# IIC3675: Tarea 3

Bruno Cerda Mardini

## a)

**Q-learning** es *off-policy* ya que en su función de actualización, el valor futuro de la state-action value function ($Q(S', A')$) se elige de manera *greedy*. En específico, se elige la acción que maximiza dicho valor, como se ve en la fórmula:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[ R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right]$$

**SARSA** es *on-policy* ya que, a diferencia de Q-learning, el valor futuro de la state-action value function ($Q(S', A')$) se determina eligiendo la siguiente acción según la política que se está optimizando actualmente:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[ R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t) \right]$$

## b)

Si ambos algoritmos empiezan a elegir acciones de manera greedy, entonces se vuelven muy similares, pero de todas formas **no van a ser equivalentes**.

Esto debido al orden en que se toman las acciones. En **SARSA**, se elige una acción ($A'$) para el siguiente estado ($S'$), luego se hace la actualización, y después esta misma acción ($A'$) se utiliza en el siguiente paso. En cambio, **Q-learning** NO elige una acción ($A'$) y se hace la actualización basado en la acción que simplemente maximice el valor de Q ($\max_a Q(S', a)$), de esta manera Q-learning no se compromete con ninguna acción, luego en el siguiente paso Q-learning es capaz de elegir una acción pero con la tabla Q ya actualizada.

Un ejemplo claro en donde ambos algoritmos podrían tomar decisiones distintas es en caso de empate de valores, debido a que la política greedy no podría elegir un claro ganador, es una situación en donde ambos algoritmos podrían diverger en resultados y la toma de acciones, el ejemplo es el siguiente:

Supongamos que un agente se encuentra en el estado $S$, toma la acción $A$ y debido a esto ahora se encuentra en el estado $S'$. Supongamos también que en la tabla Q, el valor para el estado $S'$ tiene un empate:

- $Q(S', A_1) = -1$

- $Q(S', A_2) = -1$

Debido a que hay un empate, la política greedy no puede elegir una acción que sea claramente mejor, por lo que lo tiene que hacer de manera uniforme. **SARSA** va a actualizar $Q(S, A)$, para esto va a elegir la siguiente acción $A'$, pero como hay un empate, supongamos que va a terminar eligiendo $A_1$, por lo que en el siguiente paso va a estar obligado a ejecutar $A_1$. **Q-Learning** va a actualizar $Q(S, A)$ usando la acción que maximice el valor de retorno, pero como hay un empate, supongamos que va a tomar la acción $A_2$ como la acción que maximice el valor de $Q(S', a)$ **(Pero no la guarda para el siguiente paso)**. La actualización para $Q(S, A)$ de ambos algoritmos va a ser la misma, pero **eligieron acciones distintas**, por lo que no son equivalentes.

## c)

En la **Figura 1** podemos observar que SARSA y 4-step SARSA son mejores que Q-learning. En específico, en los primeros episodios, 4-step SARSA obtiene retornos mejores que SARSA y Q-learning, pero después SARSA converge al mismo valor que 4-step SARSA en términos de retorno promedio para los últimos episodios, mientras que Q-learning se mantiene claramente más abajo con peores valores. La principal razón por la cual obtenemos estos resultados es debido a que SARSA y 4-step SARSA son **on-policy**, y Q-learning es **off-policy**.

**Q-learning** va a explorar de manera $\epsilon$-*greedy*, pero va a actualizar sus valores $Q(S, A)$ de manera *greedy*. Debido a esto, el algoritmo va a aprender a tomar el camino más corto, pero al mismo tiempo el más riesgoso, ya que con probabilidad $\epsilon$ se va a caer al *cliff*, donde va a obtener una recompensa de -100. A diferencia de Q-learning, **SARSA** y **4-step SARSA** son **on-policy**, por lo que van a explorar de una manera $\epsilon$-*greedy*, y también actualizarán sus valores $Q(S, A)$ en base a esa misma política. Lo anterior implica que, al explorar los estados cercanos al *cliff*, estos se actualizarán con malos valores $Q(S, A)$, ya que con probabilidad $\epsilon$ el agente se va a caer. Es por esto que el agente va a terminar aprendiendo un camino más largo y seguro.

Finalmente, la razón por la cual 4-step SARSA obtiene mejores retornos promedio de manera mucho más rápida que SARSA es debido a que las actualizaciones de $Q(S, A)$ consideran 4 pasos más adelante, por lo que logra aprender de manera más rápida que es mejor alejarse del *cliff* para evitar la recompensa de -100. Considerando todo lo anterior, el resultado que se puede observar en la figura tiene mucho sentido.

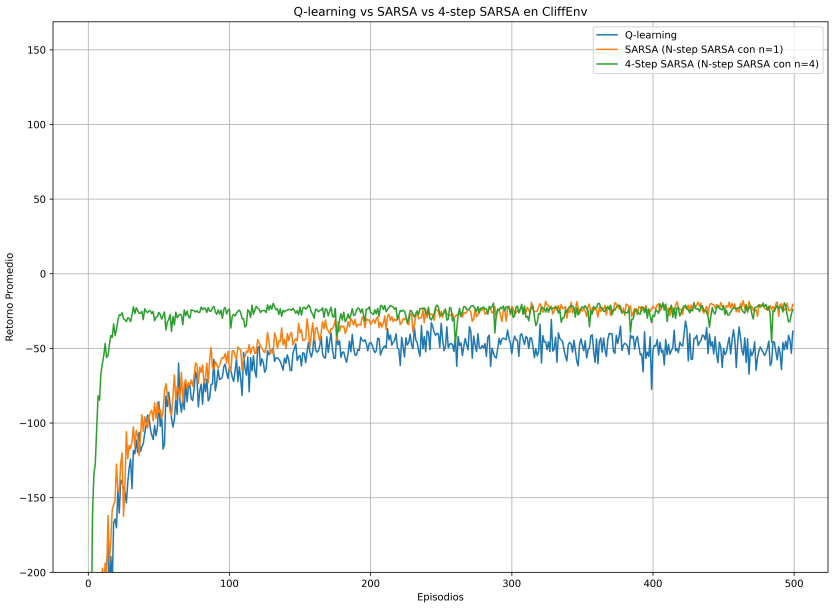


Figure 1: Comparación de retorno promedio entre Q-learning, SARSA y 4-step SARSA en Cliff Walking.