

Comparativa entre Sarsa y Expected-Sarsa en el entorno MountainCar-v0.

Para poder medir la eficiencia de los algoritmos en este entorno, tenemos que tener en cuenta que este nos provee de una observación o estado compuesta por variables continuas, es decir, debemos discretizarlas antes de poder comenzar el entrenamiento.

Para la comparación se entrenó el agente 10.000 episodios por cada α para obtener el promedio de las recompensas en cada caso y cual nos da mejores resultados. Para ϵ y gamma se utilizan los valores de 0.1 y 0.95 respectivamente para que el agente puede encontrar rápidamente la mejor política y optimizar su desempeño.

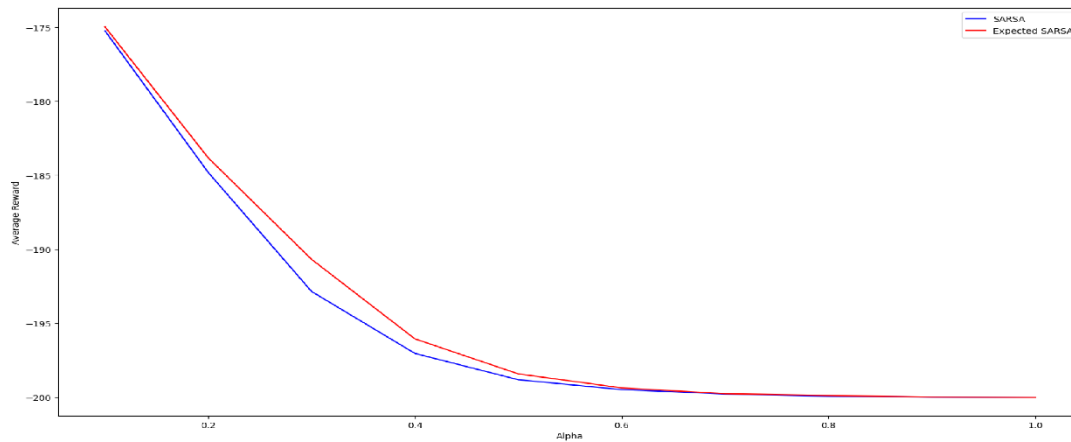


Figura 1: Sarsa and Expected SARSA en el entorno MountainCar-v0.

En la Figura 1 se puede observar el rendimiento de los 2 algoritmos, siendo así el algoritmo Expected SARSA el que consigue mejores resultados, ya que es el que consigue una recompensa más alta, a su vez, también es el que logra aprender de una forma más rápida.

Avg/ α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Sarsa	-175.1708	-184.7353	-192.0165	-196.3793	-198.6947	-199.4459	-199.8264	-199.8974	-199.9726
E-Sarsa	-174.593	-183.6123	-191.1573	-195.4393	-198.3176	-199.1569	-199.6301	-199.8102	-199.9693

Tabla1: Avg. de Sarsa and Expected SARSA para cada en α el entorno MountainCar-v0.

En la tabla 1 se observa los resultados obtenidos, y se ve como el algoritmo que con cada valor de α consigue resolver en el menor número de pasos y, por ende, conseguir una recompensa más alta es el algoritmo Expected SARSA.

Natanael Rojo C.I: 26.488.388

Heberto Gutierrez C.I: 24.752.816

Github: [computer-systems-activities/homework-10 at main · NatanaelRojo/computer-systems-activities \(github.com\)](https://github.com/NatanaelRojo/computer-systems-activities/tree/main/homework-10)