Project Work Example

Caden Hewlett

2024-03-26

The *Bellman Equation* for the frequentist Q-learning update model is given by.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\Big[r_t + \gamma \operatorname*{argmax}_{a \in A}\big\{Q(s_{t+1}, a)\big\} - Q(s_t, a_t)\Big]$$

In the literature, a 4-tuple Markov Decision Process (MDP) $\langle S, A, p_t, p_r\rangle$ is considered; where S is the state set (assumed to be discrete) and A is the action set. Critically, 4-tuple model includes p_t, the probability that action a_t at time t actually sends you to the target s_{t+1}, where $(s_t \xrightarrow{a_t \in A} s_{t+1})$. This concept can be likened to the real-world example of training to investment given a portfolio; there is some probability p_t that investing in a given company $(a \in A)$ will actually result in an increased market share $(s_{t+1} \in S.)$ In a simplified pathfinding model, however, we assume that the transition probabilities given action a at time t is not stochastic. In other words,

$$\forall t \in \mathbb{Z}^+, \forall (s_i, s_j) \in S,\ \forall a \in A \text{ s.t. } (s_i \overset{a}{\to} s_j),\ \mathbb{P}(S_{t+1} = s_{t+1} \mid \{a_t, s_t\}) \equiv \underbrace{p_t(s \overset{a}{\to} t)}_{\text{from literature}} = 1$$

Further, in the four-tuple model there is p_r; similar to p_t, it is the probability that we receive reward r in the rewards set R given we arrive at state s_{t+1} after taking $(s_t \xrightarrow{a_t \in A} s_{t+1})$ at time t. In the investment example, this means that given the investment and increased market share, p_r is the probability of a given return on investment $r \in R \subseteq \mathbb{R}$. In the pathfinding example, there isn't need for p_r in the initial implementation (though this may be revisited eventually as Bayesian models may perform better on p_r.) We assume that the rewards set R is defined by a deterministic function $h(\cdot \mid s)$ or, more simply, that R is the realization of a random variable, but is itself non-random.

In effect, the preliminary implementation is a 2-tuple MDP $\langle S, A\rangle$, which is more in-line with "simplified" Q-Learning implementations

in a Bayesian sense

$$V^\star(s) = \operatorname*{argmax}_{a \in A}\big\{Q(s_{t+1}, a)\big\}$$