**Machine Learning (CS60050) – Weekly Report**

**Kaushal Banthia (19CS10039)**

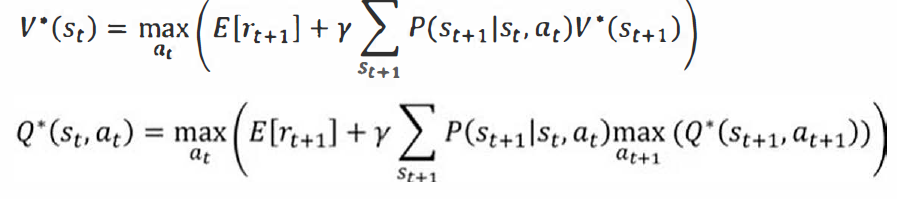
**Week 12: 3rd – 5th November, 2021**

**Topics Covered:**

* Learning with a critic
* K-Arm Bandit
* Markov Decision Process (MDP)
* Model Based Learning
* Temporal difference algorithm
* Deterministic and Non-Deterministic Environment

**Summary (Topic Wise):**

* Learning with a critic
* A critic is not a teacher as it does not tell us what to do. It only evaluates past performances. Also, the feedback is scarce and delayed
* The learner (Agent) interacts with an environment (at some state), which may change its state. It gets a reward / penalty sometimes (by the critic) for the steps it takes. In the end, it tries to reach a goal (a state).
* The learner learns a series of actions to reach the goal. It does this by maximizing the total rewards from any state.
* K-Arm Bandit
* Simple Learner:
  + There are K-levers in a simple machine with one state, where is the value of action . Initially . Store after each
  + The action is to pull a lever to win a reward ().
  + The task is to decide which lever to pull to maximize the reward.
  + Supervised Learning works to give the correct class, namely, the lever leading to maximum earning labeled by a teacher.
  + Reinforcement learning works by trying 0ut different levers and keeping track of the best.
  + Choose
* Non deterministic:
  + Here, the action is to pull a lever to win a reward with probability
  + The task is to decide which lever to pull to maximize the reward.
  + Reinforcement learning tries different levers and keeps track of the best.
  + : Estimate of the Value of action at time an average of all the past rewards when was chosen.
  + Delta Rule: , where is the learning factor that decreases with time.
  + Choose
* In a more complex environment, there are multiple states and the action also affects the next state. The agent senses state after an action () and may get a reward (). The non-deterministic reward is and we have to learn , which is the value of taking action in state . The rewards maybe delayed and thus, there is a need for the immediate estimation of a prospective reward.
* Markov Decision Process (MDP)
* An agent takes an action at state at discrete time
* Its goal is reaching a terminal state and then it remains there for any action with probability without any reward.
* The sequence of actions from the start to the terminal state is called an episode or a trial, while the mapping from the states of the environment to actions is called a policy.
* Value of a policy
* Discounted value with infinite horizon is
* Optimal policy
* As an alternative to this, we can learn , which is value at state with action.
* Optimal Policy is



Choose , providing or Choose , if

* Model Based Learning
* The models: are known.
* Directly solve for the optimal value function and policy using dynamic programming. There are two approaches for this, namely, Value iteration algorithm and policy iteration algorithm.
  + Value Iteration Algorithm:
    1. Initialize V(s) to arbitrary values
    2. Repeat
    3. For all s S
    4. For all a A


    8. Until V(s) converge
  + Policy Iteration Algorithm (Update policy directly from intermediate values):
    1. Initialize to arbitrary values
    2. Repeat
    4. Compute value using
    6. Improve the policy at each stage
    8. Until
* Temporal difference algorithm
* No prior knowledge of model. No required.
* It requires exploration of the environment to query the model to see the value of the next state and reward.
* Uses this information to update the value of the current state
* It is called as temporal difference algorithm, as it examines the difference between the current estimate of the value and the discounted value of the next state and the reward received.
* Deterministic and Non-Deterministic Environment
* Deterministic Environment:
  + For any state-action power, a single reward and state transition are possible.
  + Update at every exploration by adding immediate reward with discounted estimate of the next state-action pair.
  + Later updates are more reliable
  + Converges when all the pairs are stable (little changes with iteration).
* Nondeterministic Environment:
  + Varying reward or next state for a state-action pair.
  + Keeps a running average of values (Q-Learning)
  + Choose next action randomly (-Greedy sampling)
  + Sample an action uniformly with a probability initially.
  + Actions providing higher values would have higher probability.
  + Softmax over Q-values done using a temperature variable (T) and at every iteration, T increases, favoring higher Q-values.
  + Large number of states and actions are not feasible through tabular search. Use of regression is done to predict Q-values, given the current value, the reward and the next state. This requires supervisory information or labels.

**Concepts Challenging to Comprehend:** None yet.

**Interesting and Exciting Concepts:** K-Arm Bandit and Temporal difference algorithm.

**Concepts not understood:** None yet.

**A novel idea:** Instead of just choosing the best lever for the K-Arm Bandit, we could also keep a track of the second-best lever, third-best lever, …, and so on, for the Bandit. This can be done by keeping note of the values for all the levers and then sorting them in descending order. This way, if the first-best output might not have been the one that we had desired (or due to any other such reasons), we can try out the second-best lever, third-best lever, … and so on. This gives us more flexibility and control over our model!