PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

INTELIGENCIA ARTIFICIAL (INF371)

APRENDIZAJE POR REFUERZO (PARTE 2)

Dr. Edwin Villanueva Talavera

Contenido

- Politica Óptima. Ecuaciones de Bellman
- Iteración de Valor
- Iteración de Política
- Aprendizaje por Refuerzo Pasivo
  - Estimación Directa de la Utilidad
  - Programación Dinámica Adaptativa
  - Aprendizaje de Diferencias Temporales

Bibliografía:

Capitulo 17.1 - 17.3; 21.1 - 21.4 del libro:

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

Procesos de decisiones de Markov

Evaluación de Políticas

- Dada la naturaleza no determinista de PDM, una política no genera una única secuencia de estados, si no un conjunto de secuencias que difieren según el modelo de transición
- Para evaluar una política π podemos usar el valor esperado de la utilidad de las secuencias (recompensas descontadas) que puede generar la política. En el estado **s** este valor seria:

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

donde S_t es una variable aleatoria que representa el estado del agente en el tiempo t cuando se ejecuta la política π empezando en ***s***

Procesos de decisiones de Markov

Política Optima

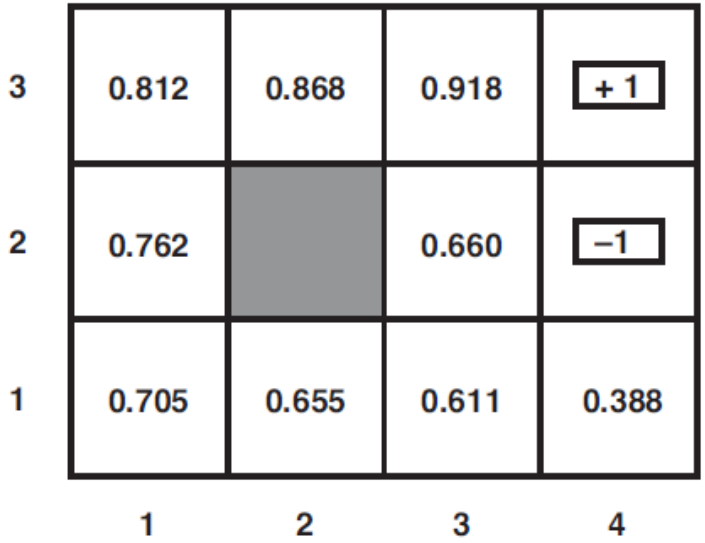
- La acción recomendada en s por una política óptima, **π*(s)**, es aquella que genera el mas alto valor $U^{\pi}(s)$ de todas las políticas posibles que comienzan en s:

$$\pi^*(s) = \underset{\pi}{\operatorname{argmax}}\, U^{\pi}(s)$$

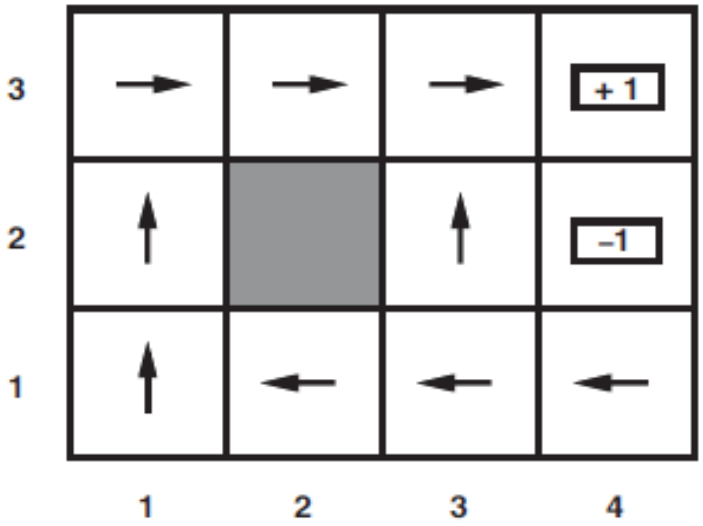
- Definimos la utilidad de un estado *s* como el valor $U(s) = U^{\pi^*}(s)$ esto es, la expectativa de la suma de recompensas descontadas a partir de **s** dado que el agente ejecuta una política optima

Procesos de decisiones de Markov

Ejemplo de Utilidades



Utilidades (com politica óptima)



Politica óptima

No es lo mismo Utilidad de um estado (U(s)) que Recompensa del estado (R(s) !

En la práctica, Cómo encontramos la política optima?

Recuerde, la política debe especificar una acción para cada estado

Procesos de decisiones de Markov

Ecuación de Bellman:

- Ecuación recursiva definiendo la utilidad de un estado:

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

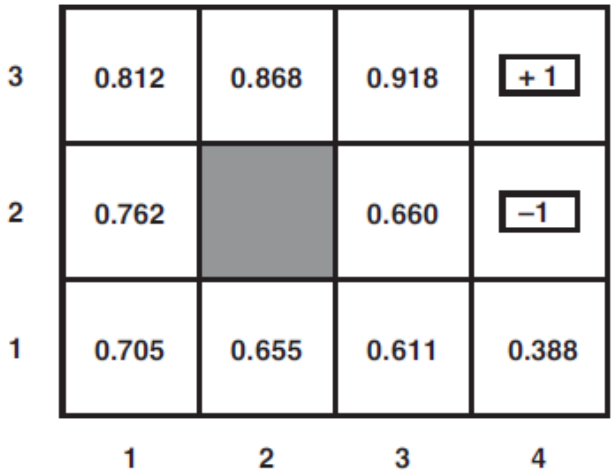
es la recompensa inmediata correspondiente a ese estado + la utilidad descontada esperada del próximo estado, suponiendo que el agente escoja la acción óptima

- Al resolver la ecuación de Bellman y obtener las utilidades *U(s)* para cada estado **s** se puede encontrar la política optima:

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

Procesos de decisiones de Markov

Ejercicio Ecuación de Bellman:



Utilidades encontradas al resolver la ecuación de Bellman con R(s) = 0.04, $\gamma=1$

¿Que acción ejecutar en (1,1)?

$$U(1,1) = -0.04 + \gamma \max\{\ 0.8U(1,2) + 0.1U(2,1) + 0.1U(1,1),$$ Arriba

$$0.9U(1,1) + 0.1U(1,2),$$ Izquierda

$$0.9U(1,1) + 0.1U(2,1),$$ Abajo

$$0.8U(2,1) + 0.1U(1,2) + 0.1U(1,1)\ \}$$ Derecha

Arriba seria la acción recomendada por la política optima.

Procesos de decisiones de Markov

Resolviendo la Ecuación de Bellman:

- ¿Por que no usar algoritmos de búsqueda?
  - Árbol puede ser infinito
  - Tendríamos que hacer una búsqueda para cada estado
  - Repite muchas veces los mismos cálculos siempre que el mismo estado fuese alcanzado.
- Idea: **Iteración de valor**
  - Calcular valores de utilidad óptimos para todos los estados simultáneamente, usando aproximaciones sucesivas.

Iteración de Valor

Algoritmo de Iteración de Valor:

- Si existe n estados posibles, tendremos n ecuaciones de Bellman (una para cada estado) con n incógnitas (las funciones de utilidad).
- Para encontrar la política optima tenemos que resolver ese sistema de ecuaciones NO Lineares (por el operador max).

Iteración de Valor

Algoritmo de Iteración de Valor:

- Valores iniciales arbitrarios para las utilidades $U_0(s)$
- Repetir hasta llegar al equilibrio
  - Calcular el lado derecho de la ecuación de Bellman
  - Actualizar la utilidad de cada estado a partir de la utilidad de sus vecinos

Lo que se hace en cada iteración es la actualización de Bellman:

$$U_{i+1}(s) \leftarrow R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s' \mid s, a) U_i(s')$$

- La convergencia es garantizada si se usa recompensas descontadas
- Los valores finales serán soluciones para las ecuaciones de Bellman

Iteración de Valor

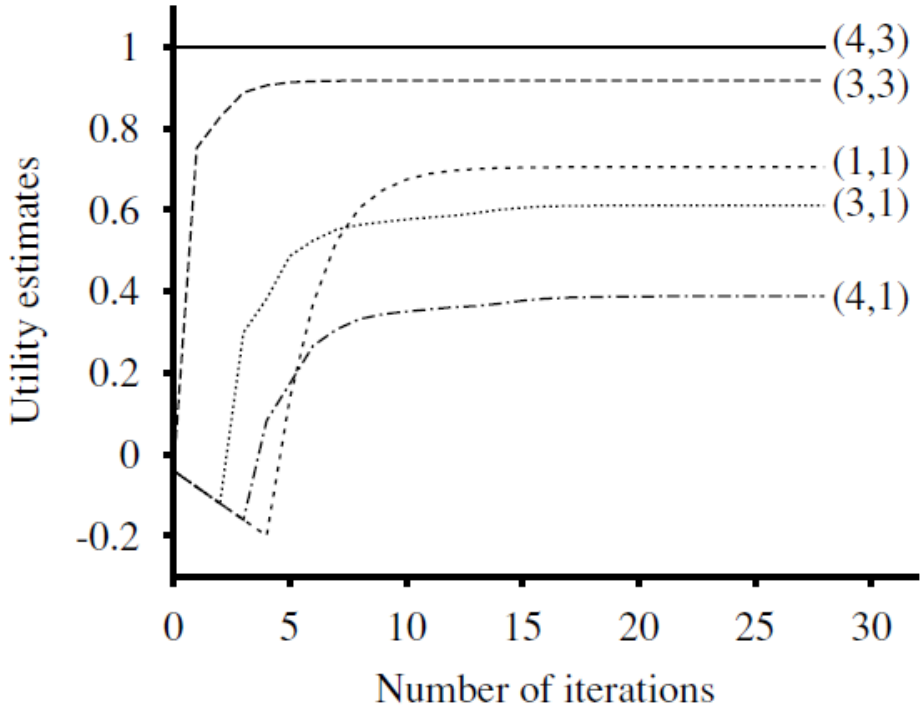
Algoritmo de Iteración de Valor:

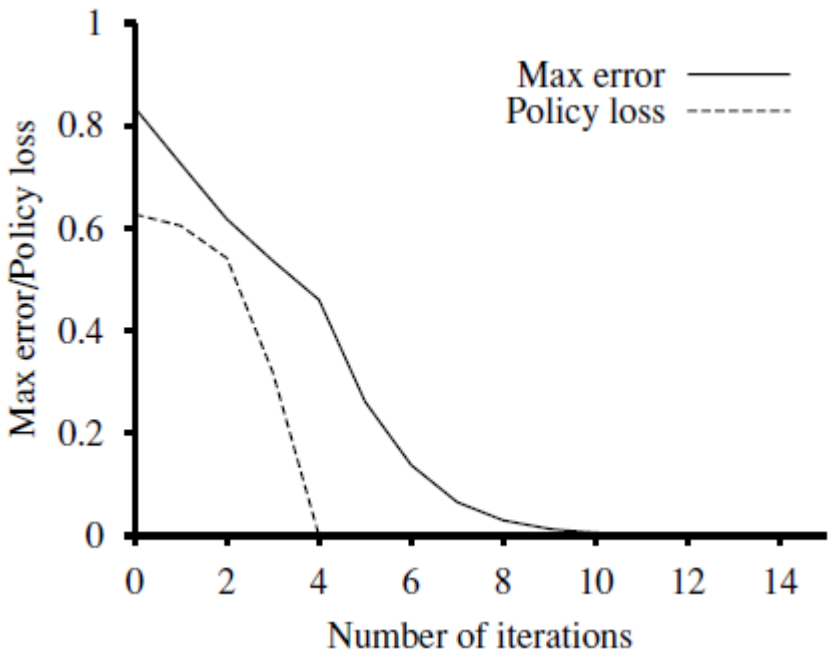
function VALUE-ITERATION(mdp, ϵ) **returns** a utility function
  inputs: mdp, an MDP with states S, actions $A(s)$, transition model $P(s' \mid s, a)$,
    rewards $R(s)$, discount γ
    ϵ, the maximum error allowed in the utility of any state
  local variables: U, U', vectors of utilities for states in S, initially zero
    δ, the maximum change in the utility of any state in an iteration

  repeat
    $U \leftarrow U'$; $\delta \leftarrow 0$
    for each state s **in** S **do**
      $U'[s] \leftarrow R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s' \mid s, a)\ U[s']$
      if $|U'[s] - U[s]| > \delta$ **then** $\delta \leftarrow |U'[s] - U[s]|$
  until $\delta < \epsilon(1-\gamma)/\gamma$
  return U

Iteración de Valor

Evolución de utilidades y politicas en el ejemplo de PDM con el algoritmo de Iteración de Valor:





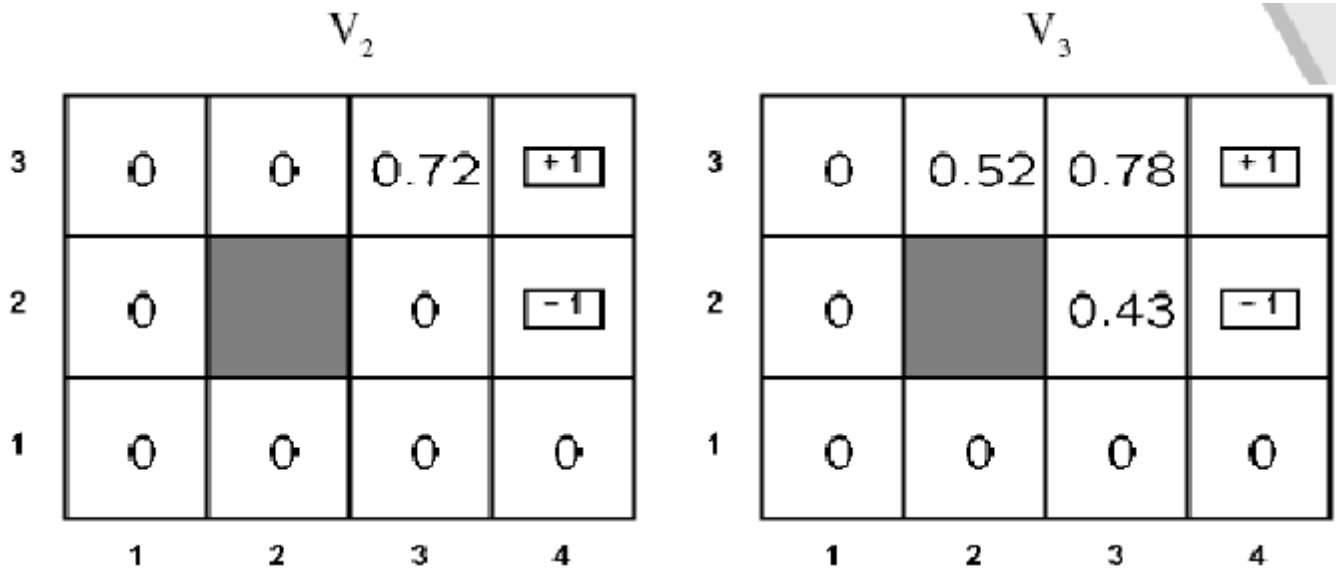
Evolución de Utilidades

Evolución de maximo Error: max($||U_i - U||$)

Evolución de Error de Politica: $||U^{\pi_i} - U||$

Iteración de Valor

Evolución de utilidades en el ejemplo de PDM con el algoritmo de Iteración de Valor:



La información se propaga para fuera a partir de los estados terminales.

Iteración de Política

Fundamentos:

- Muchas veces no se necesita una estimación precisa de las utilidades para generar una política óptima
- Iteración de Política se basa en dicha idea para simplificar el cálculo de función de utilidades
- Empezando con una política inicial π_0 se itera:
  - Evaluación de política: dada la política π_i se calcula la utilidad U_i (sin usar el operador MAX) para cada estado
  - Mejoramiento de política: Calcular la nueva política π_{i+1} maximizando la utilidad esperada (calculada en base a U_i)

$$\pi_{i+1}(s) = \underset{a \in A(s)}{\operatorname{argmax}} \sum_{s'} P(s' \mid s, a)U_i(s')$$

Iteración de Política

Algoritmo:

function POLICY-ITERATION(mdp) **returns** a policy
  inputs: mdp, an MDP with states S, actions $A(s)$, transition model $P(s' \mid s, a)$
  local variables: U, a vector of utilities for states in S, initially zero
    π, a policy vector indexed by state, initially random

  repeat
    $U \leftarrow$ POLICY-EVALUATION(π, U, mdp)
    $unchanged? \leftarrow$ true
    for each state s **in** S **do**
      if $\max_{a \in A(s)} \sum_{s'} P(s' \mid s, a)\ U[s'] > \sum_{s'} P(s' \mid s, \pi[s])\ U[s']$ **then do**
        $\pi[s] \leftarrow \operatorname{argmax}_{a \in A(s)} \sum_{s'} P(s' \mid s, a)\ U[s']$
        $unchanged? \leftarrow$ false
  until $unchanged?$
  return π

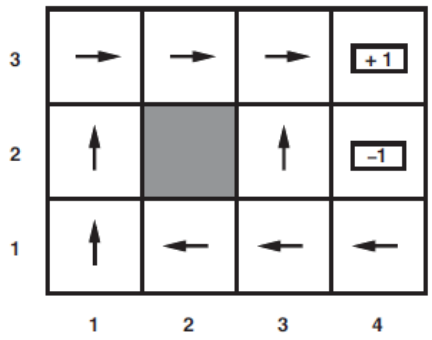
Iteración de Política

Como implementar POLICY-EVALUATION:

- No se necesita resolver las ecuaciones de Bellman tradicionales, ya que las acciones están fijadas por la política
- En iteración i la política π_i especifica la acción $\pi_i(s)$ en estado *s*
  Esto significa que la utilidad de s a partir de sus vecinos es:

$$U_i(s) = R(s) + \gamma \sum_{s'} P(s' \mid s, \pi_i(s)) U_i(s')$$

- Para el ejemplo, se tiene las ecuaciones:



$$\begin{aligned} U_i(1,1) &= -0.04 + 0.8U_i(1,2) + 0.1U_i(1,1) + 0.1U_i(2,1) \ , \\ U_i(1,2) &= -0.04 + 0.8U_i(1,3) + 0.2U_i(1,2) \ , \\ U_i(2,1) &= \\ &\vdots \end{aligned}$$

Iteración de Política

Como implementar POLICY-EVALUATION:

- El conjunto de ecuaciones se puede resolver con algebra lineal en $O(n^3)$, aunque puede ser prohibitivo para n grande
- Alternativamente, se puede usar una versión iterativa simplificada de la actualización de Bellman:

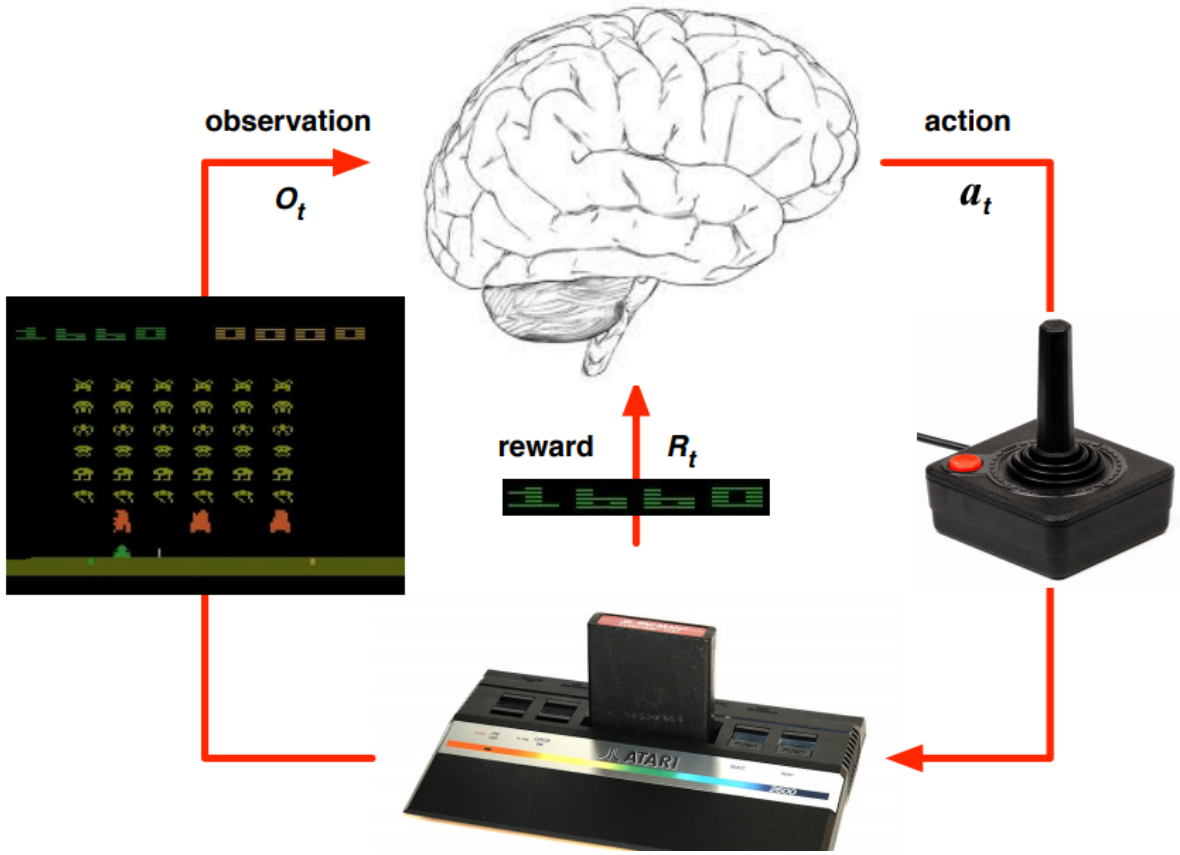
$$U_{i+1}(s) \leftarrow R(s) + \gamma \sum_{s'} P(s' \mid s, \pi_i(s)) U_i(s')$$

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 $\pi^*(s)$

Aprendizaje por Refuerzo

Ejemplo: Juegos de Atari



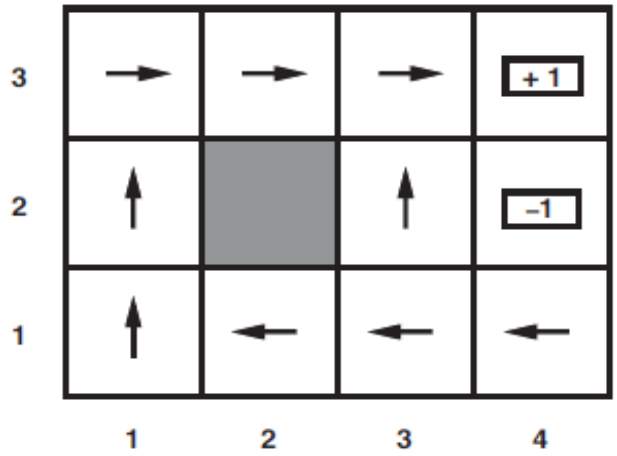
- Reglas del juego desconocidas. Se necesita aprender de la experimentación: mover el joystick y ver los pixels y el score resultante

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) :

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

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

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

¿Cómo estimar $U^{\pi}(s)$ 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)

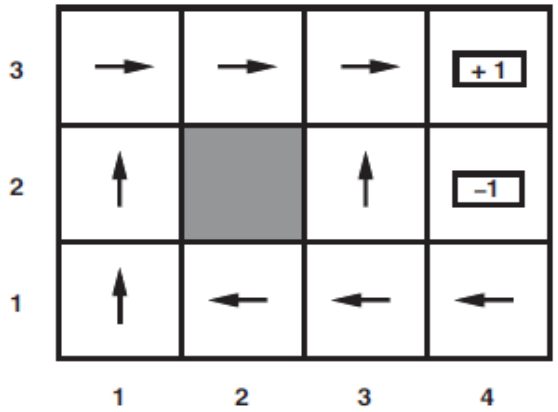
function PASSIVE-ADP-AGENT($percept$) **returns** an action
  inputs: $percept$, a percept indicating the current state s' and reward signal r'
  persistent: π, a fixed policy
    mdp, an MDP with model P, rewards R, discount γ
    U, a table of utilities, initially empty
    N_{sa}, a table of frequencies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

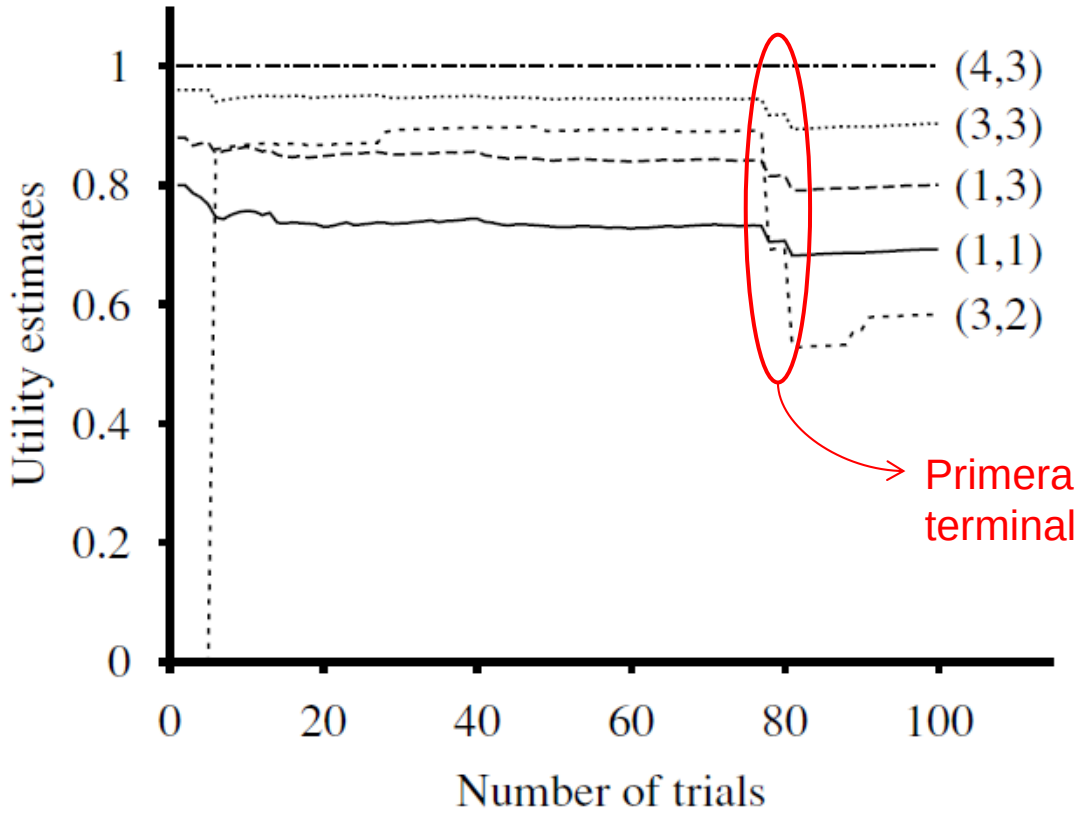
$$(1,1)_{\text{-.04}} \rightsquigarrow (1,2)_{\text{-.04}} \rightsquigarrow (1,3)_{\text{-.04}} \rightsquigarrow (1,2)_{\text{-.04}} \rightsquigarrow (1,3)_{\text{-.04}} \rightsquigarrow (2,3)_{\text{-.04}} \rightsquigarrow (3,3)_{\text{-.04}} \rightsquigarrow (4,3)_{\text{+1}}$$
$$(1,1)_{\text{-.04}}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

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 $N_s[s]$

$U[s] \leftarrow U[s] + \alpha(N_s[s])(r + \gamma U[s'] - U[s])$

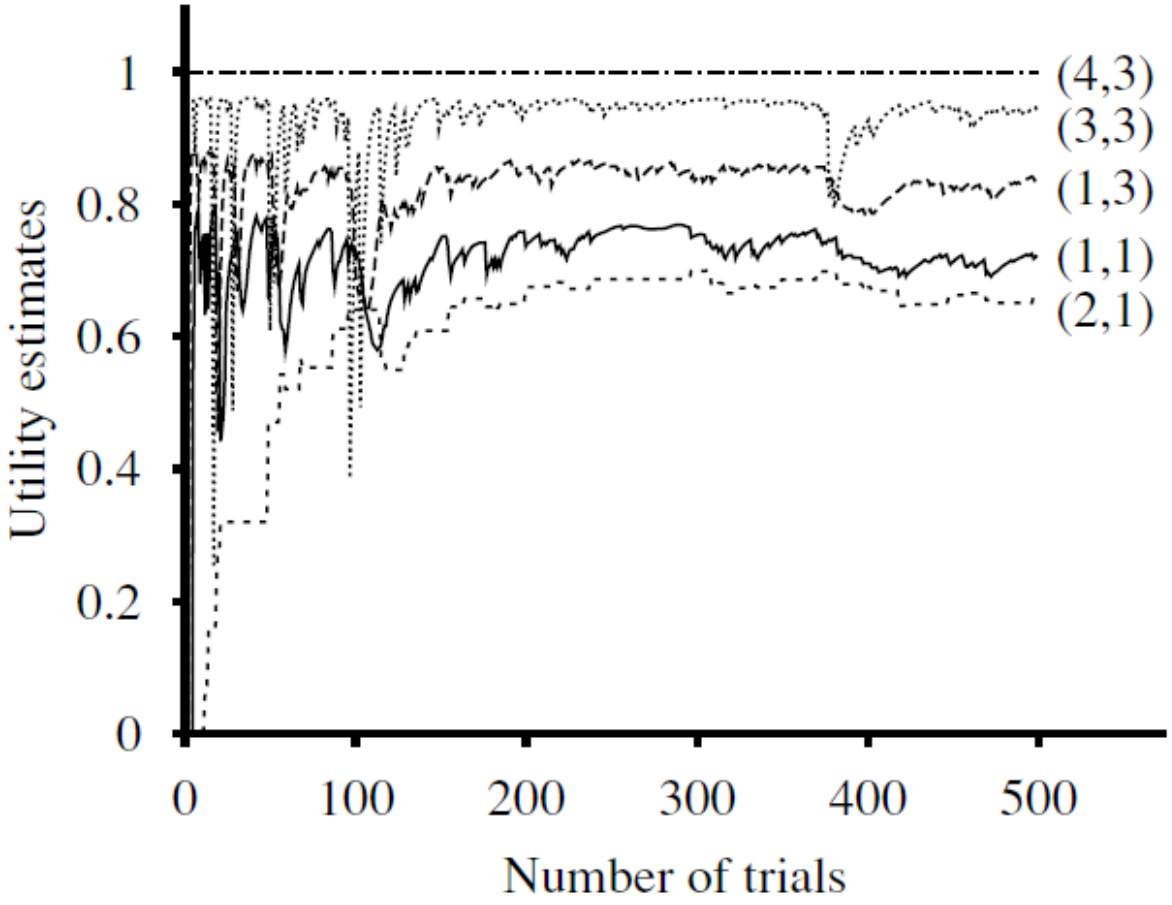
if s'.TERMINAL? **then** $s, a, r \leftarrow$ null **else** $s, a, r \leftarrow s', \pi[s'], r'$

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

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?