Assignment-3

Partially Observable
Markov Decision Processes

Team Members:
Nagelli Balamallesh (B16CS018)
N.Mourya Mithra (B16CS019)
Qazi Sajid Azam (B16CS026)
Sai Kishore (B16CS028)
Sanchit Taliyan (B16CS031)
Anurag Shah (B16CS034)
Srijan Agarwal (B16EE036)
Vishakh S (B16CS038)
Zaid Khan (B16CS040)

1. Introduction and problem statement

The goal is to implement a reinforcement learning algorithm based on a partially observable Markov decision process. The agent is presented with a two-alternative decision task. Over a large number of trials, the agent must be able to correctly choose and perform one of the two alternatives based on an input stimulus. Here, the stimulus varies in value from -0.5 to 0.5. When the stimulus is less than 0, the agent must choose the Left option in order to make a correct decision. When it is more than 0, the agent must choose Right. When the stimulus is 0, the agent chooses between Left and Right randomly. Over the trials, the agent must learn to choose the right action for a stimulus. A reinforcement/ reward is given to the agent for each correct action chosen.

2. Underlying assumptions

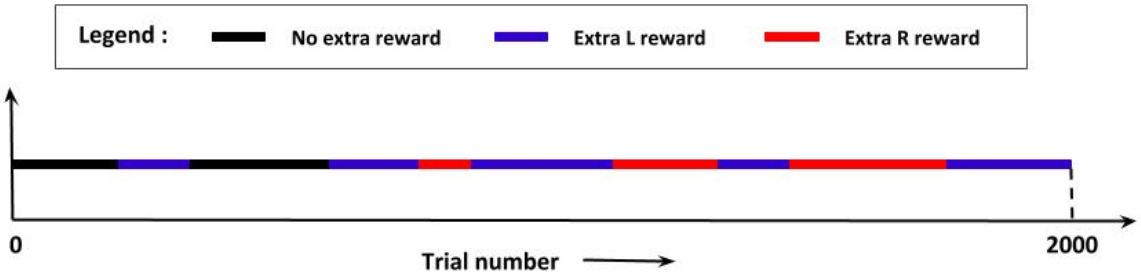
The problem solved here is a much-simplified version of the one implemented in the paper. In the experiments mentioned in the paper, the stimulus for the agent is provided inside a video frame. The agent has to first analyze the video frame to decipher the stimulus, before taking any action. However, the computational power required for processing video frames is very high. To overcome the resource bottleneck we faced, the stimulus in this experiment is assumed to be a decimal value sampled at random from the space [-0.5, 0.5].

Further, the set of possible actions is similar to that in the robot gesture control experiment, but limited to a size of 2 .i.e. ***Left*** and ***Right***. In the robot gesture control experiment, there were 4 possible actions - ***go left***, ***go right***, ***stop,*** and ***go forward***. This has also been done to reduce the computations needed.

3. Agent design

The agent must choose one of the 2 actions - ***Left*** and ***Right*** at each trial, based on a perceived stimulus. The agent receives a reward for a correct action to encourage it to go for the correct actions. The agent is rewarded in an asymmetric fashion. .i.e. The entire trial space is divided into several sections with different rewards corresponding to the actions taken. This division of the trial space has been performed at random.

In this experiment, 3 such sections are considered - ‘no extra reward’, ‘extra L reward’, and ‘extra R reward’. One such sample division for 2000 trials (.i.e. trial space [0, 2000]) is depicted below:



If the trial falls within the 'extra L reward' section, the agent receives a bonus over the usual reward if the expected outcome is Left and the agent guesses it correctly. Similarly, in the 'extra R reward' section, the agent receives a bonus over the usual reward if the expected outcome is Right and the agent guesses it correctly. No bonus is given during the trials in the 'no extra reward' section and the section with an extra reward for the opposite action. If the action taken by the agent does not match the actual choice (wrong prediction), then zero rewards are given to the agent, irrespective of the block the trial belongs to. The following table summarizes the reinforcement system:

Stimulus	Action	Reward block	Reward
Less than 0 (correct decision - Left)	Left	Extra L reward	1 + extra
		Extra R reward	1
		No extra reward	1
	Right	Any	0
More than 0 (correct decision - Right)	Left	Any	0
	Right	Extra L reward	1
		Extra R reward	1 + extra
		No extra reward	1
0 (correct decision randomly assigned to be Left/ Right)	Left	Extra L reward	1 + extra
		Extra R reward	1
		No extra reward	1
	Right	Extra L reward	1
		Extra R reward	1 + extra
		No extra reward	1

The model that is used here is a **P**artially **O**bservable **M**arkov **D**ecision **P**rocess, abbreviated as POMDP. The POMDP describes the effects of an agent's actions, the utilities of each state, and the relationships between those states. The model allows the agent to predict the long-term effects of its actions and take actions based on the predictions.

4. Data collection

We were directing our efforts towards obtaining a facial expression dataset (link). Due to a resource bottleneck that arose as a result of the suspension of academic activities owing to COVID-19, our progress was stalled and we couldn't proceed with a video or image dataset further. This is primarily due to the lack of adequate computational resources. Hence, we have provided the stimulus directly as an array with random values in the range [-0.5, 0.5].

5. Implementation and setting of parameters

The model implements a POMDP using the parameter Q. The Q-values signify the agent's value of choosing any particular action. In a way, these model the value function involved in the POMDP equation. So higher the Q-value, the higher is the value given by the agent to any particular action.

Steps taken are as follows :

- At the beginning of each trial, the agent receives a stimulus, denoted as **s**. How clearly the agent is able to perceive the stimulus is dependent on the absolute value of **s**.
- In order for obtaining generalization, the stimulus is mixed with some noise to model the imperfect perception of the stimulus. As for the implementation, the perceived stimulus is sampled from a normal distribution with mean (**μ**) = **s** (stimulus) and some standard deviation (**σ**). Here, **σ** is a parameter.
- Using this noisy stimulus, the agent forms a belief as to which side the stimulus corresponds to. The agent then computes the cumulative probability of the variable to find the probability of the stimulus being on a particular side.
- The agent then goes on to combine the belief about the current state with the values of ***Q*** already known.
- The computed belief is used to take a particular ***action A***. The agent receives a reward for its action, based on :
    - the correctness of the executed action and

- the current reward block .i.e.
  - extra L reward
  - extra R reward or
  - no extra reward

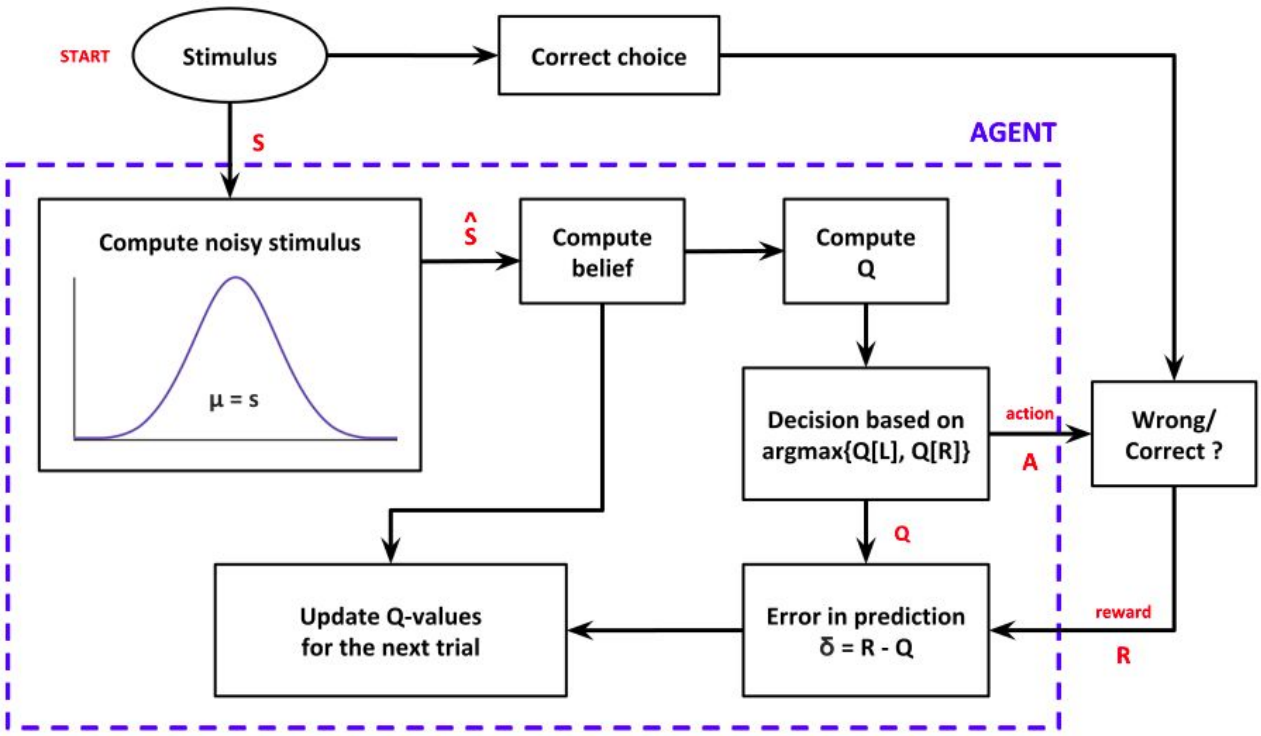
  The reward is computed as per the table given in section 3. The variable ***extra*** is a parameter of the model.

- The ***error δ*** in prediction is defined as ***reward R*** - ***Q***-value for the action taken by the agent in the current iteration.
- The ***error δ***, the agent's ***belief,*** and the agent's learning rate (**α**, a parameter) are used to update ***Q*** to obtain the ***Q***-value for the upcoming trial.

The model has three parameters, namely:

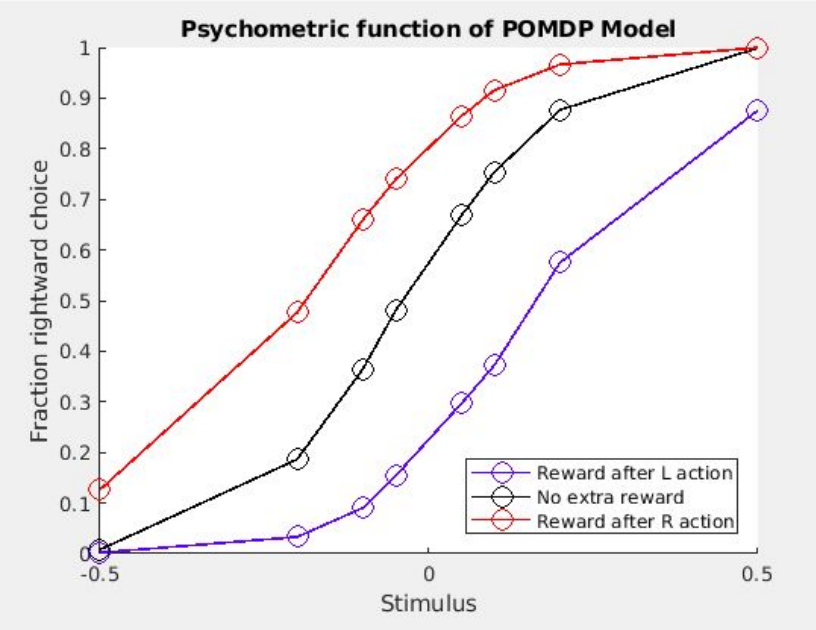
- **α**, the learning rate
- ***e***, the extra reward
- **σ**, the noise added in the stimulus (the standard deviation of the normal distribution)

The steps taken in each trial are summarized in the flowchart below:



6. Results

We have plotted a psychometric function that models the relationship between the stimulus perceived and the fraction of the choices taken by the agent that were correct.



The action-reward graph is obtained as: