Apprentissage actif par différence temporelle

- Dans le cas de l'apprentissage TD, l'approche active vorace devrait aussi utiliser

$$\pi(s) = \underset{a}{\operatorname{argmax}} \sum_{s' \in S} P(s'|s,a)\, V(s')$$

- Par contre, l'approche TD passive n'estime pas $P(s'|s,a)$

- En mode actif, on pourrait apprendre $P(s'|s,a)$ en plus de $V(s')$

Hugo Larochelle et Froduald Kabanza

Apprentissage actif avec *Q-learning*

- Peut-on éviter l'apprentissage de $P(s'|s,a)$?
- Une alternative est d'apprendre une **fonction action-valeur** $Q(s,a)$
  - on n'apprend plus $V(s)$, soit l'espérance de la somme des renforcements à partir de s jusqu'à la fin pour la politique optimale
  - on apprend plutôt $Q(s,a)$, soit l'espérance de la somme des renforcements à partir de s **et l'exécution de *a***, jusqu'à la fin pour la politique optimale
  - le lien entre $Q(s,a)$ et $V(s)$ est que $V(s) = \max_a Q(s,a)$

- Le plan de l'agent est alors $\pi(s) = \text{argmax}\ Q(s,a)$
  - plus besoin d'estimer $P(s'|s,a)$ et $V(s)$ séparément

- On appelle cette approche ***Q-learning***

Apprentissage actif avec *Q-learning*

- Selon la définition de $Q(s,a)$, on a

$$Q(s,a) = R(s) + \gamma \sum_{s' \in S} P(s'|s,a) \max_{a'} Q(s',a')$$

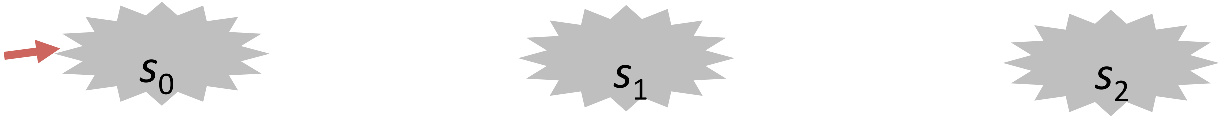
- Comme pour l'approche TD, on traduit cette équation en la mise à jour

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(R(s) + \gamma \max_{a'} Q(s',a') - Q(s,a) \right)$$

- On voit la similarité avec l'approche TD initiale

$$V(s) \leftarrow V(s) + \alpha \left(R(s) + \gamma V(s') - V(s) \right)$$

Apprentissage actif avec *Q-learning*

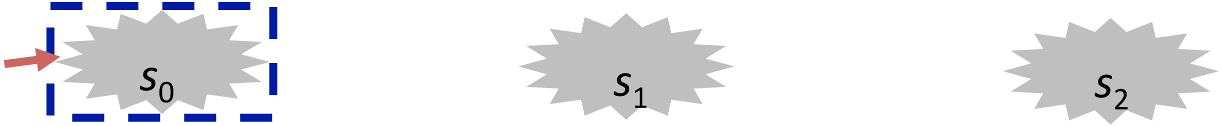


- Initialisation:

  $Q(s_0,a_1) = 0$ $\quad$ $Q(s_0,a_2) = 0$
  $Q(s_1,a_2) = 0$ $\quad$ $Q(s_1,a_3) = 0$
  $Q(s_2,\text{None}) = 0$

- On va utiliser $\alpha = 0.5$, $\gamma = 0.5$

Apprentissage actif avec *Q-learning*

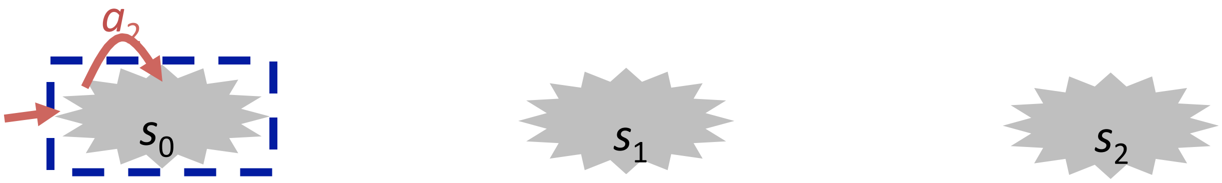


- Observations: (s_0) $_{-0.1}$

On ne fait rien (on a besoin d'un triplet (s, a, s'))

- Action à prendre $\pi(s_0)$ $= \text{argmax}\{\ Q(s_0,a_1), Q(s_0,a_2)\ \}$
  $= \text{argmax}\{\ 0, 0\ \}$
  $= a_2$ (arbitraire, ça aurait aussi pu être a_1)

Apprentissage actif avec *Q-learning*

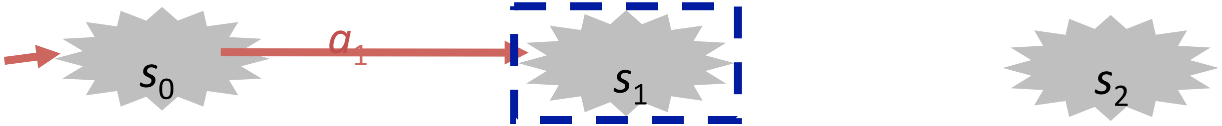


- Observations: $(s_0)_{\,-0.1} \xrightarrow{a_2} (s_0)_{\,-0.1}$

  $Q(s_0,a_2) \leftarrow Q(s_0,a_2) + \alpha\ (\ R(s_0) + \gamma \max\{\ Q(s_0,a_1), Q(s_0,a_2)\} - Q(s_0,a_2)\ )$
  $= 0 + 0.5\ (\ -0.1 + 0.5 \max\{\ 0, 0\} - 0)$
  $= -0.05$

- Action à prendre $\pi(s_0)$ $= \text{argmax}\{\ Q(s_0,a_1), Q(s_0,a_2)\ \}$
  $= \text{argmax}\{\ 0, -0.05\ \}$
  $= a_1$ **(changement de politique!)**

Apprentissage actif avec *Q-learning*

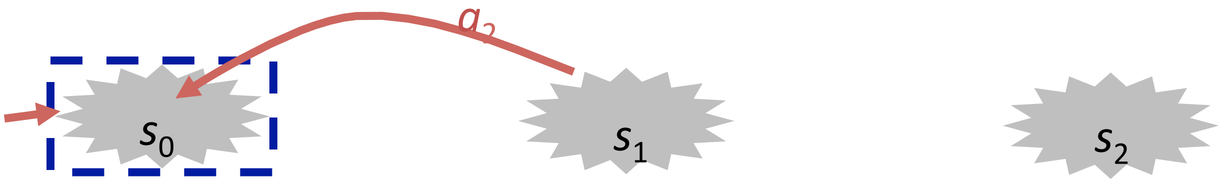


- Observations: $(s_0)_{-0.1} \xrightarrow{a_2} (s_0)_{-0.1} \xrightarrow{a_1} (s_1)_{-0.1}$

  $Q(s_0,a_1) \leftarrow Q(s_0,a_1) + \alpha\ (\ R(s_0) + \gamma \max\{\ Q(s_1,a_2),\ Q(s_1,a_3)\} - Q(s_0,a_1)\ )$
  $= 0 + 0.5\ (\ -0.1 + 0.5 \max\{\ 0,\ 0\} - 0)$
  $= -0.05$

- Action à prendre $\pi(s_1)$ $= \text{argmax}\{\ Q(s_1,a_2),\ Q(s_1,a_3)\ \}$
  $= \text{argmax}\{\ 0,\ 0\ \}$
  $= a_2$ (arbitraire, ça aurait aussi pu être a_3)

Apprentissage actif avec *Q-learning*

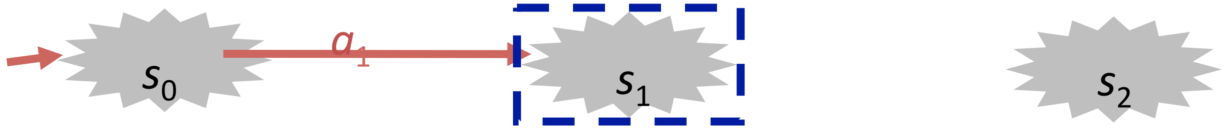


- Observations: $(s_0)_{\,-0.1} \xrightarrow{a_2} (s_0)_{\,-0.1} \xrightarrow{a_1} (s_1)_{\,-0.1} \xrightarrow{a_2} (s_0)_{\,-0.1}$

  $$Q(s_1,a_2) \leftarrow Q(s_1,a_2) + \alpha\,(\,R(s_1) + \gamma\,\max\{\,Q(s_0,a_1),\,Q(s_0,a_1)\} - Q(s_1,a_2)\,)$$
  $$= 0 + 0.5\,(\,-0.1 + 0.5\,\max\{\,-0.05,\,-0.05\} + 0)$$
  $$= -0.0625$$

- Action à prendre $\pi(s_0)$ $= \text{argmax}\{\,Q(s_0,a_1),\,Q(s_0,a_1)\,\}$
  $= \text{argmax}\{\,-0.05,\,-0.05\,\}$
  $= a_1$ (arbitraire, ça aurait aussi pu être a_2)

Apprentissage actif avec *Q-learning*

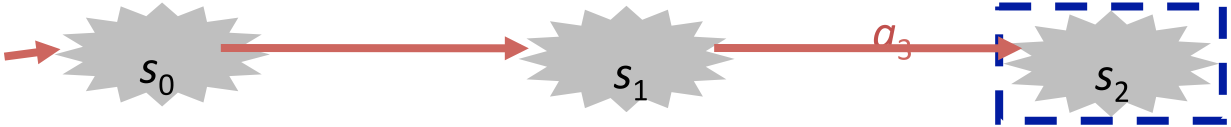


- Observations: $(s_0)\ _{-0.1} \xrightarrow{a_2} (s_0)\ _{-0.1} \xrightarrow{a_1} (s_1)\ _{-0.1} \xrightarrow{a_2} (s_0)\ _{-0.1} \xrightarrow{a_1} (s_1)\ _{-0.1}$

  $$Q(s_0,a_1) \leftarrow Q(s_0,a_1) + \alpha\ (\ R(s_0) + \gamma \max\{\ Q(s_1,a_2),\ Q(s_1,a_3)\} - Q(s_0,a_1)\ )$$
  $$= -0.05 + 0.5\ (\ -0.1 + 0.5 \max\{\ -0.0625,\ 0\ \} + 0.05)$$
  $$= -0.075$$

- Action à prendre $\pi(s_1)$ $= \text{argmax}\{\ Q(s_1,a_2),\ Q(s_1,a_3)\ \}$
  $= \text{argmax}\{\ -0.0625,\ 0\ \}$
  $= a_3$ **(changement de politique!)**

Apprentissage actif avec *Q-learning*



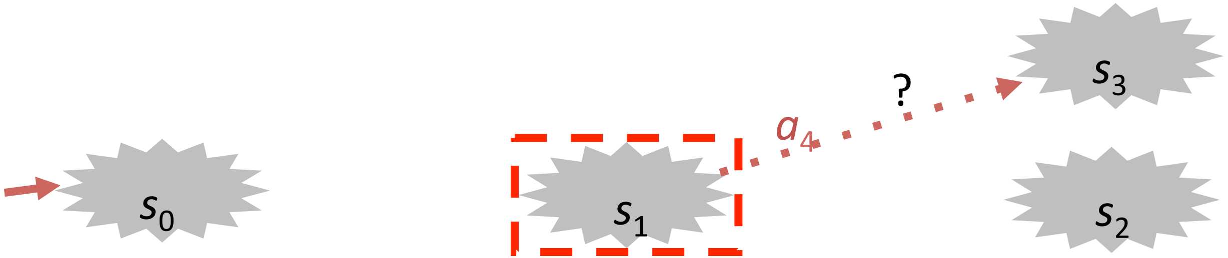
- Observations: $(s_0)_{-0.1} \xrightarrow{a_2} (s_0)_{-0.1} \xrightarrow{a_1} (s_1)_{-0.1} \xrightarrow{a_2} (s_0)_{-0.1} \xrightarrow{a_1} (s_1)_{-0.1} \xrightarrow{a_3} (s_2)_{1}$

  État terminal: $Q(s_2,\text{None}) = 1$

  $$\begin{aligned} Q(s_1,a_3) \leftarrow\ & Q(s_1,a_3) + \alpha\ (\ R(s_1) + \gamma \max\{\ Q(s_2,\text{None})\ \} - Q(s_1,a_3)\ ) \\ &= 0 + 0.5\ (\ -0.1 + 0.5 \max\{\ 1\ \} + 0) \\ &= 0.2 \end{aligned}$$

- On recommence un nouvel essai...

Apprentissage actif avec *Q-learning*



- Supposons qu'on puisse aussi faire l'action a_4 à l'état s_1, pour mener à s_3 tel que $R(s_3) = 1000$

- Puisque $Q(s_1,a_4) = 0$ à l'initialisation, et que $Q(s_1,a_3) > 0$ après un essai, une approche vorace n'explorera jamais s_3!

Apprentissage actif avec *Q-learning*

- On peut également contrôler la balance entre l'exploration et l'exploitation dans *Q-learning*

function Q-LEARNING-AGENT(*percept*) **returns** an action
  inputs: *percept*, a percept indicating the current state s' and reward signal r'
  persistent: Q, a table of action values indexed by state and action, initially zero
    N_{sa}, a table of frequencies for state–action pairs, initially zero
    s, a, r, the previous state, action, and reward, initially null

  if TERMINAL?(s) **then** $Q[s, None] \leftarrow r'$
  if s is not null **then**
    increment $N_{sa}[s, a]$
    $Q[s,a] \leftarrow Q[s,a] + \alpha(N_{sa}[s,a])(r + \gamma \max_{a'} Q[s',a'] - Q[s,a])$
  $s, a, r \leftarrow s', \underbrace{\operatorname{argmax}_{a'} f(Q[s',a'], N_{sa}[s',a'])}, r'$
  return a

la fonction d'exploration *f* contrôle la balance via N_e