Actor Critic model

Mattis Levik Trygstad

1 Temporal Difference (TD) Learning

- - Variables
  - Update function {#sec:update}
  - Eligibility Traces
  - TD basic sequence of events (*tabular* version)
  -
- Markdown elements

1.1 Variables

- a - action
- s - state
- s' - successor state
- V(s) - state value
- r - reinforcement (reward)
- Q(s,a) - state-action pair (SAP)

1.2 Update function

If agent is in state s and executes action a, which produces state s' and incurs reinforcement r. The information is stored by updating V(s):

$$V(s) = V(s) + \alpha \cdot [r + \gamma \cdot V(s') - V(S)] \cdot e(s) \tag{1}$$

- α - learning rate
- γ - discounting factor (0.9 - 0.99)
- δ - [...] term is the Temporal Difference (TD)

Small negative reinforcement is applied to each step, large positive reinforcement is given for the action leading to a goal state.

1.3 Eligibility Traces

TD provides backup to all states after every move. Implemented as continuous-valued flags attached to each state s (or SAP). Indicates the elapsed time since s was last encountered during problem solving search. As this time increases, *the eligibility decreases*, indicating that s or (s,a) is *less deserving* of an update to V(S). Conversely, states with a high eligibility should be more impacted by the recent reinforcement (positive or negative).

$$e_t(s) = \begin{cases} \gamma\lambda e_{t-1}(s) & \text{if } s \neq s_t \\ 1 & \text{if } s = s_t \end{cases} \tag{2}$$

where

- s_t is the state encountered at state t
- γ is the discount factor
- λ is the *trace-decay*

$s = s_t$ at current time step, will decrease each time step afterwards.

1.4 TD basic sequence of events (*tabular* version)

1. a $\leftarrow$ the action dictated by the current policy when the state is s, $\Pi(s)$
2. Performing action a from state s moves the system to state s' and achieves the immediate reinforcement r
3. $\delta \leftarrow r + \gamma V(s') - V(s)$
4. $e(s) \leftarrow 1$ (using the eligibility update function)
5. $\forall s \in S$
   a. $V(s) \leftarrow V(s) + \alpha\delta e(s)$
   b. $e(s) \leftarrow \gamma\delta e(s)$

2 Actor-Critic Model

The *actor* module contains the policy $\Pi(s)$, while the *critic* manages the value function V(s) or Q(s,a). Many models, but focus on $TD(\lambda)$ and the ise of eligibility traces to update both $\Pi(s)$ and $V(s)$. $\Pi(s)$ represents the action recommended by the actor when the system is in state s, and $\Pi(s,a)$ denotes the actor's quantitative evaluation of the desirability of choosing action a when in state s. Thus $\Pi(s) = argmax_a \Pi(s,a)$.

An ϵ-greedy strategy makes a random choice of action with a probability of ϵ, and the greedy choice with a probability of $1 - \epsilon$. ϵ should decrease from early to late episodes ($0.5 \rightarrow 0.001$).

2.1 Algorithm

1. CRITIC: initialize V(s) with small random values
2. ACTOR: Initialize $\Pi(s,a) \rightarrow 0 \forall s, a$
3. Repeat for each episode:
   1. Reset eligibilities in actor and critic: $e(s,a) \leftarrow 0, e(s) \leftarrow 0 \forall s, a$
   2. Initialize $s \leftarrow s_{init}, a \leftarrow \Pi(s_{init})$
   3. Repeat for each step of the episode:
      1. Execute action a from state s, moving the system to state s' and receiving the reward r
      2. ACTOR: $a' \leftarrow \Pi(s')$ the action dictated by the current policy for state s'
      3. ACTOR: $e(s,a) \leftarrow 1$ the actor keeps SAP-based eligibilities
      4. CRITIC: $\delta \leftarrow r + \gamma V(s') - V(s)$
      5. CRITIC: $e(s) \leftarrow 1$ the critic needs state-based eligibilities
      6. $\forall (s,a) \in$ current episode:
         1. CRITIC: $V(s) \leftarrow V(s) + \alpha_c \delta e(s)$
         2. CRITIC: $e(s) \leftarrow \gamma\lambda e(s)$
         3. ACTOR: $\Pi(s,a) \leftarrow \Pi(s,a) + \alpha_a \delta e(s,a)$
         4. ACTOR: $e(s,a) \leftarrow \gamma\lambda e(s,a)$
      7. $s \leftarrow s'; a \leftarrow a'$
   4. Until s is an end state

Using a table critic, each state has a table entry corresponding to its evaluation, which gets modified via $V(s) \leftarrow V(s) + \alpha_c \delta e(s)$.

However, when the critic uses an function approximator (F) instead of a table, no unique location within the neural network corresponds to a particular problem-solving state s or its value V(s). We wish to tune F such that, when presented with s as input, it produces a realistic V(s) as output.

3 Markdown elements

- ☒ test
- ☐ test
- ☒ todo

Table 1: Caption

col	col
1	2

ref. sec. 1.2

ref. tbl. 1