theta = 0.52$.
- ☐ None of the above.

Solution:

HMM Part 1

(1) $P(X_2 = (2,2)|X_1 = (2,2)) = 0.5000$

(2) $P(X_1 = (2,2)|E_1 = (2,2)) = 0.2800$

(3) $P(X_2 = (2,2)|E_1 = (2,2)) = 0.1475$

Solution:

HMM Part 2

(4) It will not change.

(5) It will change. Actually changes to $P(X_t|e_{1:t}) = \frac{P(X_t|e_{1:t-1})\sum_{z_t} P(z_t)P(e_t|X_t,z_t)}{\sum_{x_t} P(x_t|e_{1:t-1})\sum_{z_t} P(e_t|x_t,z_t)P(z_t)}$.

Solution:

Particle Filtering

(1) Position $(3,3)$, weight 0.03. And position $(3,1)$, weight 0.03.

(2) Position $(4,3)$, $\theta = 0$. And Position $(3,3)$, $\theta = 0$.

3 POS Tagging (10 pt)

You are given a sentence: **I love eating banana**.

In this problem, you will use a linear-chain CRF to obtain the POS tags of the above sentence. A linear-chain CRF is very similar to a HMM model that we learned in class. It also has the transition model $S_T(y_{i+1}, y_i)$ (the score of transiting from y_i to y_{i+1}) and the emission model $S_E(y_i, x_i)$ (the score of the i-th word x_i and label y_i). However, these models specify scores instead of probabilities, i.e., the scores do not have to sum up to 1. (Note: Here we assume a uniform initial distribution over y_1, so it can be ignored.)

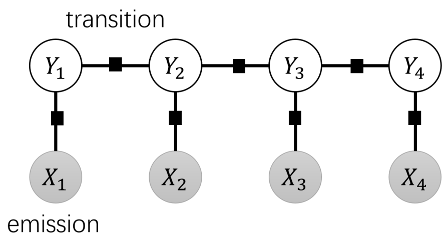


Figure 1: linear-chain CRF

The possible POS tags are: $L = \{$`PRP`, `NN`, `VBG`, `VBP`$\}$. And the transition/emission models are shown in the following tables.

y_i \ x_i	I	love	eating	banana
PRP	3	2	0	1
NN	2	3	1	2
VBG	1	3	2	0
VBP	1	2	2	1

Table 1: Emission score $S_E(y_i, x_i)$

y_{i+1} \ y_i	PRP	NN	VBG	VBP
PRP	1	1	2	2
NN	0	2	3	2
VBG	0	0	0	2
VBP	3	1	0	0

Table 2: Transition score $S_T(y_{i+1}, y_i)$

3.1 Label sequence score. (3 pt)

Calculate the score of the given sentence having the following POS tag sequence: `NN`, `NN`, `VBP`, `NN`. Hint: The score of the label sequence $y_{1:n}$ given sentence $x_{1:n}$ can be calculated as $S(y_{1:n}, x_{1:n}) = S_E(y_1, x_1) \prod_{i=1}^{n-1} S_T(y_{i+1}, y_i) \cdot S_E(y_{i+1}, x_{i+1})$.

Solution:

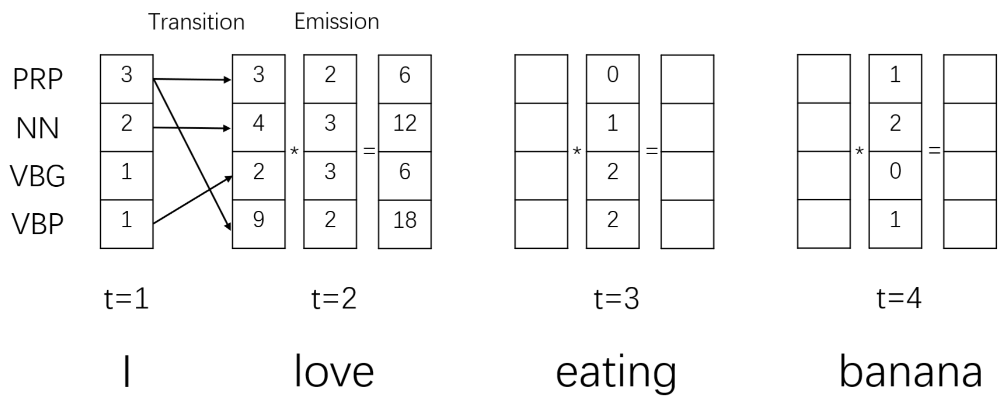
$2 \times 2 \times 3 \times 1 \times 2 \times 2 \times 2 = 96$

3.2 Best POS sequence. (7 pt)

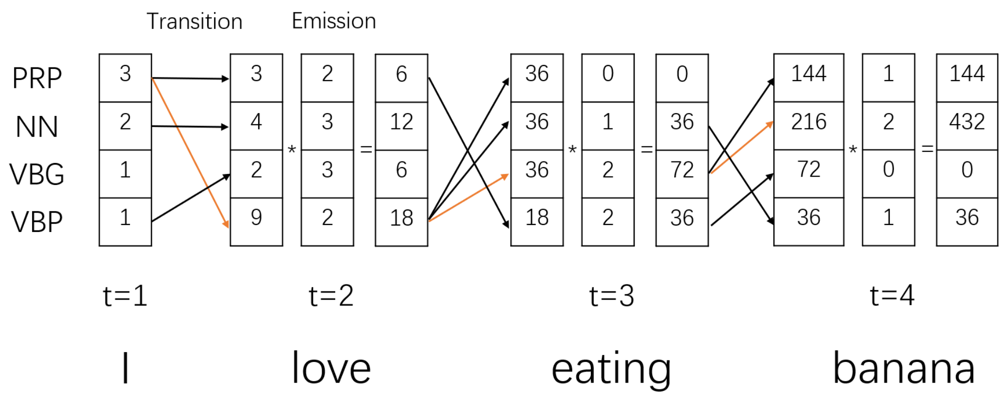
The best label sequence in CRF $y^*_{1:4} = \text{argmax}_{y_{1:4}} S(y_{1:4}, x_{1:4} =$ “I love eating banana”) can be found using the Viterbi algorithm in a similar way to that of the HMM. Recall that at step t, the Viterbi algorithm computes the highest score of paths that end up with $y_t \in L$ after reading $x_{1:t}$. We give the Viterbi decoding steps for $t = 1$ to $t = 2$ below. Please **finish the remaining steps by filling the blanks and drawing the lines** below, and then write down the **best POS sequence** $y^*_{1:4}$. We show the same transition/emission tables below for your convenience.

x_i \ y_i	I	love	eating	banana
PRP	3	2	0	1
NN	2	3	1	2
VBG	1	3	2	0
VBP	1	2	2	1

y_i \ y_{i+1}	PRP	NN	VBG	VBP
PRP	1	1	2	2
NN	0	2	3	2
VBG	0	0	0	2
VBP	3	1	0	0



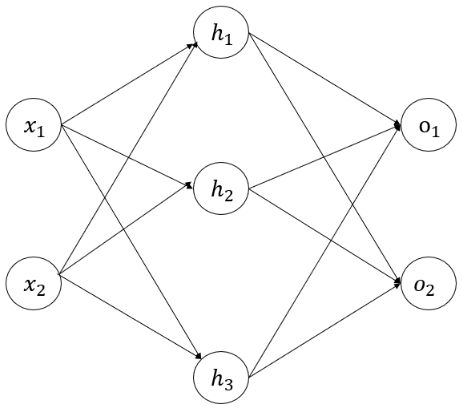
Solution:



The tag sequence: `PRP, VBP, VBG, NN.`

4 Supervised and Unsupervised Machine Learning (10 pt)

4.1 MLP (5 pt)



We have the following multi-layer preceptron. Denote the two input features as x_1 and x_2, the output of three neurons in the hidden layer as h_1, h_2, and h_3, and the output of the final layer as o_1, o_2. Let the weight from x_i to h_j be $w^{[1]}_{i,j}$, and the weight from h_j to o_k be $w^{[2]}_{j,k}$. $i, k \in \{1, 2\}$, $j \in \{1, 2, 3\}$. Assume we use the sigmoid function $\sigma(a) = \frac{1}{1+e^{-a}}$ as the nonlinear activation function for all the layers (except for the inputs). The loss function is

$$L = (o_1 - \hat{o}_1)^2 + (o_2 - \hat{o}_2)^2$$

where $\hat{o}_1$ and $\hat{o}_2$ are the target values.

4.1.1 Forward

Denote the activation of the hidden neuron j as a_j and the activation of the output neuron k as z_k. Derive the forward process to give the expression of o_1 in terms of input features and weights.

Solution:

$$\begin{aligned} o_1 &= \sigma(z_1) \\ z_1 &= w^{[2]}_{1,1} h_1 + w^{[2]}_{2,1} h_2 + w^{[2]}_{3,1} h_3 \\ h_j &= \sigma(w^{[1]}_{1,j} x_1 + w^{[1]}_{2,j} x_2), \quad j = 1, 2, 3 \end{aligned}$$

4.1.2 Backward

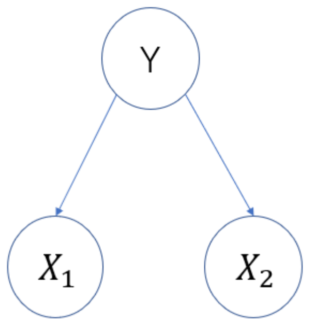
What is the gradient descent update to $w^{[2]}_{1,2}$ for the loss L defined above? Your answer should be written in terms of the presented symbols. (You can use the intermediate variables)

Solution:

$$\begin{aligned} \frac{\partial L}{\partial w^{[2]}_{1,2}} &= \frac{\partial L}{\partial o_2} \frac{\partial o_2}{\partial z_2} \frac{\partial z_2}{\partial w^{[2]}_{1,2}} \\ &= 2(o_2 - \hat{o}_2) \cdot \sigma(z_2)[1 - \sigma(z_2)] \cdot h_1 \end{aligned}$$

4.2 Naive Bayes(2 pt)

You are given a naive Bayes model, shown below, with label Y and features X_1 and X_2.



The model is trained with the following data:

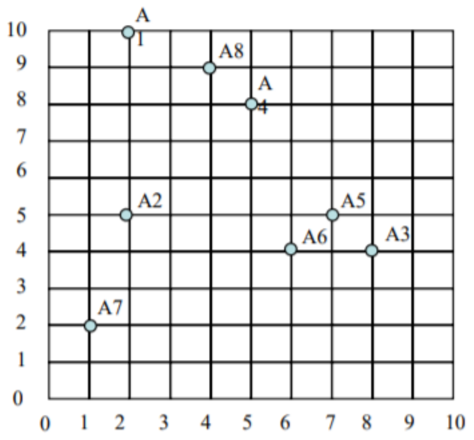
sample number	1	2	3	4	5	6	7	8	9	10
X_1	0	0	1	0	1	0	1	0	1	1
X_2	0	0	0	0	0	0	0	1	0	0
Y	0	0	0	0	0	0	0	1	1	1

What are the maximum likelihood estimates for the following probabilities?

$P(X_1 = 0|Y = 0)$: $\frac{4}{7}$ $P(X_2 = 0|Y = 0)$: 1 $P(Y = 0)$: 0.7

4.3 K-means (3 pt)

Use the k-means algorithm and Euclidean distance to cluster the following 8 examples into 3 clusters: A1=(2,10), A2=(2,5), A3=(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9). Suppose that the initial seeds (centers of each cluster) are A1, A4 and A7. Run the k-means algorithm for 1 epoch only. At the end of this epoch show: the new clusters, i.e., the examples belonging to each cluster, and the corresponding centers.



cluster1 :	A1	center1 :	(2,10)
cluster2 :	A3, A4, A5, A6, A8	center2 :	(6,6)
cluster3 :	A2, A7	center3 :	(1.5,3.5)

5 MDP and RL: Recommendation System (10 pt)

Recommendation system has been increasingly crucial as e-commerce developed. Typical recommendation systems adopt a static view of the recommendation process and treat it as a prediction problem. However, scientists from Ben-Gurion University argued that it is more appropriate to view the problem of generating recommendations as a sequential optimization problem and, consequently, that Markov decision process (MDP) provide a more appropriate model for Recommendation systems. Here in this problem, we will implement a toy recommendation system with MDP.

Suppose we only make recommendation based on the last item that customers have already bought, and we only consider three types of products, s_1, s_2, s_3, so the state space is defined by $S = \{s_1, s_2, s_3\}$. In this way, the action a_j is defined by $j \in \{1, 2, 3\}$, meaning that we recommend s_j to our customer. After that, customer in state s_i will buy s_j with probability p_{ij} and the state is transitioned into s_j. With probability $1 - p_{ij}$, customer will not buy it and state will not change. Here, we assume that customer will not continuously buy two same product, and the action space for s_i is $\{1, 2, 3\} \backslash i$.

If the recommendation i succeed, we will have a reward of $r = profit(i)$. Otherwise, the recommendation fail, waste an opportunity thus has a negative reward $r = -1$. Note that the cost of product are not constant, so it may differ from time to time.

Suppose that discount factor $\gamma = 0.5$.

5.1 MDP (4 pt)

Firstly, let's assume that $p_{ij} = 0.4 \quad \forall i, j \in \{1, 2, 3\}, i \neq j$ and $profit(i) = 10 \quad \forall i \in \{1, 2, 3\}$.

5.1.1 Transition and Reward

(2 pts) Choose the proper transition model $T(s_i, a_j, s_k)$ and reward $R(s_i, a_j, s_k)$ function for the Markov decision process described above. Mark the correct answers with ■.

$$\square T(s_i,a_j,s_k) = \begin{cases} 0.4, & \text{if } k \neq j \\ 0.6, & \text{if } k \neq i \\ 0, & \text{otherwise} \end{cases} \quad \square T(s_i,a_j,s_k) = \begin{cases} 0.4, & \text{if } k = j \\ 0.6, & \text{if } k = i \\ 0, & \text{otherwise} \end{cases} \quad \square T(s_i,a_j,s_k) = \begin{cases} 0.4, & \text{if } j = i \\ 0.6, & \text{if } k \neq i \\ 0, & \text{otherwise} \end{cases}$$

$$\square R(s_i,a_j,s_k) = \begin{cases} 10, & \text{if } k = i \\ -1, & \text{if } k = j \end{cases} \quad \square R(s_i,a_j,s_k) = \begin{cases} 10, & \text{if } k \neq j \\ -1, & \text{if } k \neq i \end{cases} \quad \square R(s_i,a_j,s_k) = \begin{cases} 10, & \text{if } k = j \\ -1, & \text{if } k = i \end{cases}$$

Solution:

$$T(s_i,a_j,s_k) = \begin{cases} 0.4, & \text{if } k = j \\ 0.6, & \text{if } k = i \\ 0, & \text{otherwise} \end{cases} \qquad R(s_i,a_j,s_k) = \begin{cases} 10, & \text{if } k = j \\ -1, & \text{if } k = i \end{cases}$$

5.1.2 Value iteration

(2 pts) Try to apply value iteration twice, which has been illustrated in the lecture, and fill in the table below.

State	s_1	s_2	s_3
V_0	0	0	0
V_1			
V_2			

Solution:

State	s_1	s_2	s_3
V_0	0	0	0
V_1	3.4	3.4	3.4
V_2	5.1	5.1	5.1

5.2 Reinforcement Learning (6 pt)

In practice, it is impossible to know what the customer is thinking about, so we can not actually know the underlying MDP model. However, we can learn them from experience. Suppose the system choose action based on some policy π and generate following samples.

t	s_t	a_t	s_{t+1}	r_t
1	s_1	a_2	s_2	10
2	s_2	a_3	s_2	-2
3	s_3	a_1	s_3	-2
4	s_2	a_1	s_1	8
5	s_1	a_3	s_3	12
6	s_1	a_2	s_1	-1

Assume a discount factor $\gamma = 0.5$ and a learning rate $\alpha = 0.4$.

5.2.1 Model Based Learning

(2 pts) In model based learning, we first need to estimate the transition model and reward function. Based on the sample above, estimate the following parameters.

$T(s_1, a_2, s_2) =$ 0.5 $\quad T(s_2, a_1, s_2) =$ 0

5.2.2 Model Free: Q-learning

(2 pts) Assume that all the Q-values are initialized to 0. Apply Q-learning we learnt in class. What are the Q-values learned by running Q-learning with all the transitions shown above? Note that we update Q-values immediately after we receive a sample.

$Q(s_1, a_2) =$ 2.96 $\quad Q(s_2, a_1) =$ 4

5.2.3 Modified Q-learning

(2 pts) In the Q-learning algorithm we performed above, it is guaranteed to converge to optimal Q-value function. Another reinforcement learning algorithm is called SARSA, and it performs the update:

$$Q\left(s_t, a_t\right) \leftarrow Q\left(s_t, a_t\right) + \alpha\left[R(s_t, a_t, s_{t+1}) + \gamma Q\left(s_{t+1}, a_{t+1}\right) - Q\left(s_t, a_t\right)\right]$$

Agent acts at each step as follows: with probability 0.5 it follows a fixed (not necessarily optimal) policy π and otherwise it chooses an action uniformly at random. Assume that updates are applied infinitely often, state-action pairs are all visited infinitely often, and learning rates α are decreased at an appropriate pace.

Is this process also guaranteed to converge to the optimal Q-value function? Mark your choice with ●.

○ Yes.

○ No.

Solution:

No.
It will converge to the Q-values of the policy pi' being executed (it is an on-policy method). In this case, pi' is the policy which follows π with probability 0.5 and acts uniformly at random otherwise.