**Universidad del Valle de Guatemala**
**Facultad de ingeniería**


# Laboratorio # 5: RL

**Cayetano Molina 20211**
**Estefanía Elvira 20725**
**Priscilla Gonzalez 20689**

**Guatemala 13 de agosto del 2024**

1. **¿Qué es “Expected SARSA”?**

   **Expected SARSA** es una variación del algoritmo SARSA, que actualiza el valor de una acción en un estado específico basado en la recompensa recibida y el valor esperado de las acciones futuras. En lugar de usar la recompensa observada directamente, Expected SARSA calcula la expectativa de las recompensas futuras considerando todas las posibles acciones desde el próximo estado, ponderadas por la política actual .

   a. **Diferencia con SARSA:** La principal diferencia entre SARSA y Expected SARSA radica en cómo se actualizan los valores de las acciones:

      - **SARSA** (State-Action-Reward-State-Action) usa la recompensa recibida y la acción realmente tomada en el siguiente estado para actualizar el valor de la acción. Es un método on-policy, es decir, sigue la política actual tanto para la acción como para la evaluación .
      - **Expected SARSA** también es on-policy, pero en lugar de usar la acción realmente tomada, utiliza la expectativa sobre todas las posibles acciones en el siguiente estado según la política actual para la actualización .

   b. **Utilidad de las modificaciones en SARSA:** Expected SARSA reduce la variabilidad en las actualizaciones del valor de la acción al considerar todas las acciones posibles en el siguiente estado, lo que conduce a una convergencia más estable y rápida, menos influenciada por la aleatoriedad .

**2. Qué es “n-step TD”?**

**n-step TD** (n-step Temporal Difference) es una extensión del método TD en la cual las actualizaciones se basan en la recompensa recibida después de n pasos, en lugar de uno solo. Este método combina elementos de TD(0) y de Monte Carlo, siendo una forma intermedia entre ambos .

**a. Diferencia con TD(0):**

- **TD(0)** actualiza el valor de un estado basado en la recompensa inmediata y el valor del siguiente estado.

- **n-step TD** utiliza una secuencia de n recompensas y la estimación del valor del estado después de esos n pasos, lo que proporciona una representación más robusta de las recompensas futuras .

b. **Utilidad de esta modificación:** El **n-step TD** permite un equilibrio entre la precisión y la variabilidad de las actualizaciones. Un valor mayor de n reduce la variabilidad, similar al enfoque Monte Carlo, pero introduce un retardo en las actualizaciones. Esto puede llevar a un aprendizaje más eficiente, especialmente cuando n se elige adecuadamente para el entorno específico .

c. **Objetivo utilizado:** El objetivo en **n-step TD** es el valor estimado del retorno total esperado después de n pasos, sumado a la estimación del valor del estado futuro después de esos n pasos. Esto mejora la política de decisión del agente .

**3. Diferencia entre SARSA y Q-learning**

**SARSA** y **Q-learning** son algoritmos de aprendizaje por refuerzo que utilizan el enfoque de Q-learning, pero difieren en cómo actualizan los valores de las acciones:

- **SARSA**: Es un algoritmo on-policy que actualiza los valores de acción usando la política actual. La actualización se basa en la acción realmente tomada en el siguiente estado, lo que hace que SARSA sea más conservador y dependiente de la política actual .

- **Q-learning**: Es un algoritmo off-policy que utiliza la mejor acción posible en el siguiente estado para la actualización, sin importar la política que esté siguiendo el agente. Esto hace que Q-learning sea más eficiente en la búsqueda de la política óptima, aunque también puede ser más propenso a decisiones arriesgadas .

**Conclusión**:

- **SARSA** sigue la política actual, lo que puede llevar a un comportamiento más seguro pero menos óptimo.
- **Q-learning** busca la política óptima independientemente de la política actual, lo que puede ser más eficiente pero potencialmente más arriesgado .

**Referencias:**


1. Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.
2. Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3-4), 279-292.
3. Szepesvári, C. (2010). *Algorithms for Reinforcement Learning*. Morgan & Claypool Publishers.
4. Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1), 9-44.