Machine, Data and Learning (CS7.301)

Spring 2022, IIIT Hyderabad
07 Apr, Thursday (Lecture 17)

Taught by Prof. Vikram Pudi

Markov Decision Processes (contd.)

Policies

There are various types of policies an agent can follow to take decisions. These include stationary vs. non-stationary and deterministic vs. randomised policies.

A stationary policy entails choosing the same action at every step, while non-stationary policies allow one to change decisions across epochs.
In deterministic policies, we choose the action given the state (a rule of the form $S_i \to A_i$) with a probability of 1. Randomised policies involve selecting from a set of rules $(S_i \to A_{i1}, S_i \to A_{i2}, \ldots)$ with associated probabilities.

The optimal MDP policy (denoted π^*) for an infinite-horizon agent is a stationary and deterministic policy, denoted by the symbol π^{SD}.

A policy prescribes an action for all states; there is no such contingency as *failure*. It maximises expected reward rather than reaching a goal state.

Value Iteration

Value iteration is a dynamic programming method to determine the utility of states. It uses the old utility of neighbour states given certain actions. It is defined as

$$U_{t+1}(I) = \max_A \left(R(I, A) + \sum_J P(J \mid I, A) \cdot U_t(J) \right),$$

where I is the state whose utility we are determining, R is the reward function and J is a neighbour of I.
We run this update function until the utilities become close enough across epochs, *i.e.*, they converge.

Markov Chains

Given a policy, we can obtain a Markov chain from an MDP. In a Markov chain, the next state is dependent only on the current state, and not on the action (*i.e.*, there is only one action). In other words, we use the assumption

$$P(S_{t+1} \mid S_t, S_{t-1}, \dots) = P(S_{t+1} \mid S_t).$$

Discounting

Discounting is a process that allows us to compare policies in processes which allow the agent to accumulate arbitrarily large rewards. It involves deprecating rewards at future timesteps.

More concretely, we define a factor γ such that $0 \leq \gamma < 1$. Then the reward at a future timestep i is discounted by multiplying with γ^i.

Thus γ controls the effect of future rewards on current decisions. A typical value for γ is 0.95.
It is incorporated into the value iteration algorithm by altering it as follows

$$U_{t+1}(I) = \underset{A}{\operatorname{argmax}} \left(R(I, A) + \gamma \sum_J P(J \mid I, A) \cdot U_t(J) \right)$$