Clase 163 — Markov Decision Processes

Parte: 2 — Deep Learning · Fuente: Géron, cap. 18 § Markov Decision Processes + Sutton & Barto cap. 3. Duración estimada: 60 min.

Clase 163 — Markov Decision Processes

> *Parte: 2 — Deep Learning · Fuente: Géron, cap. 18 § Markov Decision Processes + Sutton & Barto cap. 3. Duración estimada: 60 min.*

Objetivo

Formalizar el marco teórico de RL: un Markov Decision Process (MDP) = tupla (S, A, P, R, γ). Conocer la Bellman equation que define V y Q óptimos, y los algoritmos clásicos Value Iteration y Policy Iteration que los resuelven (cuando el MDP es conocido).

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Definir un MDP: states, actions, transition prob, reward function, discount.
- Escribir la Bellman equation para V: V(s) = max_a Σ P(s'|s,a)[R + γ V*(s')].
- Implementar Value Iteration: actualizar iterativamente hasta convergencia.
- Implementar Policy Iteration: alternar policy evaluation + policy improvement.
- Reconocer la limitación: requiere conocer P y R (no aplicable a entornos reales) → motivó model-free (Q-learning, clase 137).

Temas

- Propiedad de Markov: P(s_{t+1} | s_t, a_t) = P(s_{t+1} | s_t, a_t, s_{t-1}, ...).
- Componentes MDP.
- Bellman optimality equation.
- Value Iteration (synchronous update).
- Policy Iteration (alternar eval + improve).
- Convergencia garantizada (contractive operator).

Definiciones y características

- State s: representación del entorno.
- Action a: lo que el agente puede hacer.
- Transition P(s' | s, a): probabilidad de llegar a s' desde s tomando a.
- Reward R(s, a, s'): recompensa inmediata.
- Discount γ: peso de recompensas futuras.
- V*(s): máximo expected return desde s.
- Q*(s, a): máximo expected return tras tomar a en s.
- Policy óptima: π(s) = argmax_a Q(s, a).

Dataset / recursos

- MDPs sintéticos chicos (FrozenLake, Taxi de Gymnasium).
- Librerías: gymnasium, numpy.

Ejercicios

1. MDP de juguete: definir un MDP de 4 estados con P, R manualmente.
2. Value Iteration: implementar V[s] = max_a Σ P(s'|s,a)(R + γ V[s']) hasta max change < 1e-6.
3. Policy Iteration: alternar evaluación (V^π) con mejora (π' = greedy(V)) hasta estabilidad.
4. FrozenLake: cargar gym.make('FrozenLake-v1'), extraer env.unwrapped.P (modelo del MDP), resolver con VI.
5. Compare: # iteraciones VI vs PI para llegar a misma policy.

Homework verificable

Sobre FrozenLake-v1 (is_slippery=True):

1. Extraer modelo P y R desde env.unwrapped.P.
2. Implementar Value Iteration; reportar V y π greedy.
3. Evaluar la policy con 1000 episodios random; reportar success rate.

Criterio de aceptación: success rate ≥ 0.7 sobre FrozenLake slippery; V* y policy claramente prefieren caminos seguros.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
Value Iteration no converge	γ ≥ 1.0. Fix: γ < 1.0 (típicamente 0.95-0.
Policy Iteration loops infinito	Numerical issues en evaluación. Fix: limit
Acceder a env.P falla	Hay que usar env.unwrapped.P.
Asumir Markov en problemas non-Markov	Por ejemplo, control con velocidad escondi
VI vs PI: cuándo usar?	VI: más simple, más iters pero más baratas

Preguntas frecuentes

¿MDPs en problemas reales?

Casi nunca conocés P y R exactamente. Por eso → Q-learning (clase 137) y métodos model-free.

¿POMDPs?

Partially Observable: cuando no ves el state completo. Mucho más difícil. Usar agente con memoria (LSTM, Transformer).

¿Continuous state space?

VI/PI no escalan. Usar function approximation: DQN (clase 137) o policy gradient (135).

¿Bellman equation conexión con DP?

Sí, VI es dynamic programming clásico aplicado a MDPs.

¿Aprender el modelo P, R desde data?

Model-based RL: aprender un modelo, planificar con él. Más sample-efficient pero más complejo.

Referencias

- Géron, cap. 18 — Markov Decision Processes.
- Sutton & Barto, cap. 3-4.
- Bellman (1957), Dynamic Programming.

Siguiente clase

Clase 164 — TD Learning, Q-Learning, Deep Q-Networks

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb