UNIVERSIDAD DEL VALLE DE GUATEMALA
Reinforcement Learning
Sección 20
JAVIER JOSUÉ FONG GUZMÁN

Laboratorio #1

Gustavo González - 21438
María Marta Ramirez – 21342

Guatemala 15 de julio, 2025

Descripción general de la implementación

El agente se implementó utilizando el algoritmo **Policy Iteration**, compuesto por dos fases principales: **policyIterate** y **policyImprove**.

- **policyIterate**
  - Inicializa una política aleatoria para cada estado no terminal.
  - Evalúa la política actual calculando la función de valor (V(s)) de cada estado mediante iteraciones sucesivas hasta que la diferencia entre iteraciones sea menor que un umbral ((\theta)), indicando convergencia.
  - Llama repetidamente a **policyImprove** para actualizar la política y verificar si es estable.
- **policyImprove**
  - Para cada estado, calcula el valor esperado de ejecutar cada acción posible considerando las transiciones y recompensas definidas por el entorno.
  - Selecciona la acción que maximiza dicho valor esperado.
  - Si en algún estado la acción óptima difiere de la acción actual de la política, marca la política como **no estable** para continuar iterando.

Ambas funciones trabajan de forma iterativa hasta que la política converge, es decir, no se producen cambios en ninguna acción para todos los estados.

Resultados obtenidos

Durante la ejecución, el agente comenzó con una política aleatoria y, a través de cuatro iteraciones principales, alcanzó una política estable.
En cada iteración, se registraron:

- **Número de iteraciones internas de evaluación**:
  - Iteración 1: 52
  - Iteración 2: 104
  - Iteración 3: 3
  - Iteración 4: 2
- **Evolución de la función de valor (V(s))**:
  - Inicialmente, los valores eran bajos y negativos en varios estados, reflejando recompensas acumuladas pobres.
  - Posteriormente, los valores convergieron a cifras cercanas a **50** en estados favorables, lo que indica la maximización de recompensas a largo plazo.
- **Evolución de la política**:
  - Comenzó con direcciones dispersas y sin una estructura clara.
  - En la iteración final, la política muestra un camino consistente hacia los estados objetivo, evidenciando aprendizaje estable.

Análisis de convergencia

El algoritmo mostró un patrón típico de **Policy Iteration**:

- En las primeras iteraciones, la política cambia significativamente, lo que explica el alto número de evaluaciones internas.
- Conforme se acerca a la política óptima, el número de iteraciones internas disminuye drásticamente (de 104 en la iteración 2 a solo 2 en la iteración 4).
- La convergencia se logró en **4 iteraciones de mejora de política**, lo que confirma la eficiencia del método en entornos pequeños como GridWorld.

La política final es estable y garantiza la obtención de la mayor recompensa posible desde cualquier estado no terminal.

Observaciones y dificultades

- La implementación original estaba basada en gym, pero el entorno actual utilizaba gymnasium, lo que requirió modificar las importaciones para evitar conflictos.
- Fue necesario verificar que el bucle de evaluación de política terminara correctamente, ya que umbrales muy bajos pueden aumentar innecesariamente el tiempo de cómputo.
- Una dificultad inicial fue la interpretación de los valores de (V(s)) y su relación con las políticas intermedias, pero con la observación iterativa se confirmó que los valores altos coincidían con estados cercanos al objetivo.
- El resultado final demuestra que el algoritmo es robusto y eficiente para problemas de decisión de Markov finitos.

Conclusión:

La implementación de **Policy Iteration** permitió al agente encontrar una política óptima en pocas iteraciones, con una convergencia clara tanto en la función de valor como en las acciones. La principal dificultad fue la compatibilidad de librerías, la cual se resolvió migrando completamente a gymnasium.