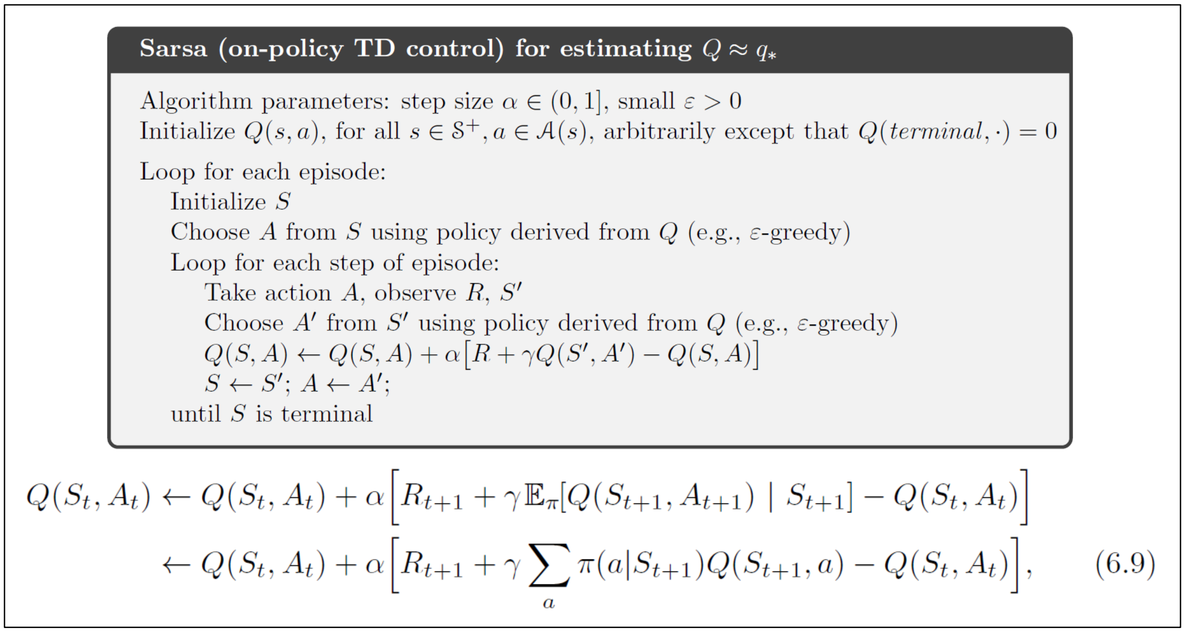
Two ideas for the project

Using SARSA or Expected SARSA (see below), we can implement a (coordination) game or a bidder.



Sarsa (on-policy TD control) for estimating $Q \approx q_*$

```
Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
   Initialize $S$
   Choose $A$ from $S$ using policy derived from $Q$ (e.g., $\varepsilon$-greedy)
   Loop for each step of episode:
      Take action $A$, observe $R$, $S'$
      Choose $A'$ from $S'$ using policy derived from $Q$ (e.g., $\varepsilon$-greedy)
      $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma Q(S', A') - Q(S, A)\big]$
      $S \leftarrow S'$; $A \leftarrow A'$;
   until $S$ is terminal
```

$$
\begin{aligned}
Q(S_t, A_t) &\leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \mathbb{E}_\pi[Q(S_{t+1}, A_{t+1}) \mid S_{t+1}] - Q(S_t, A_t)\Big] \\
&\leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) Q(S_{t+1}, a) - Q(S_t, A_t)\Big], \qquad (6.9)
\end{aligned}
$$

Coordination game

It consists of two agents (algorithms) moving simultaneously on a grid (up, down, left, right). For example: an agent is in the cell (3, 2) and chooses to go up, moving to the cell (2, 2).

In this game, an episode ends when they reach a terminal state, which happens when both are in the same cell of the grid. For example: agent A is in (3, 2) and chooses to go up, and agent B is in (2, 3) and chooses to go left.

They get a reward/punishment, depending on what we want them to learn. If they should learn to meet/avoid each other, they get a punishment/reward at each step they do not meet each other.

The avoidance game might go forever (e.g., they do not move out of two separated regions), so it should have a time discount.

For this game, a definition of the state each agent knows is the pair of cells they are.[1] For example: agent A is in (3, 2) and agent B is in (2, 3), so both know they are in the state S=S_A=S_B=((3, 2), (2, 3)).

[1] Another alternative is each agent knows the cell it is in and the cell the other agent was before (a lagged state, starting with a pair of states compatible with the lag). For example: agent A was in (3, 2) and moved from there to (3, 3), and agent B was in (2, 3) and moved from there to (1, 3), so A knows S_A=((3, 3), (2, 3)) and B knows S_B=((3, 2), (1, 3)). We can play with other alternatives as, for example, each agent knows only its cell, i.e., a state for agent A is its cell only.

Different from the algorithm above, in this game they observe R and S' after both take their actions.

Bidder

It consists of two or more agents bidding at discrete independent private values auctions. The goal is to learn the bidding functions.

More information at

Itzhak Rasooly & Carlos Gavidia-Calderon, 2020. "**The importance of being discrete: on the inaccuracy of continuous approximations in auction theory**," Papers 2006.03016, arXiv.org, revised Jan 2021.