INTELIGENCIA ARTIFICIAL PARA JUEGOS

SESIÓN 7

Dr. Edwin Villanueva Talavera

Contenido

- Definiciones
- Aprendizaje por Refuerzo Pasivo
  - Estimación Directa de la Utilidad
  - Programación Dinámica Adaptativa
  - Aprendizaje de Diferencias Temporales
- Aprendizaje por Refuerzo Activo
  - Q-learning

Bibliografía:

Capitulo 21.1, 21.2, 21.3 y 21.4 del libro:

Stuart Russell & Peter Norvig "Artificial Intelligence: A modern Approach", Prentice Hall, Third Edition, 2010

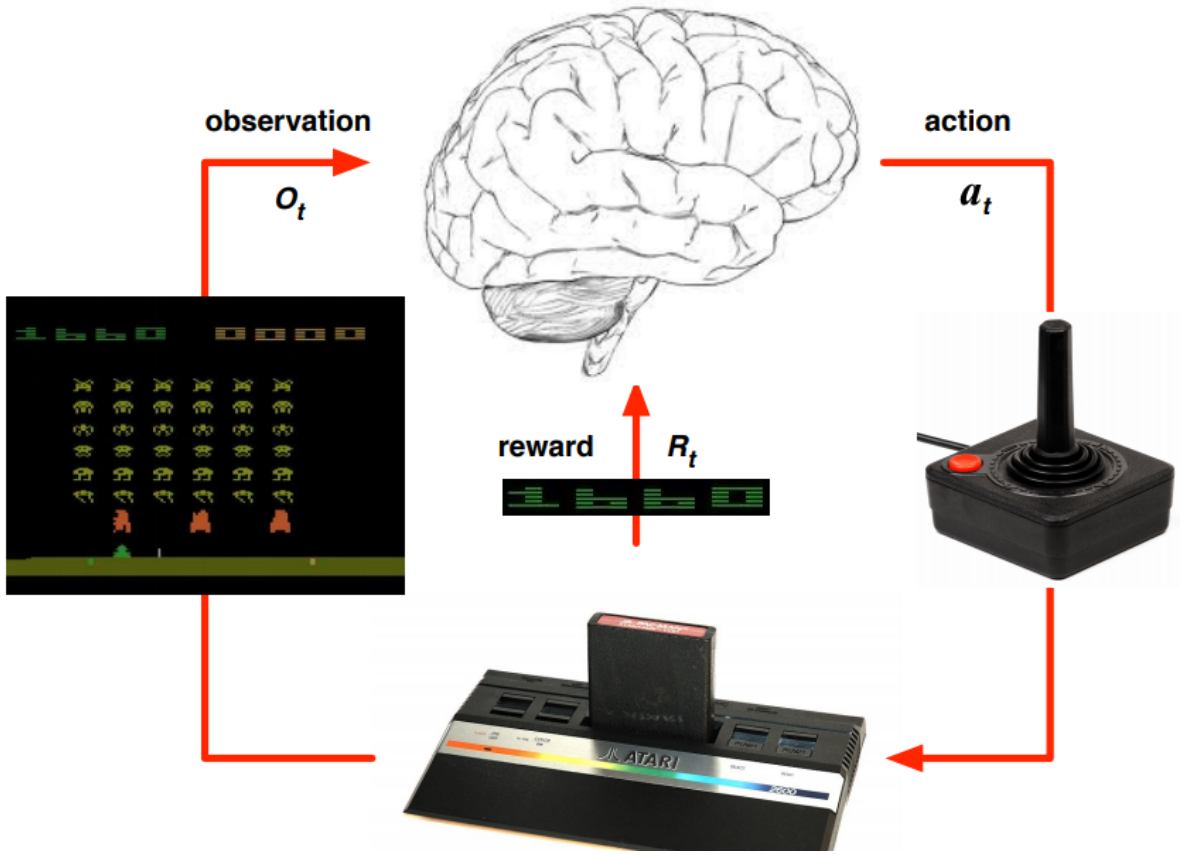
(*) imágenes tomadas de este libro

Definición de Aprendizaje por Refuerzo

- El agente actúa en un entorno PDM, especificado por:
  - un conjunto de estados $s \in S$
  - un conjunto de acciones $a \in A$
  - un estado inicial s_o
  - (Tal vez) uno o mas estados terminales
- Novedad: El agente no conoce el modelo de transición $P(s' \mid s, a)$ ni la función de recompensa R(s) (recibe ella cuando visita s)
  - El agente necesita ejecutar acciones y recibir recompensas para aprender
- Objetivo: Encontrar uma politica opima $\pi^*(s)$

Definición de Aprendizaje por Refuerzo

Ejemplo: Juegos de Atari



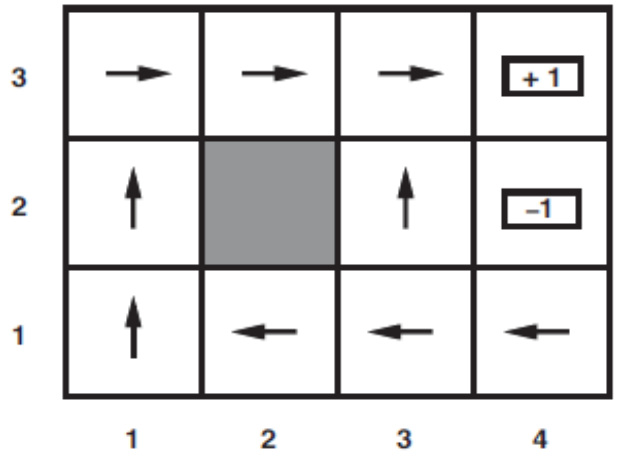
- Reglas del juego desconocidas. Se necesita aprender de la experimentación: mover el joystick y ver los pixels y el score resultante

Aprendizaje por Refuerzo Pasivo

- Tarea de aprendizaje simplificada:
  - El agente tiene una política fija π (en estado s ejecuta $\pi(s)$)
  - El agente no conoce el modelo de transición $P(s' \mid s, a)$
  - El objetivo del agente es aprender que tan buena es la política π, esto es, aprender la función de utilidad $U^{\pi}(s)$
- En esta tarea:
  - El agente no escoge acciones
  - Solamente ejecuta las acciones dictadas por la política y aprende de la experiencia su utilidad
- Métodos:
  - Estimativa directa de la utilidad
  - Programación dinámica adaptativa
  - Diferencias temporales

Aprendizaje por Refuerzo Pasivo

Ejemplo: El laberinto 4x3 *



- Tres posibles secuencias al experimentar con la política empezando en (1,1) :

$$(1,1)_{\textbf{-.04}} \rightsquigarrow (1,2)_{\textbf{-.04}} \rightsquigarrow (1,3)_{\textbf{-.04}} \rightsquigarrow (1,2)_{\textbf{-.04}} \rightsquigarrow (1,3)_{\textbf{-.04}} \rightsquigarrow (2,3)_{\textbf{-.04}} \rightsquigarrow (3,3)_{\textbf{-.04}} \rightsquigarrow (4,3)_{\textbf{+1}}$$
$$(1,1)_{\textbf{-.04}} \rightsquigarrow (1,2)_{\textbf{-.04}} \rightsquigarrow (1,3)_{\textbf{-.04}} \rightsquigarrow (2,3)_{\textbf{-.04}} \rightsquigarrow (3,3)_{\textbf{-.04}} \rightsquigarrow (3,2)_{\textbf{-.04}} \rightsquigarrow (3,3)_{\textbf{-.04}} \rightsquigarrow (4,3)_{\textbf{+1}}$$
$$(1,1)_{\textbf{-.04}} \rightsquigarrow (2,1)_{\textbf{-.04}} \rightsquigarrow (3,1)_{\textbf{-.04}} \rightsquigarrow (3,2)_{\textbf{-.04}} \rightsquigarrow (4,2)_{\textbf{-1}} \ .$$

¿Cómo aprender $U^{\pi}(s)$ con la información obtenida de diferentes experimentos?

Aprendizaje por Refuerzo Pasivo

Estimación Directa de la Utilidad

- Recordando. La utilidad de un estado es el valor esperado de la suma de recompensas (descontadas) obtenidas con la política π

$$U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(S_t)\right]$$

- La estimación directa de utilidad del estado s, $\hat{U}^{\pi}(s)$, es la media de utilidades de dicho estado (actualizada al fin de cada experimento). Para el ejemplo:
  - $\hat{U}^{\pi}(1,1)$ = (0.72 + 0.72 + (-1.16))/3 = 0.28/3
  - $\hat{U}^{\pi}(1,2)$ = ?

Aprendizaje por Refuerzo Pasivo

Programación Dinámica Adaptativa (ADP)

- Motivación: la convergencia de la Estimación Directa es lenta, ya que no considera las restricciones de las ecuaciones de Bellman:

$$U^{\pi}(s) = R(s) + \gamma \sum_{s'} P(s' \mid s, \pi(s)) U^{\pi}(s')$$

  - Ej. En el 2do intento, cuando se pasa de (3,2) a (3,3) ya se sabe que este último es de alta utilidad (del 1er intento). La eq. de Bellman diría que (3,2) también tiene alta utilidad, pero Estimación Directa no aprende nada hasta el final del intento.

- Idea: Con cada transición ejecutada de s a s' con acción a:
  - Actualizar el modelo de transición empíricamente $P(s' \mid s, a)$
  - Usar iteración de valor simplificada (POLICY-EVALUATION) para obtener las utilidades que resuelven el PDM con la política dada y el modelo de transición estimado hasta ahora

Aprendizaje por Refuerzo Pasivo

Programación Dinámica Adaptativa (ADP)

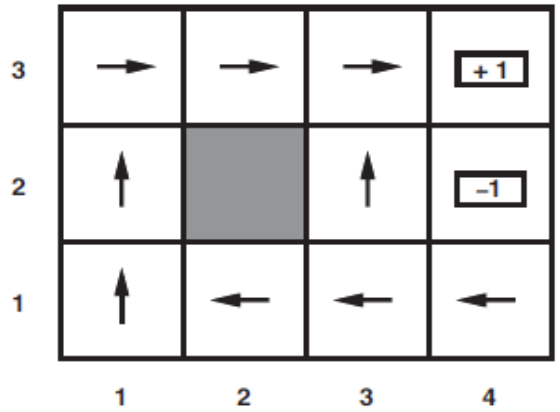
function PASSIVE-ADP-AGENT($percept$) **returns** an action
  inputs: $percept$, a percept indicating the current state s' and reward signal r'
  persistent: π, a fixed policy
    mdp, an MDP with model P, rewards R, discount γ
    U, a table of utilities, initially empty
    N_{sa}, a table of frequencies for state–action pairs, initially zero
    $N_{s'|sa}$, a table of outcome frequencies given state–action pairs, initially zero
    s, a, the previous state and action, initially null

  if s' is new **then** $U[s'] \leftarrow r'$; $R[s'] \leftarrow r'$
  if s is not null **then**
    increment $N_{sa}[s, a]$ and $N_{s'|sa}[s', s, a]$
    for each t such that $N_{s'|sa}[t, s, a]$ is nonzero **do**
      $P(t \mid s, a) \leftarrow N_{s'|sa}[t, s, a] \,/\, N_{sa}[s, a]$
  $U \leftarrow$ POLICY-EVALUATION(π, U, mdp)
  if s'.TERMINAL? **then** $s, a \leftarrow$ null **else** $s, a \leftarrow s', \pi[s']$
  return a

Aprendizaje por Refuerzo Pasivo

Programación Dinámica Adaptativa: Ejercicio

- Dada la política:



*

- Se ejecutó tres veces, resultando en los episodios:

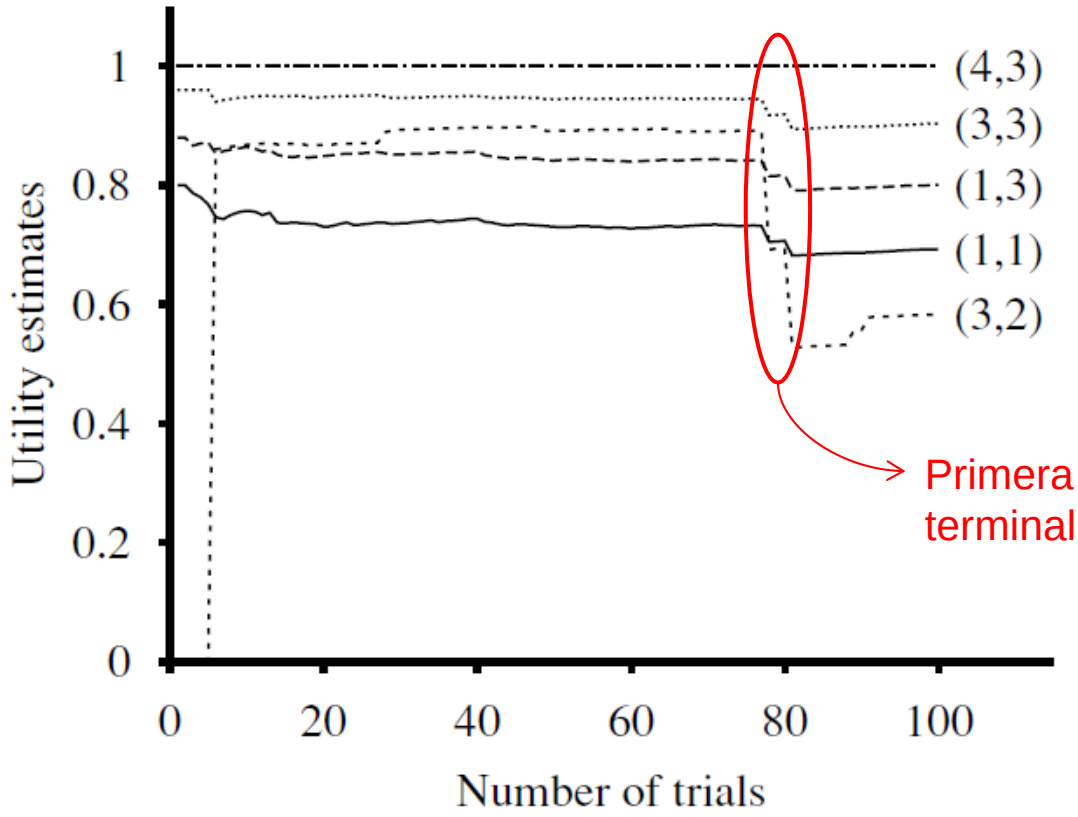
$$(1,1)_{\text{-.04}} \rightsquigarrow (1,2)_{\text{-.04}} \rightsquigarrow (1,3)_{\text{-.04}} \rightsquigarrow (1,2)_{\text{-.04}} \rightsquigarrow (1,3)_{\text{-.04}} \rightsquigarrow (2,3)_{\text{-.04}} \rightsquigarrow (3,3)_{\text{-.04}} \rightsquigarrow (4,3)_{\text{+1}}$$
$$(1,1)_{\text{-.04}} \rightsquigarrow (1,2)_{\text{-.04}} \rightsquigarrow (1,3)_{\text{-.04}} \rightsquigarrow (2,3)_{\text{-.04}} \rightsquigarrow (3,3)_{\text{-.04}} \rightsquigarrow (3,2)_{\text{-.04}} \rightsquigarrow (3,3)_{\text{-.04}} \rightsquigarrow (4,3)_{\text{+1}}$$
$$(1,1)_{\text{-.04}} \rightsquigarrow (2,1)_{\text{-.04}} \rightsquigarrow (3,1)_{\text{-.04}} \rightsquigarrow (3,2)_{\text{-.04}} \rightsquigarrow (4,2)_{\text{-1}} \ .$$

Calcular:

- $P(\ (2,3) \mid (1,3), \text{Derecha}\ ) = ?$
- $P(\ (4,2) \mid (3,2), \text{Arriba}\ ) = ?$

Aprendizaje por Refuerzo Pasivo

Programación Dinámica Adaptativa: Curvas de aprendizaje



Primera vez que se cae en estado terminal (4,2) con recompensa -1

*

ADP se torna intratable en espacios de estados grandes

Aprendizaje por Refuerzo Pasivo

Diferencias Temporales

- En lugar de aprender un modelo, se aprende la función de utilidad a partir de las transiciones observadas, pero respetando la ecuación de Bellman
  - Ej. Consideremos la transición de (1,3) -> (2,3). Imaginemos que después del primer trial se tiene:

$$U^{\pi}(1,3) = 0.84 \qquad U^{\pi}(2,3) = 0.92$$

Si dicha transición ocurriera todo el tiempo, de acuerdo a la ecuación de Bellman se esperaría:

$$U^{\pi}(1,3) = -0.04 + \gamma U^{\pi}(2,3)$$

Con γ=1 se tendría $U^{\pi}(1,3) = 0.88$, así, el actual estimado de 0.84 debería ser incrementado

Aprendizaje por Refuerzo Pasivo

Diferencias Temporales

- En el método de diferencias temporales, a cada transición observada de s para s', se realiza la siguiente actualización del valor de $U^{\pi}(s)$:

$$U^{\pi}(s) \leftarrow U^{\pi}(s) + \alpha(R(s) + \gamma U^{\pi}(s') - U^{\pi}(s))$$

Taza de aprendizaje

utilidad estimada a partir del sucesor observado

- Con α constante, el valor promedio de $U^{\pi}(s)$ converge al valor correcto.
- Si se define α como función que decrece con el aumento del numero de visitas (n) al estado s, esto es $\alpha(n)$, entonces el valor de $U^{\pi}(s)$ converge al valor correcto.

Ej: $\alpha(n) = 60/(59 + n)$

Aprendizaje por Refuerzo Pasivo

Diferencias Temporales

function PASSIVE-TD-AGENT($percept$) **returns** an action

inputs: $percept$, a percept indicating the current state s' and reward signal r'

persistent: π, a fixed policy
U, a table of utilities, initially empty
N_s, a table of frequencies for states, initially zero
s, a, r, the previous state, action, and reward, initially null

if s' is new **then** $U[s'] \leftarrow r'$

if s is not null **then**
increment $N_s[s]$
$U[s] \leftarrow U[s] + \alpha(N_s[s])(r + \gamma U[s'] - U[s])$

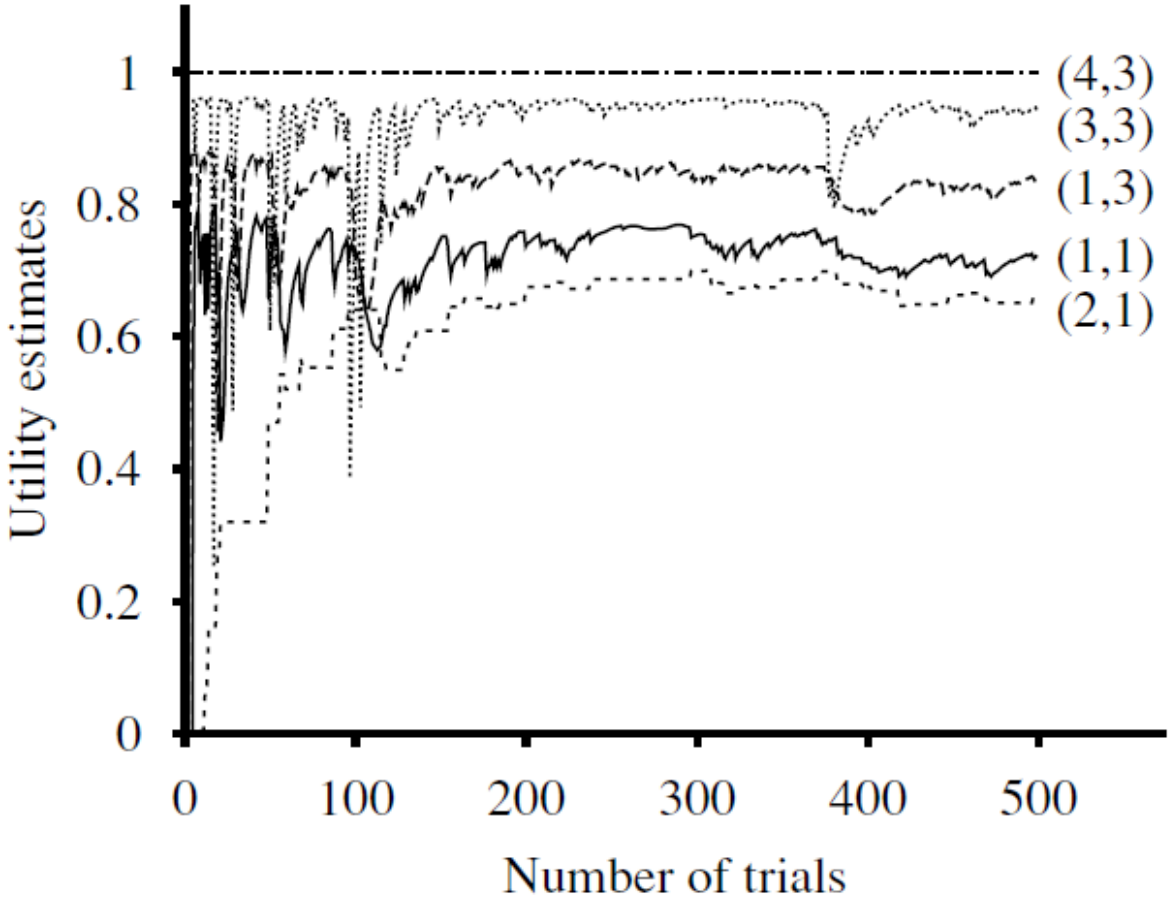
if s'.TERMINAL? **then** $s, a, r \leftarrow$ null **else** $s, a, r \leftarrow s', \pi[s'], r'$

return a

*

Aprendizaje por Refuerzo Pasivo

Diferencias Temporales: Curvas de aprendizaje



*

No es tan rápido como ADP y tiene mayor variabilidad

Aprendizaje por Refuerzo Pasivo

ADP versus Diferencias Temporales (TD)

- Diferencias Temporales requiere mas experimentos para convergir y tiene mayor variabilidad, pero requiere menos computación por observación
- ADP ajusta la utilidad del estado con todos sus sucesores que pueden ocurrir, TD solo ajusta el estado con su sucesor observado
- TD hace un simple ajuste por transición. ADP hace tantos ajustes como sean necesarios para restablecer consistencia entre las utilidades estimadas y el modelo de transiciones del entorno P

Aprendizaje por Refuerzo Activo

Definición

- El agente tiene que escoger acciones
- Objetivo: aprender las acciones óptimas, aquellas que obedecen a la ecuación de Bellman:

$$U(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s' \mid s, a) U(s')$$

- Idea: Partiendo de un modelo de transición arbitrario
  - Refinar el modelo a través de la experiencia
  - Utilizar Programación Dinámica Adaptativa + Iteración de Valor para encontrar la política óptima para el modelo actual. Repetir.
- Crucial: tenemos que dar una manera de aprender el modelo para todos los estados y acciones

Aprendizaje por Refuerzo Activo

Programación Dinámica Adaptativa Codiciosa

function GREEDY-ADP-AGENT($percept$) **returns** an action

inputs: $percept$, a percept indicating the current state s' and reward signal r'

persistent: mdp, an MDP with model P, rewards R, discount γ
U, a table of utilities, initially empty
N_{sa}, a table of frequencies for state–action pairs, initially zero
$N_{s'|sa}$, a table of outcome frequencies given state–action pairs, initially zero
s, a, the previous state and action, initially null

if s' is new **then** $U[s'] \leftarrow r'$; $R[s'] \leftarrow r'$

if s is not null **then**
increment $N_{sa}[s, a]$ and $N_{s'|sa}[s', s, a]$
for each t such that $N_{s'|sa}[t, s, a]$ is nonzero **do**
$P(t \mid s, a) \leftarrow N_{s'|sa}[t, s, a] \,/\, N_{sa}[s, a]$

$U \leftarrow$ VALUE-ITERATION(mdp)

$$\pi[s'] = \operatorname*{argmax}_{a \in A(s')} \sum_{s''} P(s'' \mid s', a) U(s'')$$

if s'.TERMINAL? **then** $s, a \leftarrow$ null **else** $s, a \leftarrow s', \pi[s']$

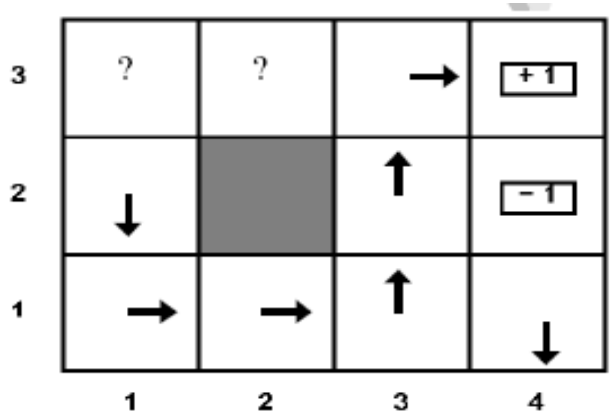
return a

*

Aprendizaje por Refuerzo Activo

Ejemplo: Programación Dinámica Adaptativa Codiciosa

- Imagine que el agente encuentra primero el siguiente camino:



*

- El agente siempre continuará usando esa política, nunca visitará los otros estados. ¿Por qué?
  - El agente no aprende las utilidades de las mejores regiones ya que con la política actual nunca se visita esas regiones.
  - La política es optima para el modelo aprendido, pero este no refleja el verdadero entorno

Aprendizaje por Refuerzo Activo

Exploración versus Explotación

- Exploración: escoger acciones sub-óptimas de acuerdo con el modelo actual para poder mejorar el modelo
- Explotación: escoger las mejores acciones para el modelo actual
- Pura explotación o pura exploración no sirve. Los agentes deben explorar más al inicio y explotar más al final.

Posibles Soluciones

- El agente escoge acciones aleatorias una fracción de tiempo (convergencia muy lenta)
- El agente escoge acciones con probabilidad inversamente proporcional al número de veces que la acción fue ejecutada en el estado. Se implementa a través de una función de exploración

Aprendizaje por Refuerzo Activo

Programación Dinámica Adaptativa con Exploración

- La iteración de valor en Greedy-ADP se realiza usando la ecuación de Bellman alterada:

$$U^+(s) \leftarrow R(s) + \gamma \max_a f\left(\sum_{s'} P(s' \mid s,a)U^+(s'),\ N(s,a)\right)$$

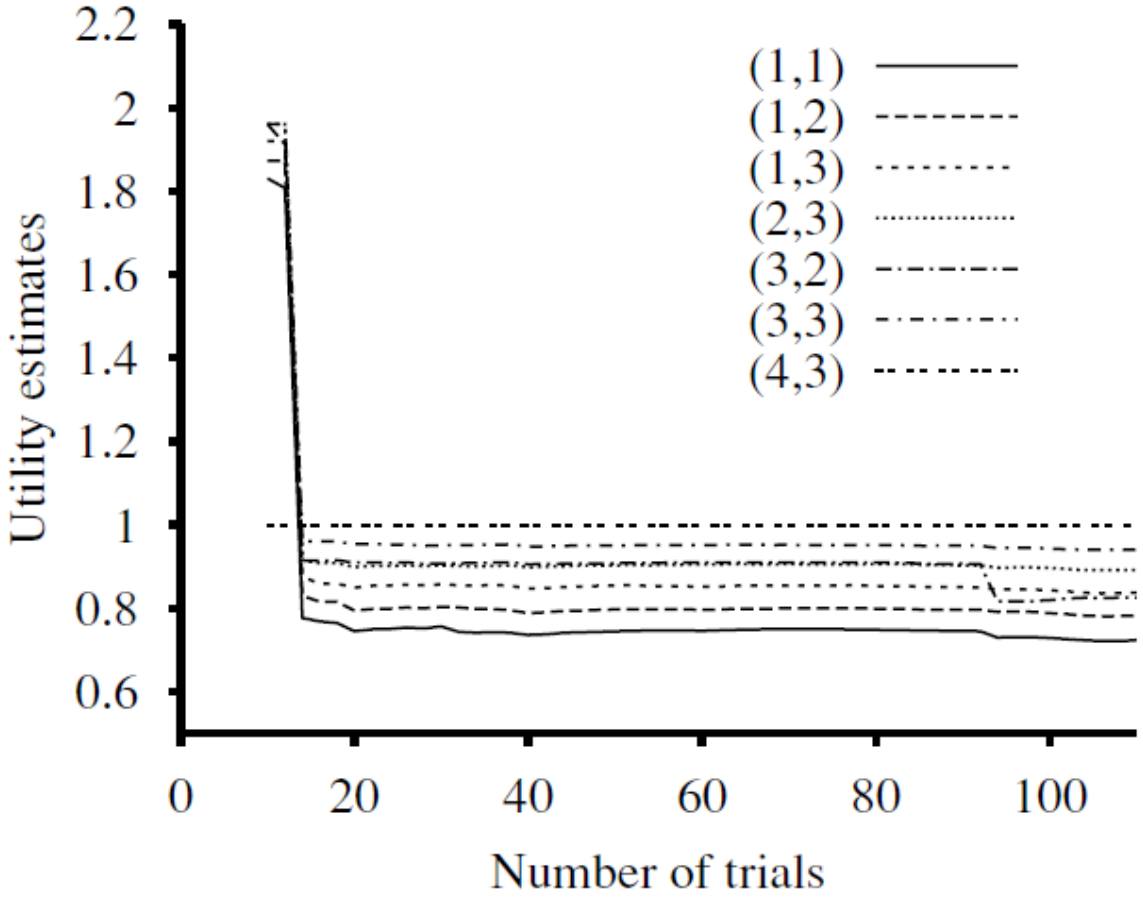
donde *f* es la función de exploración. Una forma simple es:

$$f(u,n) = \begin{cases} R^+ & \text{if } n < N_e \\ u & \text{otherwise} \end{cases}$$

- $U^+(s)$ es una utilidad optimista ya que, si el estado no ha sido visitado con frecuencia, muestra al estado con alta utilidad para hacer que el agente tome acciones que visiten dicho estado
- La acción que se toma $\pi(s)$ es la que maximiza *f*

Aprendizaje por Refuerzo Activo

Programación Dinámica Adaptativa con Exploración



*

Curva de aprendizaje de utilidades en el mundo 4x3 con ADP exploratorio. $R^+=2$, $N_e=5$

Aprendizaje por Refuerzo Activo

Diferencias Temporales en aprendizaje activo

- A diferencia del caso pasivo, en el caso activo el agente no tiene una política fija
- Una extensión natural seria aprender el modelo de transición P para poder escoger la acción que mas utilidad esperada le dé:

$$\pi(s) = \underset{a \in A(s)}{\operatorname{argmax}} \sum_{s'} P(s' \mid s, a) U(s')$$

La actualización de utilidades seria la misma forma que en el caso pasivo:

$$U^{\pi}(s) \leftarrow U^{\pi}(s) + \alpha(R(s) + \gamma\, U^{\pi}(s') - U^{\pi}(s))$$

- Existe un método alternativo que no requiere modelo: Q-learning

Aprendizaje por Refuerzo Activo

Q-learning

- Denotaremos ***Q(s, a)*** la utilidad de ejecutarse la acción ***a*** en el estado ***s***. La relación con la utilidad del estado es:

$$U(s) = \max_a Q(s,a)$$

- La función $Q(s,a)$ es útil porque a partir de ella podemos calcular directamente la política, si necesitar de modelo de transicion.
- **Ecuación de actualización**: con cada transición observada ***s*** -> ***s'*** con acción ***a*** ejecutada se actualiza $Q(s,a)$:

$$Q(s,a) \leftarrow Q(s,a) + \alpha(R(s) + \gamma \max_{a'} Q(s',a') - Q(s,a))$$

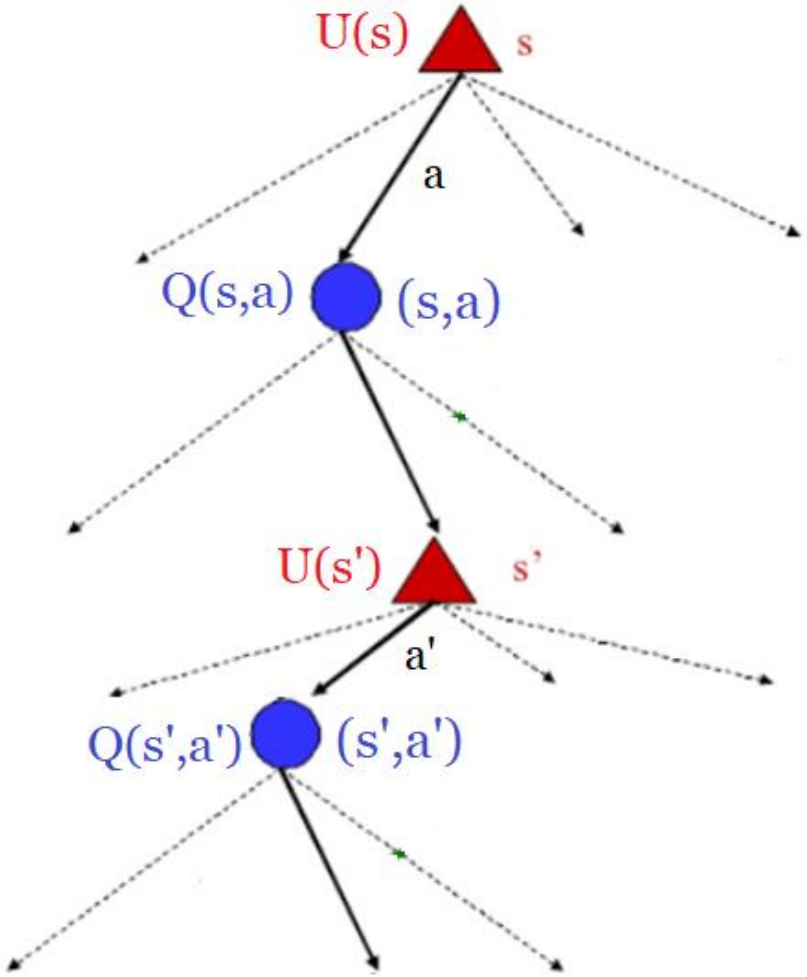
- A medida que se realiza iteraciones, los valores $Q(s,a)$ se acercan a los valores de equilibrio:

$$Q(s,a) = R(s) + \gamma \sum_{s'} P(s' \mid s,a) \max_{a'} Q(s',a')$$

Aprendizaje por Refuerzo Activo

Q-learning:

Opera en el espacio de Q-estados



Aprendizaje por Refuerzo Activo

Q-learning:

*

function Q-LEARNING-AGENT($percept$) **returns** an action
  inputs: $percept$, a percept indicating the current state s' and reward signal r'
  persistent: Q, a table of action values indexed by state and action, initially 0
    N_{sa}, a table of frequencies for state–action pairs, initially zero
    s, a, r, the previous state, action, and reward, initially null

  if TERMINAL?(s) **then** $Q[s, None] \leftarrow r'$
  if s is not null **then**
    increment $N_{sa}[s, a]$
    $Q[s, a] \leftarrow Q[s, a] + \alpha(N_{sa}[s, a])(r + \gamma \max_{a'} Q[s', a'] - Q[s, a])$
  $s, a, r \leftarrow s', \text{argmax}_{a'} f(Q[s', a'], N_{sa}[s', a']), r'$
  return a

f es la misma función de exploración del ADP exploratorio.

$$f(u, n) = \begin{cases} R^+ & \text{if } n < N_e \\ u & \text{otherwise} \end{cases}$$

Material complementar

- https://www.youtube.com/watch?v=gzFN6UTTpH0&t=178s
- https://www.youtube.com/watch?v=pljiFgRnBAQ
- https://www.youtube.com/watch?v=ZoRMKs8XLSA
- https://www.analyticsvidhya.com/blog/2017/01/introduction-to-reinforcement-learning-implementation/

Preguntas?