





---

# Aprendizaje por Refuerzo

## Diplomado en Computación Inteligente, UPB 2005

Jerónimo Castrillón Mazo  
IEO UPB.  
[www.geocities.com/jeronimocm](http://www.geocities.com/jeronimocm)



---

## Previa

- Es un tema muy vacano
- Nuestra tesis
- Presentaciones y conocimiento en general debido a Richard S. Sutton y Andrew G. Barto
- Muchas otras fuentes (ver bib.)
- Libro
- Existen muchos *papers* con modificaciones a estas técnicas.
- **Esta presentación se hizo fuertemente basada en el libro de Sutton y Barto ... me parece un libro excelente!**

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. [jeronimocm@yahoo.com](mailto:jeronimocm@yahoo.com)

2

## Previa (2)

- Ordenamiento del módulo:  
idea: introducción gradual al aprendizaje por refuerzo.
  - Introducción
  - Elementos del RL: básicamente notación y un ejemplo
  - Acercamiento al RL: ejemplo “no completo” de RL
  - El problema de RL: análisis del problema completo de RL ... con todo los “gallos”
  - Luego veremos los principales métodos para solucionar el problema completo!

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

3

## Contenido

- **Introducción**
- Elementos del RL
- Acercamiento al RL
- El problema del RL
- Métodos elementales
  - *Dynamic Programming*
  - *Monte Carlo Methods*
  - *Temporal Difference learning*

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

4



## Introducción: Contenido

- Que es RL?
- Historia

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

5



## Introducción: Que es RL?

Recordemos los otros tipos de aprendizaje

### Aprendizaje Supervisado

- Se cuenta con un instructor que indica cuál es la salida deseada ante una determinada entrada.
- Con esta información el sujeto del aprendizaje puede corregir sus parámetros internos para que la salida ante esa entrada se parezca más a la deseada.
- Ejemplos: redes neuronales, *machine learning*, *statistical pattern recognition*, etc.

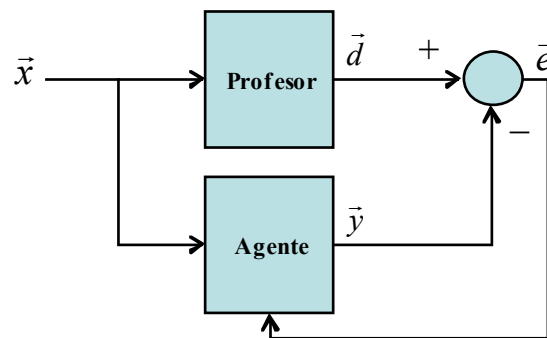
Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

6

## Introducción: Que es RL? (2)

Esquema



Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

7

## Introducción: Que es RL? (3)

### Aprendizaje no supervisado

- El sujeto del aprendizaje no cuenta con información adicional de la entrada que se le presenta.
- El objetivo del agente es construir representaciones que sean útiles para procesos de decisión, predicción, reconocimiento, razonamiento, comunicación, etc.
- Ejemplos: *data mining*, jerarquización de documentos, *clustering*, etc.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

8

## Introducción: Que es RL? (4)

### Aprendizaje con Índice de Desempeño

- Al sujeto del aprendizaje se le presenta una medida cuantitativa del desempeño.
- Ejemplos: algoritmos evolutivos (PG, AG, PE), inteligencia *swarm* (ACO, PSO, etc).

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

9

## Introducción: Que es RL? (5)

### Aprendizaje por refuerzo (*Reinforcement Learning*, RL)

- Sutton y Barto: “*Reinforcement learning is learning what to do – how to map situations to actions – so as to maximize a numerical reward signal. The learner is not told which actions to take, as in most forms of machine learning, but instead must discover which actions yield the most reward by trying them.*”
- Nosotros: “Trata de imitar la forma en la cual un organismo vivo aprende un determinado comportamiento mediante la correlación entre sus acciones y el placer o el dolor que éstas le producen”

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

10

## Introducción: Que es RL? (6)

- Nosotros: El aprendizaje por refuerzo es una metáfora del condicionamiento instrumental (Pavlov), según el cual la acción tomada por un organismo en respuesta a un estímulo tiende a ser más frecuente según el carácter agradable (placer) o desagradable (dolor) de la consecuencia ambiental de dicha acción.

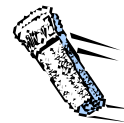
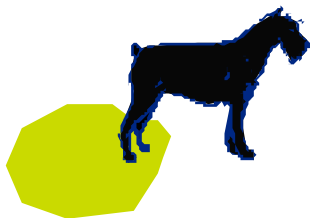
Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

11

## Introducción: Que es RL? (7)

- Ejemplo: El perro que hace sus necesidades donde no debe ...



Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

12

## Introducción: Que es RL? (8)

- Ejemplo: El perro que hace sus necesidades donde debe ...



**En ambas situaciones al perro no se le dice que debió hacer!!  
Sólo se le da una medida cualitativa de cómo lo hizo!!**

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

13

## Introducción: Que es RL? (9)

### Otros nombres

- aprendizaje con un crítico
- *Goal – oriented learning*
- *Trial and error learning*

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

14

## Introducción: Que es RL? (10)

### Ejemplos:

- Las vacas dentro de los potreros
- Mosca en un vaso
- Caminar
- Evasión de obstáculos
- ...

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

15

## Introducción: Que es RL? (11)

- Diferentes acercamientos a la solución de un problema en robótica:



Tomado de "Neural Networks and Q-Learning for Robotics"  
Claude F. TOUTET, IJCNN '99 Tutorial

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

16

## Introducción: Historia

- Dos hilos principales: *trial and error* (psicología) y teoría de control óptimo ...
- RL existe desde los principios del estudio de inteligencia artificial (1950).
- Tuvo un período muerto en la década de los 60s y 70s.
- Revivió a finales de los 70s y comienzos de los 80s con los trabajos de Harry Klopff y luego de Richard S. Sutton y Andrew G. Barto.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

17

## Introducción: Historia (2)

### Teoría de control óptimo

- Richard Bellman, 1957
- *Dynamic Programming*
- Ecuación de Bellman (**Más de esto adelante**)
- Procesos de Decisión Markovianos, *Markovian decision processes* (MDPs).

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

18

## Introducción: Historia (3)

### Ensayo y Error

- Ley del efecto 1911, “*Law of effect*”: *Of several responses made to the same situation, those which are accompanied or closely followed by satisfaction to the animal will, other things being equal, be more firmly connected with the situation, so that, when it recurs, they will be more likely to recur; those which are accompanied or closely followed by discomfort to the animal will, other things being equal, have their connections with that situation weakened, so that, when it recurs, they will be less likely to occur. The greater the satisfaction or discomfort, the greater the strengthening or weakening of the bond* (Edward Thorndike).
- *Selectional*: selecciona acciones y compara las consecuencias.
- *Assosiative*: las acciones que escoge por selección se asocian a las situaciones ... correlación estado – acción

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

19

## Introducción: Historia (4)

- Enfoque computacional, los primeros Minsky, Farley y Clark 1954 con SNARCs (*Stochastic Neural-Analog Reinforcement Calculators*)
- *Credit assignment problem*, 1961 – paper de Minsky: “*Steps Toward Artificial Intelligence*”.
- Luego la mayoría se enfocaron en generalización y reconocimiento de patrones, e.g. aprendizaje supervisado.
- John Andrae 1963, Nueva Zelanda desarrolló STeLLA
- Donald Michie 1961, 1963: MENACHE (*Matchbox Educable Noughts and Crosses Engine*)

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

20



## Introducción: Historia (5)

- Harry Klop f (1972, 1975, 1982): “*most responsible for reviving the trial-and-error thread of RL*” (Sutton y Barto 1998).
- Ayudó a la diferenciación entre aprendizaje supervisado y aprendizaje por refuerzo.
- Sutton y Barto con sus múltiples publicaciones: 1981 (4), 1982, 1985 (4), 1986, 1987 ... y finalmente **EL LIBRO DE RL: Reinforcement Learning: An Introduction**, publicado en 1998 por el MIT Press.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

21



## Introducción: Historia (6)

### Temporal Difference

- Se basan en la diferencia entre dos aproximaciones sucesivas en el tiempo de algún valor.
- Inspirado en la psicología del aprendizaje animal: Minsky (1954), Arthur Samuel (1959).
- Período muerto ...
- Witten en 1977 creó lo que ahora se conoce como TD(0)
- Sutton y Barto también retomaron estas ideas a finales de los 70s.
- Finalmente Chris Watkin desarrolló el algoritmo *Q-learning*.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

22

# Contenido

- Introducción
- **Elementos del RL**
- Acercamiento al RL
- El problema del RL

Métodos elementales

- *Dynamic Programming*
- *Monte Carlo Methods*
- *Temporal Difference learning*

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

23

# Elementos del RL

Los elementos del aprendizaje por refuerzo son:

- El agente
- El ambiente
- Estados
- Acciones
- La política (*the policy*)
- La función de recompensa
- La función de valor (*value function*)
- El modelo (opcional)
- Ejemplo: tricky

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

24



## Elementos del RL: Agente – Ambiente

### Interfaz Agente-Ambiente

- El agente: sujeto del aprendizaje, aprendiz, *decision-maker*.
- Ambiente: elemento con el cual interactúa el agente. Comprende todo lo que no sea este último.
- Agente y ambiente interactúan constantemente: el agente selecciona acciones y el ambiente responde a éstas y presenta nuevas situaciones así como la recompensa.
- **¿¿Cuál es la frontera agente-ambiente?? ... ¿será la separación física entre un robot y el entorno donde se desenvuelve?**

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

25



## Elementos del RL: Estados

- Son una representación/medición del ambiente.
- Puede entenderse como un sensado del mismo ... puede ser total o parcial.
- Ejemplo1: robot que evade obstáculos. Un estado o una percepción del ambiente será la medida en un instante dado de todos sus sensores (IR, US, Capacitancia, etc.)
- Ejemplo2: bioreactor. Los estados pueden la medida de un sensor de temperatura y de otros sensores diferentes filtrados y retardados además de entradas simbólicas representando los ingredientes químicos en el tanque.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

26

## Elementos del RL: Estados (2)

- Los estados pueden ser variables continuas o discretas
- Los estados pueden no ser sensados por completo o puede que estados diferentes sean percibidos de igual manera ... estos problemas se dice que son “Parcialmente Observables”.
- Notación:
  - $s$  : representación de un estado general
  - $s_t$  : representación del estado tomado en el tiempo  $t$
  - $S$  : conjunto de todos los estados posibles/detectables

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

27

## Elementos del RL: Acciones

- Son todas las posibles reacciones que en cada estado el agente puede tomar para modificar el ambiente, e.g. su percepción del mismo.
- Ejemplo1: robot que evade obstáculos. Las acciones pueden ser un conjunto discreto: adelante, atrás, deracha, izquierda. Estas se traducen a niveles de voltaje en los motores.
- Ejemplo2: bioreactor. Velocidad con la que se revuelven los reactivos y acciones de control sobre sistemas de regulación de temperatura, e.g. resistencias de calefacción y sistemas de ventilación.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

28

## Elementos del RL: Acciones (2)

- Las acciones también pueden ser continuas ... **complica enormemente el problema!**
- Al tomar una acción se dice que cambia el ambiente porque se espera que el valor sentido luego de la acción halla cambiado.
- Notación:
  - $a$  : representación general de una acción
  - $a_t$  : representación la acción tomada en el tiempo  $t$
  - $A$  : conjunto de todas las posibles acciones
  - $A(s_t)$  : conjunto de todas las acciones posibles en el estado  $s_t$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

29

## Elementos del RL: Política

### Política (*policy*)

- Es la que define el comportamiento del agente.
- Indica que acción se debe tomar en cada estado.
- Representa, en últimas, el objeto del aprendizaje.
- Psicología: conjunto de parejas (reglas) estímulo – respuesta.
- Sutton y Barto: *The policy is the core of a reinforcement learning agent.*
- Puede ser una simple *lookup table* o una compleja función que implique realizar procesos de búsqueda\*

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

30

## Elementos del RL: Política (2)

- Notación: la política es un mapeo del conjunto de estados al conjunto de acciones,

$$\begin{aligned} \pi: S &\rightarrow A \\ s &\rightarrow a = \pi(s) \end{aligned} \quad \text{Política determinística}$$

- En general la política es un mapeo estocástico de la forma:

$$\begin{aligned} \pi: S \times A &\rightarrow [0,1] \\ (s, a) &\rightarrow P[a_t = a | s_t = s] = \pi(s, a), \quad A = A(s_t) \end{aligned}$$

**Política estocástica: ante un estado entrega la probabilidad de realizar una determinada acción.**

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

31

## Elementos del RL: Política (3)

- Ejemplo 1: robot que evade obstáculos. Sean las acciones:  $a_1$  ir adelante,  $a_2$  atrás,  $a_3$  izquierda y  $a_4$  derecha. Sean los subconjuntos de  $S$ :  $S_1$  obstáculos adelante,  $S_2$  obstáculos a los lados,  $S_3$  obstáculos atrás.

Determinística:

$$\pi(s) = \begin{cases} a_1 & s \notin S_1 \\ a_3 & s \in S_1 \end{cases}$$

Estocástica:

$$\pi(s_i, a_j) = \begin{cases} 0.1 & i=1, j=1 & 0.5 & i=1, j=3 \\ 0.6 & i=2, j=1 & 0.5 & i=1, j=4 \\ 0.3 & i=3, j=1 & & \vdots \end{cases}$$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

32

## Elementos del RL: Política (4)

### ... Política

- Ejemplo2: Bioreactor. En él, la política está definida por un conjunto de reglas de control.
- Ejemplo3: perro casero. ¿Cuál será la mejor política??

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

33

## Elementos del RL: Función de Recompensa

### Función de Recompensa (*reward function*)

- Define la **meta** en un problema de aprendizaje por refuerzo.
- Esta función entrega un escalar como consecuencia de la acción tomada en el estado anterior.
- Puede ser función del estado actual o de la pareja estado-acción.
- El escalar indica la conveniencia intrínseca del estado (de la pareja estado-acción).
- Determina que son buenos y malos eventos
- La recompensa puede relacionarse con el **placer** y el **dolor**.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

34

## Elementos del RL: Función de Recompensa (2)

- No es alterable por el agente
- Como veremos influye directamente en la mejora de la política.
- **La manera como se define esta función es VITAL ... por experiencia.**
- Notación:

$r$  : representación general de una recompensa

$r_t$  : recompensa obtenida en el paso de tiempo  $t$

$f_r(\cdot)$  : función de recompensa

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

35

## Elementos del RL: Función de Recompensa (3)

- En general son funciones estocásticas

$$\begin{aligned} f_r : S \times A &\rightarrow \hat{R} \subset \mathbb{R} \\ (s_t, a_t) &\rightarrow r_{t+1} = f_r(s_t, a_t) \end{aligned}$$

- Ejemplo1: robot que evade obstáculos. La función de recompensa puede ser sencilla ... un  $-1$  si se choca y cero en los demás estados. **¿funcionaría bien?**

**Pensar en algo más elaborado ...**

- Ejemplo2: perro casero. **¿Cuál sería la función? ¿depende del estado y de la acción? , ¿Cómo?**

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

36

## Elementos del RL: Función de Recompensa (4)

- De nuevo: **“La manera como se define esta función es VITAL”**
- La recompensa debe reflejar lo que uno quiere que el agente haga.
- Ejemplo3: ajedrez. ¿Se debe premiar por eliminar fichas del rival? ¿Qué es en realidad lo que se quiere?

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

37

## Elementos del RL: Función de Recompensa (5)

- Ejemplo de nuestra función de recompensa

$$g^{t+1} = \begin{cases} \frac{b^{t+1}}{2} + \frac{(v_{der}^t + v_{izq}^t)}{v_{der,max} + v_{izq,max}} & \text{si no hay choque} \\ g_{choque} & \text{si hay choque} \end{cases} \quad b^{t+1} = \begin{cases} -1 & \text{si } \max_i(u_i^{t+1}) > \max_i(u_i^t) \\ 1 & \text{si } \max_i(u_i^{t+1}) < \max_i(u_i^t) \\ 0 & \text{de otra manera} \end{cases}$$

$$g_{choque} = \frac{-1.5}{1 - \gamma}$$

### Características:

- Premia el desplazamiento hacia delante a alta velocidad.
- Premia el alejamiento de los obstáculos.
- Castiga fuertemente el choque

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

38

## Elementos del RL: Función de Valor

### Función de valor (*value function*)

- Es una función que indica lo que es bueno a largo plazo.
- No es inmediatista como la recompensa.
- El valor de un estado es el valor total de recompensa que un agente puede esperar acumular en el futuro comenzando desde ese estado.
- **OJO! ... pasar a un estado de alto valor puede proveer una recompensa de bajo valor y viceversa!! ... ¿Qué es mejor?**
- Recompensas – decisor primario
- Valores – decisor secundario.

Feb-Jun 2005

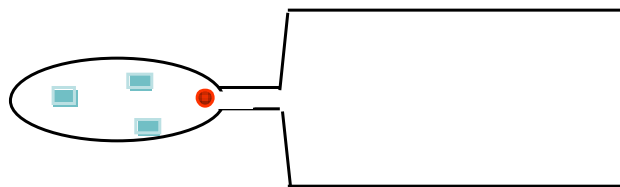
RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

39

## Elementos del RL: Función de Valor

(2)


- Nuestro agente deberá basar sus decisiones en los valores y no en las recompensas inmediatas
- Ejemplo1: robot que evade obstáculos. Robot al que se le castiga por nivel de voltaje en los sensores



Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

40



## Elementos del RL: Función de Valor

### (3)


---

- Es más complejo calcular valores que recompensas.
- La recompensa la da el ambiente.
- Los valores deben ser obtenidos mediante la experiencia.
- Sutton y Barto: *“The most important component of almost all reinforcement algorithms is a method for efficiently estimating values”*
- (\*) Existen otros métodos de RL que no se centran en la estimación de las funciones de valor sino que hayan directamente la política óptima, generalmente por métodos de búsqueda como los GA.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

41



## Elementos del RL: Función de Valor

### (4)

---

- Existen dos tipos de funciones de valor,
- Función de valor de estado: indica el valor esperado de la recompensa que se puede acumular desde ese estado.

$$V: S \rightarrow \mathfrak{R}$$

$$s_t \rightarrow v = V(s_t)$$

- Función de valor del par estado-acción: indica el valor esperado de la recompensa que se puede acumular partiendo del estado y tomando la acción.


$$Q: S \times A \rightarrow \mathfrak{R}$$

$$(s_t, a_t) \rightarrow q = Q(s_t, a_t)$$


Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

42



## Elementos del RL: Función de Valor (5)




... otros ejemplos


- Ejemplo2: algunos perros son capaces de pasar cercas electrificadas, alambres de púas, ríos, y toda suerte de experiencias no placenteras para llegar a una **perra en calor**.
- Ejemplo3: Un corredor de fondo (maratonista).
- Ejemplo 4: Una carrera profesional.
- Ejemplo5: Un boxeador!

**En fin ... las funciones de valor son ciertamente importantes**

Feb-Jun 2005 RL. Diplomado en Computación Inteligente. UPB. jeronimocm@yahoo.com 43



## Elementos del RL: El Modelo

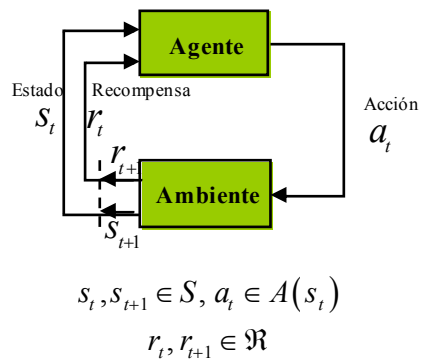


- Es opcional, e.g. no es necesario para todos los algoritmos.
- Modela el comportamiento del ambiente
- Utilizados para **planear** (*planning algorithms*)
- Utilizados para **acelerar** el aprendizaje ... aprendizaje con modelo sin esperar *trial – error*.
- Incorporación “nueva” al RL ... se verá algo de esto avanzado este módulo.

Feb-Jun 2005 RL. Diplomado en Computación Inteligente. UPB. jeronimocm@yahoo.com 44

## ... Elementos del RL (2)

- Los elementos del RL interactúan así:



- Primero están el agente y el ambiente
- El agente percibe el estado y la recompensa en el instante  $t$
- De acuerdo a esto el agente toma un acción  $a_t$
- Ante la acción el ambiente responde con un nuevo estado y una nueva recompensa

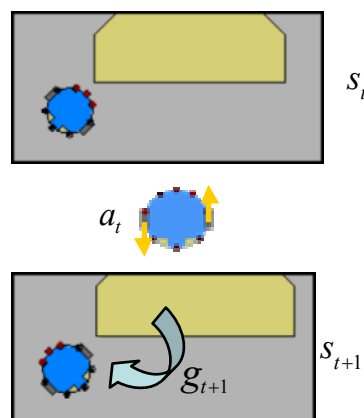
Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

45

## Elementos del RL (3)

- Ejemplo 1:



Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

46

## Elementos del RL (4)

- ¿Qué es RL?
- Se puede ahora hacer una definición desde el **objetivo del RL**

**El objetivo del agente es maximizar la recompensa total recibida a largo plazo.**

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

47

## Elementos del RL: Ejemplo

### El *tricky*

- Es un juego bastante conocido
- Nos servirá para introducir conceptualmente muchas ideas propias del RL: relación *función de valor* – *política*, *greedy*, exploración vs. explotación, *back ups*, ...

X	X	O
X	O	O
		O

**Suposición:** Oponente no profesional  
... alguien que sepa jugar *tricky* y se a  
atento puede NUNCA PERDER.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

48

## Elementos del RL: *Tricky* (2)

### Otros métodos de solución

- *Minimax*: de la teoría de juegos (equilibrio de Nash y demás). Es una opción en la que ambos jugadores deben conocer la estrategia de juego del contrario.
- Optimización clásica, *dynamic programming*: necesita un modelo completo de probabilidades del oponente (**más sobre esto adelante**).

Nota: el modelo puede ser aproximado ... similar a lo que hace RL.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

49

## Elementos del RL: *Tricky* (3)

### ... Otros métodos de solución

- Algoritmos genéticos: para explorar la mejor política de juego, esto es, la mejor movida ante cualquier configuración de X y O en el tablero.

¿Cuál sería la función de *fitness* para cada individuo?

Con este acercamiento la evaluación de cada política tarda muchos juegos para poder obtener un estimado no sesgado ... en exeso lento.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

50

## Elementos del RL: *Tricky* (3)

### Mediante RL

- **Conjunto de estados:** serían todas las posibles configuraciones del tablero de juego.
- Observe que no cualquier configuración pertenece al conjunto de estados:

X	X	X
X	X	X
X	X	X

X	X	X
X	O	X
X	X	X

X	O	X
		O
	X	X

Se podrá saber el número de estados? ... **Pregunta abierta**

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

51

## Elementos del RL: *Tricky* (4)

- **Conjunto de acciones:** una acción significa donde se pone la siguiente X o la siguiente O.
- ¿Cuál será la cardinalidad de  $A$ ?  $|A| = 9$
- Note que el conjunto de acciones disponibles en cada estado es diferente ...

$$A(s_{t_1}) \neq A(s_{t_2})$$

O	O	O
O	O	O
O	O	O

O	O	X
O	O	X
O	O	O

X	O	X
O	O	X
O	X	O

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

52



## Elementos del RL: *Tricky* (5)

- **Recompensa:** valor que se recibe luego de cada movida.
- Exigente:
  - -1: cada jugada (que no gane o pierda)... ¿con que fin?
  - -2: cada que pierda o empate
  - +1: cuando gane
- Medio:
  - 0: cada jugada (que no gane o pierda)
  - -1: cada que pierda o empate
  - +1: cuando gane

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

53



## Elementos del RL: *Tricky* (6)

- Suave:
  - 0: cada jugada (que no gane o pierda)
  - 0: cada que empate
  - -1: cada que pierda
  - +1: cuando gane
- Existen muchas formas de definir la recompensa ...
- Algunos dicen que la definición de la recompensa no debería contener mucha información, e.g., no dedicarle mucho tiempo por la misma naturaleza cualitativa de esta medida.
- Otros dicen que es la mejor manera de lograr soluciones exitosas (**estoy de acuerdo**)

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

54

## Elementos del RL: *Tricky* (7)

### FUNCIONAMIENTO

- Se crea una tabla con un *valor de estado* para cada posible estado.
- La tabla es la *función de valor*.
- Este valor es un estimativo de la probabilidad de ganar partiendo de ese estado.
- Ajustar los valores dentro de esa tabla será la tarea del aprendizaje.
- El ajuste de estos valores se hace mediante *trial and error!! ...* perdiendo y ganando muchas veces

Feb-Jun 2005

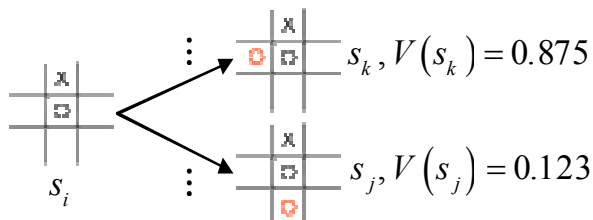
RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

55

## Elementos del RL: *Tricky* (8)

### La política

- Puede ser definida explícitamente como un mapeo entre estados y acciones (o como una función estocástica).
- Enfoque tradicional: la política es regida por la función de valor.
- Ilustración:



Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

56

## Elementos del RL: *Tricky* (9)

- La política entonces dirá que es mejor ir al estado  $s_k$ .
- Esta acción se denomina acción *greedy* (avara) y cuando el agente toma siempre esta acción se dice que actúa *greedily* o que está **explotando**.
- Si se toma una acción *non-greedy* se dice que está explorando.

¿¿Debe el agente seleccionar siempre la acción *greedy*?



**DILEMA EXPLOITATION - EXPLORATION**

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

57

## Elementos del RL: *Tricky* (10)

- **Explotar:** utilizar el conocimiento adquirido para tomar las mejores acciones conocidas hasta ese momento.
- **Explorar:** tomar una acción diferente a la mejor conocida para buscar opciones que puedan resultar mejores.

### Dilema:

El agente debe explotar lo que conoce para obtener buenas recompensas, pero también debe explorar para realizar una mejor selección de las acciones en el futuro.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

58



## Elementos del RL: *Tricky* (11)

- El dilema exploración – explotación es muy trabajado
- Ejemplo ... como hace el perro para darse cuenta por fin que queremos que haga sus necesidades en el sanitario!!!
- **Más adelante** veremos algunos métodos para **balancear** estas dos formas de actuar.
- **Note: con este acercamiento la política se deriva directamente de los valores de los estados.**

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

59



## Elementos del RL: *Tricky* (12)

- **¿Como mantener una correcta estimación de los valores de los estados?**
- Es uno de los principales problemas que buscan seleccionar las diferentes técnicas de RL. **VER**
- No es tan fácil ... para pensar: ¿¿Quién tiene la culpa?? – *the credit assignment problem*. **VER**
- Ideas??

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

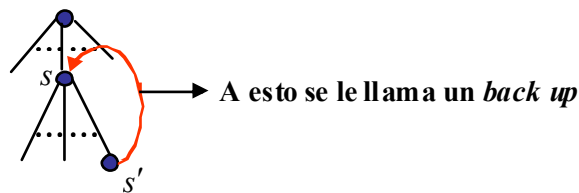
60

## Elementos del RL: *Tricky* (13)

- Una de las opciones consiste en aproximar el valor del estado anterior al valor del estado al que se llegó:

$$V(s) = V(s) + \alpha \cdot [V(s') - V(s)], \alpha: \text{paso}$$

$s$ : estado anterior,  $s'$ : estado actual



Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

61

## Elementos del RL: *Tricky* (14)

### Comentarios

- Este ejemplo sencillo permite observar como funcionan los elementos del RL.
- Permite entender la diferencia con otros posibles acercamientos.
- Note que para cambiar la política no se tiene que esperar “generaciones”.
- Con este modelo de aprendizaje se puede lograr que el agente planee ... que ponga trampas con doble jugada!!

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

62



## Elementos del RL: *Tricky* (15)

- En este caso no se ven jugadas que reporten mala recompensa en pro de un efecto final satisfactorio.
- Existen ejemplos exitosos de RL aplicado a juegos ... El TD – Gammon de Tesauro con  $10^{20}$  estado.
- Para estos ejemplos se hace obligatorio el uso de técnicas de generalización, p.e. **redes neuronales**.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

63



## Elementos del RL: *Tricky* (15)

### Para pensar (Sutton y Barto)

- ¿Qué sucederá si el agente juega contra si mismo?
- ¿Se puede tomar ventaja de la simetría del *tricky*?
- ¿Quién tendrá mejores estimativos del valor de los estado, un “explorador” o un “explotador”?

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

64

# Contenido

- Introducción
- Elementos del RL
- **Acercamiento al RL**
- El problema del RL

Métodos elementales

- *Dynamic Programming*
- *Monte Carlo Methods*
- *Temporal Difference learning*

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

65

## Acercamiento al RL

- Problema incompleto de RL: no asociativo y sin tener que enfrentar el “problema de la asignación de crédito” (*credit assignment problem*).
- Problema: *n-armed bandit*
- Analizaremos la diferencia entre **evaluar** e **instruir**
- Diferencia entre aprender con un **supervisor** y aprender con un **crítico**.
- Veremos valores de acción (más útiles que los valores de estado).
- Métodos de balance de exploración – explotación.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

66

## Acercamiento al RL: Definiciones

- Problema no asociativo: al agente sólo se le presenta **una** situación (estado) y el debe escoger una acción entre varias posibles.
- Problema de la asignación de crédito: ¿cómo saber cual acción en cual estado produjo la recompensa que ahora estoy obteniendo?
- Por ahora ... luego de tomada la acción se obtiene la recompensa y terminó el “episodio”.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

67

## Acercamiento al RL: Definiciones (2)

### *n-armed bandit problem*

- Es un problema no asociativo que no enfrenta el problema de asignación de crédito.
- Máquina traga monedas con  $n$  palancas.
- Estado: solo uno, la maquina con las palancas.
- Acciones:  $n$ , ¿Cuál palanca se opera?



Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

68

## Acercamiento al RL: Definiciones (3)

- En cada jugada se debe seleccionar una palanca
- El agente debe aprender cuales son las palancas más ganadoras y jugar éstas más.
- Problema no estacionario: la distribución de probabilidad con la cual la máquina opera no cambia en el tiempo.
- **Evaluativo:** (*evaluative feedback*) la información del agente es cualitativa ... no sabe que es lo mejor.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

69

## Acercamiento: Valores de Acción

- Representa la calidad de tomar la acción.
- Notación:

$Q^*(a)$ : calidad real de la acción  $a$

$Q_t(a)$ : calidad estimada en la jugada  $t$

- Aproximación obvia – **Promedio:**

$$Q_t(a) = \frac{\sum_{i=1}^{k_a} r_i}{k_a}, k_a : \text{veces que se ha seleccionado } a \text{ antes}$$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

70

## Acercamiento: Valores de Acción (2)

- **Ley de los números grandes (Law of the large numbers):** a medida que el número de muestras crece el promedio muestral  $\mu$  se aproxima al promedio real  $\mu^*$ , esto es:

$$Q_i(a) \Big|_{\infty} = \lim_{k_a \rightarrow \infty} \frac{\sum_{i=1}^{k_a} r_i}{k_a} = Q^*(a)$$

- Esto asegura que si cada acción es seleccionada un número **infinito** de veces se asegura la convergencia de este método a los valores reales!!

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

71

## Acercamiento: Valores de Acción (3)

- Para implementación incremental,
- Sea  $Q_k$  el estimado en la iteración  $k$  luego de  $k$  visitas.
- Ante una nueva recompensa en la iteración  $k+1$ , el valor será:

$$Q_{k+1} = \frac{\sum_{i=1}^{k+1} r_i}{k+1} = \frac{k \cdot \frac{\sum_{i=1}^k r_i}{k} + r_{k+1}}{k+1} = \frac{k \cdot Q_k + r_{k+1}}{k+1}$$

$$Q_{k+1} = \frac{1}{k+1} (k \cdot Q_k + r_{k+1})$$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

72

## Acercamiento: Valores de Acción (4)

- La expresión se puede acomodar para que se parezca a las ecuaciones de actualización con tasa de aprendizaje:

$$Q_{k+1} = \frac{1}{k+1}((k+1) \cdot Q_k - Q_k + r_{k+1})$$

$$Q_{k+1} = Q_k + \frac{1}{k+1}(r_{k+1} - Q_k)$$

Nuevo estimado      Viejo estimado      "Tasa de aprendizaje"      Objetivo, target

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

73

## Acercamiento: Balance EE

- Recuerden el dilema de Exploración – Explotación.
- Existen muchas formas de realizar este balance ...
- Veremos las principales formas de hacerlo.
- Este balance es vital para lograr un proceso de aprendizaje exitoso.
- Las más sencillas: política **greedy** (cero balance) y la política **aleatoria**.
- ¿Por qué no utilizar siempre la acción **greedy**?

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

74

## Acercamiento: Balance EE (2)

### Política $\epsilon$ -greedy

- Es una política estocástica.
- Sencilla, muy conocida y utilizada
- Asigna a la acción *greedy* un probabilidad  $1 - \epsilon$  de ser seleccionada, así:

$$\pi(s, a) = \pi(a) = \begin{cases} 1 - \epsilon & a = a_{\text{greedy}} \\ \frac{\epsilon}{|A(s)| - 1} & a \neq a_{\text{greedy}} \end{cases} \quad \text{con } a_{\text{greedy}} = \arg \max_a (Q(a)) \text{ y } |A| = n$$

**Nota:** por ahora estas ecuaciones no dependen del estado  $s$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

75

## Acercamiento: Balance EE (3)

### Política softmax:

- $\epsilon$ -greedy asigna a las acciones *non - greedy* la misma probabilidad de ser escogidas.
- *Softmax* consiste en asignar a cada acción una probabilidad diferente dependiendo de la calidad de cada una.
- El método más típico: distribución de Boltzmann o de Gibbs:

$$\pi(a) = \frac{e^{\frac{Q(a)}{\tau}}}{\sum_{a' \in A} e^{\frac{Q(a')}{\tau}}}$$

$\tau$  : temperatura, entre más baja crea mayor diferenciación.

$\tau \rightarrow 0$ : política a *greedy*.

$\tau \rightarrow \infty$ : política aleatoria

**Nota:** se prefiere  $\epsilon$ -greedy ... más fácil ajustar  $\epsilon$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

76

## Acercamiento: Balance EE (3)

### Valores iniciales optimistas (*optimistic initial values*)

- Es un método sencillo
- Consiste iniciar los valores iniciales en un valor alto para favorecer la exploración inicial ... **¿por qué?**
- Se dice que este método está sesgado (*biased*) por la inicialización.
- Es una forma de introducir conocimiento previo al problema (*prior knowledge*) ... mediante otros métodos no es tan sencillo.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

77

## Acercamiento: Balance EE (4)

### Exploración dirigida

- Tema muy interesante pero que agrega bastante complejidad al problema
- Se busca que el espacio de pares estado – acción sean explorados de manera óptima según algún criterio de desempeño.
- Basado en contador (*counter-based*): consiste en llevar un contador de visitas a los pares estado acción,

$L_s^a$ : número de veces que en el estado  $s$  se tomó la acción  $a$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

78

## Acercamiento: Balance EE (5)

### ... Exploración dirigida

- Al momento de tomar una acción, la probabilidad de que esta sea seleccionada dependerá de:

$$\frac{L_s^a}{\sum_{a' \in A(s)} L_s^{a'}}$$

- Con esto se logra que los pares estado – acción menos visitados tengan mayor probabilidad de ser seleccionados
- Método muy utilizado en técnicas basadas en modelo (*model based*)

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

79

## Acercamiento: Ejemplo *n-Armed Bandit*

- Suponga un 10-armed bandit.
- La recompensa de cada palanca se selecciona de una distribución normal con:

$$\mu = Q^*(a), \quad \sigma = 1$$

- Con  $0 \leq Q^*(a) \leq 1$
- Se simulan 1000 jugadas de 7 diferentes agentes
- Luego se les dan 200 intentos para que jueguen explotando
- A continuación gráficas de desempeño

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

80

## Acercamiento: *n*-Armed Bantit (2)

### Agentes:

- Negro: *greedy*
- Azul:  $\epsilon$ -*greedy* con  $\epsilon = 0.01$
- Verde:  $\epsilon$ -*greedy* con  $\epsilon = 0.1$
- Rojo: Boltzmann con  $\tau = 0.8$
- cyan: Boltzmann con  $\tau = 0.4$
- amarillo: valores optimistas y  $\epsilon$ -*greedy* con  $\epsilon = 0$
- Magenta: valores optimistas y  $\epsilon$ -*greedy* con  $\epsilon = 0.1$
- Negro punteado: política aleatoria

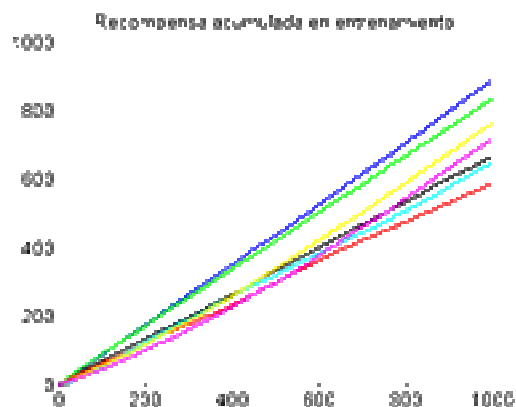
Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

81

## Acercamiento: *n*-Armed Bantit (3)

- Gráfica de la recompensa que obtienen durante el aprendizaje



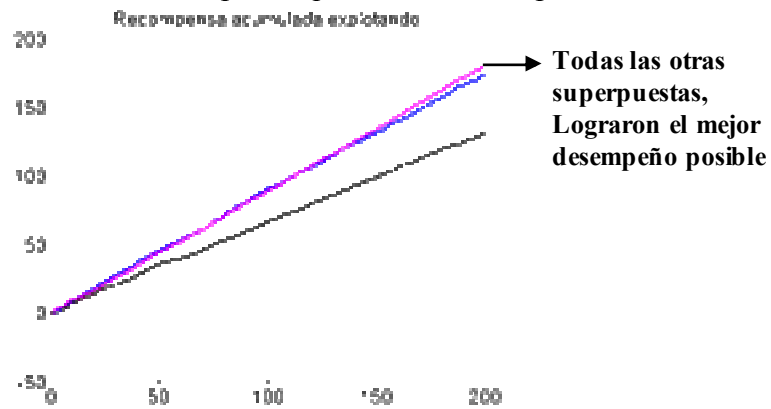
Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

82

## Acercamiento: $n$ -Armed Bantit (4)

- Gráfica de la recompensa que obtienen al explotar



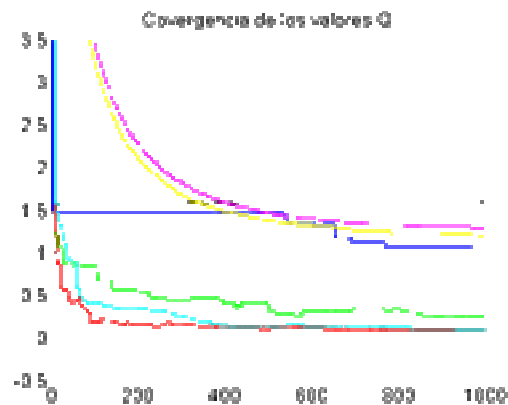
Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

83

## Acercamiento: $n$ -Armed Bantit (5)


- Convergencia de los valores  $Q$  aprendidos respecto a los reales:




Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

84



## Acercamiento: *n*-Armed Bantit (6)



### Comentarios

- VER COD MATLAB
- Para un análisis más detallado (promediado sobre 2000 diferentes pruebas) **ver libro de Sutton y Barto capítulo 2.**
- ¿Por qué el *greedy* no obtiene la máxima recompensa?
- ¿Por qué el 0.01-*greedy* obtiene la mayor recompensa en entrenamiento y no en la explotación?
- ¿Por qué las convergencias de  $Q$ ? ¿Por qué la de los valores optimistas no convergen? ¿Afecta esto la explotación?

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

85



## Acercamiento: Métodos no Basados en Valores de Acción



- **HABLAR DE:  
REINFORCEMENT COMPARISON  
PERSUIT METHODS**

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

86

# Contenido

- Introducción
- Elementos del RL
- Acercamiento al RL
- **El problema del RL**

Métodos elementales

- *Dynamic Programming*
- *Monte Carlo Methods*
- *Temporal Difference learning*

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

87

# El problema del RL

- Definiciones
- Retorno
- Propiedad de Markov
- MDPs
- Funciones de Valor
- Funciones de Valor Óptimas
- Ecuación de Bellman
- Solución

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

88

## Problema del RL: Definiciones

### No estacionario

- La distribución de probabilidad de los eventos cambia en el tiempo.
- La estrategia del oponente en *tricky* puede cambiar.
- La probabilidad de ganar en cada palanca del *n-armed bandit* cambia en el tiempo.
- Los sensores de un robot se degradan en el tiempo (rápidamente)
- En últimas ... el ambiente (su percepción) varía en el tiempo

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

89

## Problema del RL: Definiciones (2)

- **¿Cómo seguir un problema no estacionario?**
- No se puede “confiar” en los promedios

$$Q_{k+1} = Q_k + \frac{1}{k+1}(r_{k+1} - Q_k)$$

- Se le debe dar mayor importancia a lo aprendido recientemente.
- Una idea: utilizar una tasa de aprendizaje constante

$$Q_{k+1} = Q_k + \alpha \cdot (r_{k+1} - Q_k), \quad 0 < \alpha \leq 1$$

- **¿Por qué?**

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

90

## Problema del RL: Definiciones (3)

- Observe:

$$Q_k = Q_{k-1} + \alpha \cdot (r_k - Q_{k-1})$$

$$Q_k = \alpha \cdot r_k + (1-\alpha)Q_{k-1} = \alpha \cdot r_k + (1-\alpha)[\alpha \cdot r_{k-1} + (1-\alpha)Q_{k-2}]$$

$$Q_k = \alpha \cdot r_k + (1-\alpha)\alpha \cdot r_{k-1} + (1-\alpha)^2 Q_{k-2}$$

$$Q_k = \alpha \cdot r_k + (1-\alpha)\alpha \cdot r_{k-1} + (1-\alpha)^2 \cdot \alpha \cdot r_{k-2} + (1-\alpha)^3 \cdot Q_{k-3}$$

$\vdots$

$$Q_k = (1-\alpha)^k \cdot Q_0 + \sum_{i=1}^k (1-\alpha)^{k-i} \cdot \alpha \cdot r_i$$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

91

## Problema del RL: Definiciones (4)

- Note que la expresión anterior representa un promedio ponderado!!

$$\begin{aligned} (1-\alpha)^k + \sum_{i=1}^k (1-\alpha)^{k-i} \cdot \alpha &= (1-\alpha)^k + \alpha \cdot \sum_{i=1}^k (1-\alpha)^{k-i} \\ &= (1-\alpha)^k + \alpha \cdot [(1-\alpha)^{k-1} + (1-\alpha)^{k-2} + \dots + (1-\alpha) + 1] \\ &= (1-\alpha)^k + \alpha \cdot \left[ \frac{(1-\alpha)^k - 1}{(1-\alpha) - 1} \right] = (1-\alpha)^k + \alpha \cdot \left[ \frac{(1-\alpha)^k - 1}{-\alpha} \right] \\ &= (1-\alpha)^k + 1 - (1-\alpha)^k \\ &= 1 \end{aligned}$$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

92

## Problema del RL: Definiciones (5)

- **Utilizar una tasa de aprendizaje constante equivale a realizar un promedio ponderado!!**
- De esta forma una recompensa recibida  $k_t$  instantes de tiempo atrás será ponderada por:  $\alpha \cdot (1 - \alpha)^{k - k_t}$
- Mientras más atrás menor será su efecto en el promedio ...
- $0 < 1 - \alpha \leq 1$  “*exponential, recency-weighted average*”
- ¿Que pasa si  $\alpha = 1$ ?
- ¿Que pasa si  $\alpha = 0$ ?

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

93

## Problema del RL: Definiciones (6)

- Se acostumbra decaer la tasa de aprendizaje con las iteraciones
- Para garantiza convergencia la tasa de aprendizaje deben satisfacer:

$$\sum_{k=1}^{\infty} \alpha_k = \infty, \quad \sum_{k=1}^{\infty} \alpha_k^2 < \infty$$

- Observe que si  $\alpha_k = \frac{1}{k}$  se satisfacen las condiciones
- En este caso es un promedio, que por la ley de los “números grandes” converge

Feb-Jun 2005

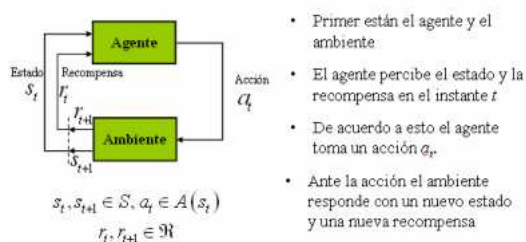
RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

94

## Problema del RL: Definiciones (7)

### Asociativo

- La calidad de las acciones se deben asociar con la situación (o el estado). Debe aprender a asociar situación – acción – placer.
- Recuerde el modelo del aprendizaje por refuerzo:



- Primer están el agente y el ambiente
- El agente percibe el estado y la recompensa en el instante  $t$
- De acuerdo a esto el agente toma una acción  $a_t$
- Ante la acción el ambiente responde con un nuevo estado y una nueva recompensa

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

95

## Problema del RL: Definiciones (8)

### Tareas Episódicas vs. Continuas

- Episódicas: noción natural de tiempo final
- Jugadas de *tricky*.
- Continuas: tareas que no tienen un estado final. Continúan indefinidamente.
- Tareas en robótica que incluyen toda la vida del robot.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

96

## Problema del RL: Definiciones (9)

### Problema de la asignación de crédito

- Problema de la asignación de crédito: la recompensa obtenida no depende sólo de la última decisión tomada
- Minsky, 1961: *how do you distribute credit for success among the many decisions that may have been involved in producing it?*

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

97

## Problema del RL: Retorno

- Recuerde: **el objetivo del agente es maximizar la recompensa que recibe en el largo plazo.**
- Formalmente el agente trata de maximizar el *retorno esperado* a partir de el tiempo  $t$ .
- Si a partir de este tiempo el agente recibe una secuencia de recompensas:  $r_{t+1}, r_{t+2}, \dots, r_T$
- Una primera forma de definir el retorno es:

$$R_t = r_{t+1} + r_{t+2} + \dots, r_T = \sum_{k=0}^{T-(k+1)} r_{t+k+1}$$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

98

## Problema del RL: Retorno (2)

- ¿Qué pasa en tareas continuas? – el retorno tiende a infinito!
- Por eso se introduce el **factor/rata de descuento  $\gamma$** .
- El agente selecciona la acción  $a_t$  para maximizar:

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, \quad 0 \leq \gamma \leq 1$$

- $\gamma$  **determina la importancia en el presente de recompensas futuras.**
- Una recompensa obtenida  $k$  pasos después valdrá  $\gamma^{k-1}$  veces lo que valdría en el presente.
- Si  $\gamma$  es menor que 1 y los valores de  $r$  están acotados ...  $R_t$  converge

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

99

## Problema del RL: Retorno (3)

- Es un acercamiento a la solución del *problema de asignación de crédito*.
- Si  $\gamma = 0$ : el agente es miope!!
- Si en una determinada tarea cada acción afecta sólo la recompensa inmediata ... entonces un agente miope podría funcionar.
- Entre más cercano a 1 el factor de descuento más “planeador” será el agente ... más tratará de lograr recompensa futura con su presente acción! ... ¿Es siempre eso deseable?

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

100

## Problema del RL: Retorno (4)

- Para tareas episódicas se puede utilizar:

$$R_t = r_{t+1} + r_{t+2}, \dots, r_T = \sum_{k=0}^{T-(k+1)} r_{t+k+1}$$

- Para tareas continuas se tiene que utilizar la suma descontada.
- La suma descontada es más general, Si
  - $\gamma = 1$  y
  - Si en tareas episódicas se crea un estado de *absorción*

Se puede trabajar sólo con:

$$R_t = \sum_{k=0}^{\infty} \gamma^k \cdot r_{t+k+1}$$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

101

## Problema del RL: Retorno (5)

- **Ejemplo1:** péndulo invertido: objetivo: no dejar caer el péndulo y no salirse de la pista

Tomado de: SUTTON, Richard. y BARTO, Andrew., Reinforcement Learning: An Introduction.

- ¿Como se podría definir la recompensa? ¿Cómo queda el retorno?

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

102

## Problema del RL: Retorno (6)

- Episódica.  $r = +1$  a cada paso de simulación con  $\gamma = 1$ .
- Continua (con descuento).  $r = -1$  cuando se caiga.

**Con ambas definiciones al maximizar el retorno esperado desde cada estado se estará maximizando el tiempo que el péndulo se mantienen erguido**

- **Ejemplo2:** Pensar en recompensa para salir de un laberinto.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

103

## Problema del RL: Propiedad de Markov

- **Informalmente:** Una señal tiene la *propiedad de Markov* si contiene (retiene) la información suficiente y necesaria para tomar una decisión

**Ejemplo:** La señal de estado  $s$ .

- La señal de estado puede ser más que medidas sensoriales inmediatas.
- Puede hacer cálculos que incluyan valores pasados, p.e. la velocidad, la aceleración tomando medidas de distancia.
- Así una señal de estado puede ser lo suficientemente completa.
- ... un humano puede barrer con la mirada y luego recordar algunos objetos relevantes que observó.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

104

## Problema del RL: Propiedad de Markov (2)

- OJO! ... tampoco se trata de proveer al agente con más información de la necesaria.

### Formalmente:

- Considere tanto  $S$  como  $\hat{R}$  conjuntos discretos.
- En el caso general causal: el estado y la recompensa siguiente depende de todo lo sucedido antes:

$$P_r = P\{s_{t+1} = s', r_{t+1} = r' \mid s_t, a_t, s_{t-1}, a_{t-1}, \dots, s_0, a_0\}$$

- Si se posee la propiedad de Markov:

$$P_r = P\{s_{t+1} = s', r_{t+1} = r' \mid s_t = s, a_t = a\}$$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

105

## Problema del RL: Propiedad de Markov (3)

- Una tarea posee la propiedad de Markov si y sólo si:

$$P\{s_{t+1} = s', r_{t+1} = r' \mid s_t, a_t, \dots, s_0, a_0\} = P_r\{s_{t+1} = s', r_{t+1} = r' \mid s_t, a_t\} \\ \forall s', r' \wedge s_t, a_t, s_{t-1}, a_{t-1}, \dots, s_0, a_0$$

- Esta propiedad es MUY importante en RL ... aunque puede no estar satisfecha por completo!
- Ejemplo: péndulo invertido. Estado: posición y velocidad del carrito, ángulo y velocidad angular del péndulo.

**¿Cumplirá la propiedad de Markov?**

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

106

## Problema del RL: Procesos de Decisión Markovianos

- MDP: *Markov Decision Process*.
- *Finite* MDP: con espacios de estado y acción finitos ( $A, S$ ).
- La mayoría de los métodos de RL pretenden solucionar un MDP finito *one-step dynamics*.
- En adelante se considera que se trabaja con MDPs finitos.
- Sin embargo las técnicas funcionan aún con pequeñas violaciones a esta propiedad.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

107

## Problema del RL: MDP (2)

### Elementos

- Conjunto de estados  $S$  y de acciones disponibles por estado  $A(s_t)$  finitos  $\forall s_t \in S$
- Conjunto de probabilidades de transición de estado:

$$P_{ss'}^a = P\{s_{t+1} = s' | s_t = s, a_t = a\}$$

- Conjunto de valores esperados de recompensa:

$$R_{ss'}^a = E\{r_{t+1} | s_t = s, a_t = a, s_{t+1} = s'\}$$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

108

## Problema del RL: Función de Valor

- Recuerde: el valor de un estado indica que tan “bueno” es.
- Recuerde: la política  $\pi$  asigna una probabilidad de tomar la acción  $a$  en el estado  $s$
- Formalmente: **el valor del estado  $s$  bajo la política  $\pi$  es el valor esperado del retorno empezando en  $s$  y siguiendo la política  $\pi$**
- Matemáticamente, para MDPs:

$$V^\pi(s) = E_\pi \{ R_t | s_t = s \} = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k \cdot r_{t+k+1} | s_t = s \right\}$$

**Función de valor de estado para la política  $\pi$**

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

109

## Problema del RL: FV (2)

- Análogamente, la función de valor de acción para la política  $\pi$  se define:

$$Q^\pi(s, a) = E_\pi \{ R_t | s_t = s, a_t = a \} = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k \cdot r_{t+k+1} | s_t = s, a_t = a \right\}$$

- Indica “qué tan bueno” es tomar la acción  $a$  en el estado  $s$ , en términos de la cantidad de recompensa que el agente puede esperar obtener a partir de ello.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

110

## Problema del RL: FV Óptima

- Solucionar una tarea de RL: hallar la mejor política
- Mejor política: la que más retorno reporta al agente.
- $V^\pi$  define una relación de orden parcial entre políticas:

$\pi$  es mejor que  $\pi'$  si y solo si  
 $V^\pi(s) > V^{\pi'}(s) \quad \forall s \in S$

- Todas las políticas óptimas están asociadas a la misma función de valor de estado o de acción:

$$V^*(s) = V^{\pi^*}(s) = \max_{\pi} V^{\pi}(s), \quad \forall s \in S$$

$$Q^*(s, a) = Q^{\pi^*}(s, a) = \max_{\pi} Q^{\pi}(s, a), \quad \forall s \in S, a \in A(s)$$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

111

## Problema del RL: FV Óptima (2)

- Valor de acción óptima: valor esperado del retorno al tomar la acción  $a$  en el estado  $s$  y seguir con la política óptima.
- Si al tomar dicha acción se obtiene la recompensa  $r_{t+1}$ :

$$Q^*(s, a) = E \left\{ r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \right\} = E \left\{ r_{t+1} + \gamma V^*(s_{t+1}) \right\} \text{ ¿Por qué?}$$

- Relación entre valor de estado y valor de acción óptima:

$$V^*(s) = \max_{a \in A(s)} Q^*(s, a)$$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

112

## Problema del RL: Ec. de Bellman

- Richard Bellman, 1957.
- Establece la relación entre el valor de un estado y el valor de sus estados sucesores.
- Para una política  $\pi$ , se plantea la siguiente relación recursiva:

$$V^\pi(s) = E_\pi \{ R_t | s_t = s \} = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k \cdot r_{t+k+1} | s_t = s \right\}$$

$$V^\pi(s) = E_\pi \left\{ r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k \cdot r_{t+k+2} | s_t = s \right\}$$

$$V^\pi(s) = \sum_{a \in A(s)} \pi(s, a) \sum_{s' \in S} P_{ss'}^a \cdot \left[ R_{ss'}^a + \gamma E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k \cdot r_{t+k+2} | s_t = s \right\} \right]$$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

113

## Problema del RL: Ec. de Bellman (2)

- ... Finalmente: Ecuación de Bellman para  $V^\pi$ .

$$V^\pi(s) = \sum_{a \in A(s)} \pi(s, a) \sum_{s' \in S} P_{ss'}^a \cdot [R_{ss'}^a + \gamma \cdot V^\pi(s')]$$

- Palabras: el valor del estado  $s$  dependen de la probabilidad de tomar cada acción, de la probabilidad que dicha acción me lleve al estado  $s'$ , de la recompensa esperada en esta transición y del valor del nuevo estado.
- Note que la ecuación de Bellman es un sistema de ecuaciones  $|S| \times |S|$  dadas todas las probabilidades  $(\pi, P, R)$ .

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

114

## Problema del RL: Ec. de Bellman (3)

- ¿Cómo sería la ecuación de Bellman con una política determinística?

$$V^{\pi}(s) = \sum_{s' \in S} P_{ss'}^{\pi(s)} \cdot [R_{ss'}^{\pi(s)} + \gamma \cdot V^{\pi}(s')]$$

- La ecuación de Bellman para  $Q^{\pi}$  es:

$$Q^{\pi}(s, a) = \sum_{s' \in S} P_{ss'}^a \left[ R_{ss'}^a + \gamma \cdot \sum_{a' \in A(s')} \pi(s', a') Q^{\pi}(s', a') \right] \quad (*)$$

- Si la política es determinista:

$$Q^{\pi}(s, a) = \sum_{s' \in S} P_{ss'}^a \left[ R_{ss'}^a + \gamma \cdot Q^{\pi}(s', \pi(s')) \right]$$

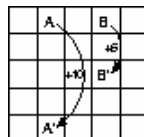
Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

115

## Problema del RL: Ec. de Bellman (4)

- **Ejemplo: Gridworld.** Tomado del libro de Sutton y Barto. En sus palabras: *“The cells of the grid correspond to the states of the environment. At each cell, **four actions are possible**: north, south, east, and west, which **deterministically** cause the agent to move one cell in the respective direction in the grid. Actions that would take the agent off the grid leave its location unchanged, but also result in a reward of -1. Other actions result in a reward of 0, except those that move the agent out of the special states A and B. From state A, all four actions yield a reward of +10 and take the agent to A'. From state B, all actions yield a reward of +5 and take the agent to B'”*



Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

116

## Problema del RL: Ec. de Bellman (5)

- Suponga que en cada estado se seleccionan las cuatro acciones con igual probabilidad y que  $\gamma = 0.9$ .
- La solución del sistema de ecuaciones para los valores de estado es:

3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

- ¿Cuál es el mejor estado?
- ¿Por qué el retorno es menor a 10?
- ¿Por qué el retorno de B sí es mayor a 5?
- ¿Qué política se puede inferir?

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

117

## Problema del RL: Ec. de Bellman (6)

- Una prueba:

3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

→  $s$

→ posibles sucesores  $s'$

$$0.25 \cdot 1[0 + 0.7] + 0.25 \cdot 1[0 - 0.4] + \\ 0.25 \cdot 1[0 + 0.4] + 0.25 \cdot 1[0 + 2.3] = 0.75 \approx 0.7$$

**Problema de cifras decimales**

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

118

## Problema del RL: Ec. de Bellman (7)

- Algunas ecuaciones ...

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

$$V_1 = 0.25 \cdot [(-1 + \gamma V_1) + (-1 + \gamma V_1) + (0 + \gamma V_2) + (0 + \gamma V_6)]$$

$$\Rightarrow V_1 = 0.40909 \cdot [V_1 + V_6 - 2.222]$$

$$V_2 = 0.25 \cdot [4 \cdot (10 + 0.9 \cdot V_{22})]$$

$$V_3 = 0.25 \cdot [0.9 \cdot V_2 + 0.9 \cdot V_4 + 0.9 \cdot V_8 + 0.9 \cdot V_3 - 1]$$

$$V_4 = 0.25 \cdot [4 \cdot (5 + 0.9 \cdot V_{14})]$$

⋮

Hacer otras pruebas

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

119

## Problema del RL: Ec. de Bellman (8)

- También se pueden escribir las ecuaciones de Bellman para las funciones de valor óptimas: *Bellman optimality equation*.
- Para la función de valor de estado:

$$V^*(s) = \max_{a \in A(s)} Q^*(s, a) = \max_{a \in A(s)} E_{\pi^*} \{ R_t | s_t = s, a_t = a \}$$

$$V^*(s) = \max_{a \in A(s)} E \{ r_{t+1} + \gamma V^*(s_{t+1}) | s_t = s, a_t = a \}$$

$$V^*(s) = \max_{a \in A(s)} \sum_{s' \in S} P_{ss'}^a \cdot [R_{ss'}^a + \gamma \cdot V^*(s')]$$

No depende de la política!!

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

120

## Problema del RL: Ec. de Bellman (9)

- Para la función de valor de acción:

$$Q^*(s, a) = E_{\pi^*} \{ R_t | s_t = s, a_t = a \}$$

$$Q^*(s, a) = E \{ r_{t+1} + \gamma V^*(s_{t+1}) | s_t = s, a_t = a \}$$

$$Q^*(s, a) = E \left\{ r_{t+1} + \gamma \max_{a' \in A(s)} Q^*(s_{t+1}, a') | s_t = s, a_t = a \right\}$$

$$Q^*(s, a) = \sum_{s' \in S} P_{ss'}^a \cdot \left[ R_{ss'}^a + \gamma \cdot \max_{a' \in A(s')} Q^*(s', a') \right]$$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

121

## Problema del RL: Ec. de Bellman (10)

### Notas:

- Las ecuaciones de optimalidad son también un sistema  $|S| \times |S|$  pero no lineal (para  $V$ )
- Cuál es la política óptima? – la que asigne probabilidades diferente de cero sólo a las acciones que producen el máximo en la ecuación de optimalidad de Bellman ... **¿No es esto corto plazo?**
- Con los valores  $Q^*$  es aún más fácil ... **¿Cómo?**

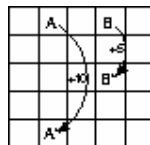
Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

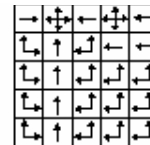
122

## Problema del RL: Ec. de Bellman (11)

- Ejemplo: el mismo *gridworld* anterior:



22	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7



Valores óptimos,  $V^*$

Política inferida

- ¿Por qué son los valores todos positivos y mayores a los de antes?

Feb-Jun 2005

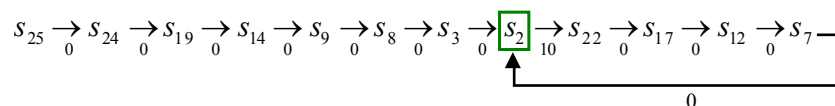
RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

123

## Problema del RL: Ec. de Bellman (12)

- ¿Será que si está bien?
- Recuerde que:  $V^{\pi^*}(s) = E_{\pi^*} \{R_t | s_t = s\}$
- Supongamos que partimos del estado número 25
- Una posible secuencia de estados y recompensas siguiendo la política óptima sería:

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25



- ¿Cómo sería la expresión del retorno?

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

124

## Problema del RL: Ec. de Bellman (13)

- Retorno, partiendo del estado número 25:

$$R_0 = \sum_{k=0}^{\infty} \gamma^k \cdot r_{k+1} = 0 + 0 + 0 + 0 + 0 + 0 + 0 + 10 \cdot \gamma^7 + \dots + 10 \cdot \gamma^{12} + \dots$$

$$R_0 = 10 \cdot \sum_{j=0}^{\infty} \gamma^{7+5j} = 10 \cdot \gamma^7 \cdot \sum_{j=0}^{\infty} (\gamma^5)^j, \quad \text{como } |\gamma| < 1$$

$$R_0 = \frac{10 \cdot \gamma^7}{1 - \gamma^5} = \frac{10 \cdot 0.9^7}{1 - 0.9^5}$$

$$R_0 = 11.6797$$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

125

## Problema del RL: Solución

**Solucionar la ecuación de optimalidad de Bellman es una ruta para solucionar el Problema de RL!!**

**Sin embargo tiene sus problemas:**

- Supone el conocimiento de  $P_{ss'}^a$  y de  $R_{ss'}^a$
- Mucha complejidad computacional (TD-Gammon,  $10^{20}$  estados ... tomaría millones de años hallar una solución)
- Supone que se satisface la propiedad de Markov

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

126

## Problema del RL: Solución (2)

- La primera suposición generalmente no se puede hacer
- La segunda es muy determinante
- La tercera por lo general se fuerza

**Por esto aparecen varios métodos que tratan de resolver estas ecuaciones indirectamente**

- Solución iterativa: *Dynamic programming*
- Solución *trial-and-error* muestreando  $P_{ss'}^a$  y  $R_{ss'}^a$ .

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

127

## Contenido

- Introducción
- Elementos del RL
- Acercamiento al RL
- El problema del RL

Métodos elementales

- ***Dynamic Programming***
- *Monte Carlo Methods*
- *Temporal Difference learning*

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

128



# Dynamic Programming

- Introducción
- Evaluación de la Política
- Mejoramiento de Política
- Iteración de Política
- Iteración de Valor
- DP asincrónica
- GPI
- Conclusiones

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

129



## DP: Introducción

- Programación dinámica (*Dynamic Programming*, DP): concepto introducido por Richard Bellman en 1957.
- Primera relación con RL en 1961 (Minsky)

**Se refiere a una colección de algoritmos que hayan políticas óptimas dado un modelo perfecto del ambiente como un MDP.**

- Su importancia es más teórica que práctica.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

130

## DP: Introducción (2)

- Recordar las ecuaciones de optimalidad:

$$V^*(s) = \max_{a \in A(s)} \sum_{s' \in S} P_{ss'}^a \cdot [R_{ss'}^a + \gamma \cdot V^*(s')]$$

$$Q^*(s, a) = \sum_{s' \in S} P_{ss'}^a \cdot [R_{ss'}^a + \gamma \cdot \max_{a' \in A(s')} Q^*(s', a')]$$

- DP básicamente lo que trata es convertir estas ecuaciones en **reglas de actualización**, e.g., solución numérica.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

131

## DP: Evaluación de la Política

- Evaluación de la Política – *Policy Evaluation*
- Se refiere a encontrar los valores de estado ante una determinada política:  $V^\pi(s) \forall s \in S$
- Se “evalúa” la ecuación de Bellman para encontrar los valores.
- También se le llama: problema de predicción (*prediction problem*) ... se toma una política y se predicen los valores de estado.
- Evaluar la política equivale a solucionar la ecuación de Bellman:

$$V^\pi(s) = \sum_{s' \in S} P_{ss'}^{\pi(s)} \cdot [R_{ss'}^{\pi(s)} + \gamma \cdot V^\pi(s')]$$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

132

## DP: Evaluación de la Política (2)

- La solución se hace iterativamente, según:

$$V_0(s) \text{ arbitrario} \quad a.$$

$$V_{k+1}(s) = \sum_{a \in A(s)} \pi(s, a) \sum_{s' \in S} P_{ss'}^a \left[ R_{ss'}^a + \gamma \cdot V_k(s') \right] \quad \forall s \in S \quad b.$$

- Se llama: *iterative policy evaluation*
- Converge a  $V^\pi$  si  $\gamma < 1$  o si se alcanza un estado terminal bajo cualquier política.
- Se termina cuando:

$$\max_{s \in S} |V_{k+1}(s) - V_k(s)| \leq tol$$

Feb-Jun 2005

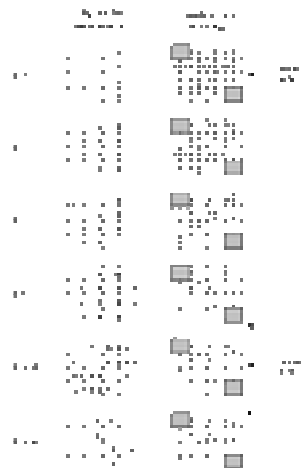
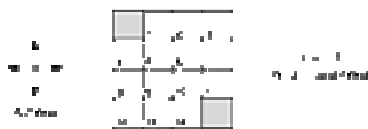
RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

133

## DP: Evaluación de la Política (3)

### Ejemplo: *gridworld*

- Sutton y Barto, sección 4.1.
- Gridworld 4x4. Dos estados terminales, 14 no terminales.
- Todas las transiciones:  $r = -1$



Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

134

## DP: Evaluación de la Política (4)

- Solución final:

0	-14	-20	-22
-14	-18	-20	-20
-20	-20	-18	-14
-22	-20	-14	0

VER COD. MATLAB

¿Por qué esos valores?

¿Cuáles serían los valores si se sigue una política óptima?

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

135

## DP: Mejoramiento de Política

- Mejoramiento de Política – *Policy Improvement*
- El objetivo de evaluar la política es hallar mejores políticas

### Teorema del mejoramiento

- Sean dos políticas determinísticas diferentes:  $\pi$  y  $\pi'$ .
- Si  $Q^\pi(s, \pi'(s)) \geq V^\pi(s)$
- Entonces:  $V^{\pi'}(s) \geq V^\pi(s)$

→ Es decir, la política nueva es mejor o igual a la anterior!!

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

136

## DP: Mejoramiento de Política (2)

- El mejoramiento de la política se lleva a cabo utilizando la política *greedy*.

$$\pi'(s) = \arg \max_a Q^\pi(s, a) \quad a.$$

$$\pi'(s) = \arg \max_a E \{ r_{t+1} + \gamma \cdot V^\pi(s_{t+1}) | s_t = s, a_t = a \} \quad b.$$

$$\pi'(s) = \arg \max_a \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma \cdot V^\pi(s')] \quad c.$$

- El mejoramiento de la política siempre produce una política  $\pi'$  igual o mejor a la anterior.

$$Q^\pi(s, \arg \max_a Q^\pi(s, a)) = \max_a Q^\pi(s, a) \geq V^\pi(s)$$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

137

## DP: Mejoramiento de Política (3)

- ¿Cuándo es igual?

$$V^\pi(s) = \max_a Q^\pi(s, a)$$

$$V^\pi(s) = \max_a \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma \cdot V^\pi(s')]$$



**Ecuación de Optimalidad de Bellman!! VER**

- Por lo tanto: **con el mejoramiento siempre se logrará un política mejor y sólo será igual cuando se haya llegado a la óptima!**

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

138

# DP: Iteración de Política

- Iteración de política – *Policy Iteration*
- Se utiliza para hallar la política óptima en DP.
- Alterna procesos de evaluación (E) y de mejoramiento (M) partiendo de una política inicial cualquiera  $\pi_0$ .
- El proceso continúa hasta encontrar la política óptima y su correspondiente valor.

$$\pi_0 \xrightarrow{E} V^{\pi_0} \xrightarrow{M} \pi_1 \xrightarrow{E} V^{\pi_1} \xrightarrow{M} \pi_2 \xrightarrow{E} \dots \xrightarrow{M} \pi^* \xrightarrow{E} V^*$$

- Por lo general  $V^{\pi_{i-1}}$  se emplea como valor inicial para la evaluación de  $V^{\pi_i}$ . Esto acelera bastante la convergencia

Feb-Jun 2005

RL, Diplomado en Computación Inteligente.  
UPB, jeronimocm@yahoo.com

139

- $$\pi_0 \xrightarrow{E} V^{\pi_0} \xrightarrow{M} \pi_1 \xrightarrow{E} V^{\pi_1} \xrightarrow{M} \pi_2 \xrightarrow{E} \cdots \xrightarrow{M} \pi^* \xrightarrow{E} V^*$$

- Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

139

## DP: Iteración de Política (2)

- Para el ejemplo del *gridworld*: sin descuento ( $\gamma = 1$ ) el episodio se termina en los estados terminales o después de 1000 pasos.
- Iniciando con política determinística: todos hacia arriba
- Estos fueron los valores:

	Valores de Estado	Política
Iteración 1		
Iteración 2		

Feb-Jun 2005

R.L. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

140

- |             | Valores de Estado                              | Política                                       |
|-------------|--|--|
| Iteración 1 | $\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$ | $\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$ |
| Iteración 2 | $\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$ | $\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$ |

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

140

## DP: Iteración de Política (3)

... continuando:

	Valores de Estado	Política
Iteración 3	0 -1 -2 -3	0 1 1 1
	-1 -2 -3 -2	4 1 1 3
	-2 -3 -2 -1	4 1 2 3
	-3003 -2 -1 0	2 2 2 1
Iteración 4	0 -1 -2 -3	0 1 1 1
	-1 -2 -3 -2	4 1 1 3
	-2 -3 -2 -1	4 1 2 3
	-3 -2 -1 0	2 2 2 1

- ¿Por qué valores tan altos en las iteraciones?
- ¿Por qué los valores finales?

**VER COD. MATLAB**

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

141

## DP: Iteración de Valor

- Iteración de Valor – *Value Iteration*
- El proceso de mejoramiento de política puede ser muy largo  
... en cada paso se debe evaluar por completo la política.
- Consiste en realizar sólo un paso de evaluación e inmediatamente el mejoramiento:

$V_0(s)$  arbitrario

$$V_{k+1}(s) = \max_a \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma \cdot V_k(s')]$$

- Es equivalente a convertir la ecuación de optimalidad de Bellman en una regla de actualización. **VER**

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

142

## DP: DP Asincrónica

- *Asynchronous Dynamic Programming.*
- No se barre todo el espacio de estados.
- Puede barrer los estados de cualquier modo
- Igual los tiene que barrer todos
- Sirve para intercalar con experiencia real.

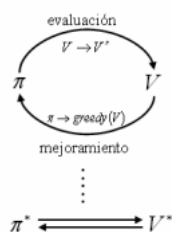
Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

143

## DP: GPI

- Iteración Generalizada de Política – *Generalized Policy Iteration, GPI.*
- Concepto general del intercalamiento entre evaluación y mejoramiento de política.



**La mayoría de los métodos de RL se basan en GPI**

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

144

## DP: Conclusiones

- DP es un método interesante ... soluciona la ecuación de optimalidad de Bellman iterativamente.
- Funciona muy bien para problemas pequeños: *Gridworld* y en los que se conocen  $P_{ss'}^a$  y  $R_{ss'}^a$
- Generalmente NO se conoce el modelo del ambiente!
- DP es costoso computacionalmente.
- DP realiza *bootstrapping*: estimar valores sobre otros estimados

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

145

## Contenido

- Introducción
  - Elementos del RL
  - Acercamiento al RL
  - El problema del RL
- Métodos elementales
- *Dynamic Programming*
  - *Monte Carlo Methods*
  - *Temporal Difference learning*

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

146

## Monte Carlo Methods

- Introducción
- Evaluación de Política
- Mejoramiento de Política
- Control
- *On/off policy*
- MCM incremental

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

147

## MCM: Introducción

- Métodos de Monte Carlo – *Monte Carlo Methods*
- “Monte Carlo” nace en 1940 ... “Todo método de estimación con un gran **componente aleatorio**” ... por lo general requiere de muchas iteraciones
- Primera utilización en RL: 1968, Michie y Chambers
- No requiere modelo
- Aprende de interacción directa o simulada con el ambiente.
- Para simulación se necesitan **muestras** del modelo **NO** el modelo completo! ... **Mucho más sencillo**
- Se basa en promedios ... sólo funciona con tareas episódicas.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

148

## MCM: Evaluación de Política

- Recuerde – **Evaluación**: hallar los valores de los estados dependiendo de una determinada política.
- Se basa en la aproximación de los valores de los estados a partir de promedios de retornos completos obtenidos en diferentes episodios.
- *first-visit MCM*: considera sólo la primera visita al estado en cada episodio.
- *every-visit MCM*: considera todas las visitas a cada estado dentro de un mismo episodio.
- Ambos son similares ... el de la primera visita es más utilizado.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

149

## MCM: Evaluación de Política (2)

### Algorítmicamente

- Inicializar los valores de estado
$$V_0 = [V_0(s_1), V_0(s_2), \dots, V_0(s_i), \dots], \forall s \in S$$
- Simular/experimentar la política  $\pi$  durante muchos episodios.
- Mantener una lista con los retornos recibidos desde cada estado luego de la primera visita.
- Al final de cada episodio calcular el promedio de los retornos y ponerlo como valor de cada estado.

$$V_k(s_i) = \frac{\sum_{i=1}^k R_{ik}}{k}$$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

150

## MCM: Evaluación de Política (3)

- Observe que MCM no realiza *bootstrapping*!!
- ¿Para que sirven los valores  $V^\pi$  si no se tiene el modelo?
- Se sabe cual es el mejor estado ... pero como se llega a él??

**La mayoría de los métodos en adelante se basan en la estimación de Valores de Acción  $Q^\pi$**

- ¿Que problema hay si se utilizan políticas determinísticas?

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

151

## MCM: Evaluación de Política (4)

- Política determinística ... muchos pares estado acción sin visitar ... **Dilema Exploración – Explotación**

### **Solución:**

- Utilizar políticas estocásticas (**vistas al ppio**).
- Iniciar con un par estado-acción aleatorio cada episodio (*exploring starts*)

Funciona igual al que estima el valor de estado ... en vez de una lista se tiene una **matriz de valores  $Q$**

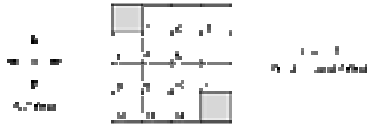
Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

152

## MCM: Evaluación de Política (3)

### Ejemplo ... El mismo *gridworld*



- Al evaluar la política óptima encontrada por DP, el algoritmo converge alrededor de 35 episodios a los valores óptimos (sin descuento):

0	-1	-2	-3
-1	-2	-3	-2
-2	-3	-2	-1
-3	-2	-1	0

VER COD. MATLAB

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

153

## MCM: Evaluación de Política (4)

- Solución para la política aleatoria (luego de 5000 episodios)

0	-14.163	-20.305	-22.710
-14.672	-18.437	-20.534	-20.668
-20.531	-20.546	-18.646	-14.209
-22.180	-20.312	-14.584	0

- Comparar con la solución de DP. [VER](#)
- ¿Se puede evaluar cualquier política? **VER COD. MATLAB**
- Ver ejemplo *Blackjack*, Sutton y Barto (5.1)

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

154

## MCM: Mejoramiento de Política

- En DP: mejoramiento *greedy* respecto a  $V^\pi$ .
- En MCM: mejoramiento *greedy* respecto a  $Q^\pi$ .
- Se encuentra la política que escoge, para cada estado  $s$ , la acción  $a$  con mayor valor  $Q$  según:

$$\pi(s) = \arg \max_a Q^\pi(s, a)$$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

155

## MCM: Control

- Control: problema de encontrar la política óptima
- De nuevo: GPI
- De manera similar a DP pero con valores de acción:

$$\pi_0 \xrightarrow{E} Q^{\pi_0} \xrightarrow{M} \pi_1 \xrightarrow{E} Q^{\pi_1} \xrightarrow{M} \pi_2 \xrightarrow{E} \dots \xrightarrow{M} \pi^* \xrightarrow{E} Q^*$$

- $E$ : paso en el que se realizan “**infinitos**” episodios para hallar los valores  $Q$  asociados a la política de turno.
- $M$ : se mejora la política para hacerla *greedy* respecto a los valores  $Q$ .
- Se utilizan “**inicios exploratorios**” (*exploring starts*)

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

156

## MCM: Control (2)

- **Por el teorema del mejoramiento converge a la política y los valores de acción óptimos**
- No se necesita realizar el paso  $E$  completamente como se vio con GPI para DP.
- En el caso extremo se puede hacer un paso de evaluación y un paso de mejoramiento.
- Si además se hace *in-place* se obtiene el algoritmo: *Monte Carlo ES* ...

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

157

## MCM: Control (3)

*MC – ES:*

Inicialización

- Iniciar arbitrariamente:  $Q(s, a) \forall s, a$  y  $\pi(s)$
- Iniciar lista retornos:  $R(s, a)$  vacía

Repetir por siempre

- Selecciona un par estado acción aleatorio (ES)
- Generar episodio con  $\pi$
- Almacenar retornos en la lista  $R(s, a)$ , recalcular  $Q$  (promedio)
- Mejorar la política

$$\pi(s) = \arg \max_a Q(s, a)$$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

158

## MCM: Control (4)

- MC – ES converge ... pero no ha sido demostrada esta convergencia: Sutton y Barto:  
*“In our opinion this is one of the most important open theoretical questions in reinforcement learning”*
- ¿Cómo evitar tener que utilizar inicios exploratorios?

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

159

## MCM: Control (5)

### Ejemplo ... El mismo *gridworld*

- Sin descuento y con mejoramiento *greedy*.
- 5 pasos max. Porque hay políticas que no llevan al terminal

- Valores  $Q$  obtenidos:

Estados

-1.08	-3.19	-3.33	-1.00
-2.63	-4.00	-3.00	-2.02
-3.37	-3.20	-3.66	-3.00
-1.03	-3.11	-3.00	-1.72
-2.00	-3.02	-2.81	-2.25
-3.03	-2.78	-3.28	-3.08
-3.08	-2.10	-2.72	-4.06
-2.02	-2.77	-4.06	-2.47
-2.06	-3.02	-3.03	-3.00
-3.34	-1.98	-2.00	-3.09
-3.25	-1.00	-1.26	-3.00
-3.00	-3.38	-3.00	-3.53
-4.04	-2.61	-2.00	-2.97
-3.18	-1.14	-1.00	-3.17
0	0	0	0

Acciones:  
Arriba, abajo,  
derecha e izquierda

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

160

## MCM: Control (6)

- Política aprendida:

	←	←	←
↑	↑	↓	↓
↑	↑	↓	↓
↑	→	→	

**Nota: No siempre es la óptima**

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

161

## MCM: Control (7)

- A partir de los valores de acción se pueden obtener los valores de estado (¿Como?):

	-1.00	-2.02	-3.00
-1.03	-2.00	-2.78	-2.10
-2.00	-2.96	-1.98	-1.00
-3.00	-2.00	-1.00	

**¿Por qué -2.78?**

- Valores similares a los obtenidos mediante DP y MCM anteriores.
- Ver ejemplo *blackjack* (5.3)

**VER COD. MATLAB**

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

162

## MCM: *On/off policy*

¿Como eliminar el uso de ES ?, Existen dos acercamientos:

- Métodos *on-policy*

Se evalúa y se mejora la política empleada para la toma de decisiones ... el agente mejora la política que sigue

- Métodos *off-policy*

Se evalúa y mejora una política diferente a la política empleada en la toma de decisiones ...

Política Comportamental (*behavior policy*): a partir de la cual se seleccionan acciones

Política de Estimación (*estimation policy*): la iterada.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

163

## MCM: *On-policy*

- Utiliza políticas *soft* ...

Ninguna acción posee probabilidad cero ningún estado:

$$\pi(s, a) > 0, \forall s \in S, a \in A(s)$$

- Ejemplos: Las vistas en Balance EE. [VER](#)

$$\pi(s, a) = \begin{cases} 1 - \epsilon & a = \arg \max_a (Q(s, a)) \\ \frac{\epsilon}{|A(s)| - 1} & a \neq a_{greedy} \end{cases} \quad \pi(s, a) = \frac{e^{\frac{Q(s, a)}{\tau}}}{\sum_{a' \in A(s)} e^{\frac{Q(s, a')}{\tau}}}$$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

164

## MCM: *On-policy* (2)

- La idea ... como siempre **GPI**
- OJO: para que el teorema del mejoramiento funcione la política que se sigue debe ser cercana a la *greedy*, por ejemplo la  $\epsilon$ -*greedy*.  
→ **Toda política *soft* respecto a  $Q^\pi$  es mejor o igual a  $\pi$ .**
- La idea entonces es mejorar la política haciéndola  $\epsilon$ -*greedy* respecto a los valores de acción
- Nota: ver prueba de convergencia a  $Q^*$  (pag. 123)

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

165

## MCM: *On-policy* (3)

$\epsilon$ -*greedy on-policy MCM*:

Inicialización

- Iniciar arbitrariamente:  $Q(s, a) \forall s, a$  y  $\pi(s, a)$
- Iniciar lista retornos:  $R(s, a)$  vacía

Repetir por siempre

- Generar episodio con  $\pi$
- Almacenar retornos en la lista  $R(s, a)$ , recalcular  $Q$  (prom.)
- Mejorar la política

$$\pi(s, a) = \begin{cases} 1 - \epsilon & a = \arg \max_a (Q(s, a)) \\ \frac{\epsilon}{|A(s)| - 1} & a \neq a_{\text{greedy}} \end{cases}$$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

166

## MCM: *On-policy* (4)

### Ejemplo ... El mismo *gridworld*

- Valores  $Q$  luego de 5000 episodios:

```

-1.59 -3.50 -4.50 -1.19
-2.50 -4.71 -3.41 -2.36
-4.97 -3.57 -4.01 -3.47
-1.18 -4.09 -4.50 -1.45
-3.54 -2.98 -3.33 -2.38
-3.45 -3.57 -3.67 -3.63
-2.72 -2.26 -3.39 -4.81
-2.43 -4.20 -4.75 -3.55
-3.60 -3.51 -3.61 -3.87
-2.92 -2.18 -2.46 -3.07
-3.77 -1.10 -1.95 -3.41
-4.16 -3.82 -3.41 -3.85
-4.28 -2.46 -2.35 -2.90
-4.54 -1.23 -1.16 -3.64
  0      0      0      0
    
```



Valores de Estado

	-1.19	-2.36	-3.47
-1.18	-2.38	-3.45	-2.26
-2.43	-3.51	-2.18	-1.10
-3.41	-2.35	-1.16	

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

167

## MCM: *On-policy* (5)

- Política *greedy* aprendida:

VER COD. MATLAB

	←	←	←
↑	←	↑	↓
↑	↓	↓	↓
→	→	→	

- ¿Comparar con la política aprendida por otros métodos?, ¿Son iguales? ¿importa?
- ¿Por qué son más grandes estos valores de estado que los encontrados por MCM-ES?

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

168

## MCM: *On-policy* (6)

- Algunas variaciones ... cambiando los valores esperados de recompensa:

15	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15



	←	←	←
↑	←	↑	↓
↑	←	↓	↓
→	→	→	

¿Cambiará algún valor de estado?

15	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15



	←	←	↓
↑	↑	↑	↓
↑	↑	→	↓
↑	↑	↑	

¿Cambiará algún valor de estado?

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

169

## MCM: *On-policy* (7)

- Valores  $Q$  para los dos ejemplos anteriores

-1.2836	-4.1429	-4.4286	-1.1371	-1.2836	-4.1429	-4.4286	-1.1371
-2.4093	-4.9231	-3.0761	-2.2099	-2.4093	-4.9231	-3.0761	-2.2099
-3.9259	-3.3535	-3.6552	-3.7000	-3.9259	-3.3535	-3.6552	-3.7000
-1.2227	-3.9286	-3.7500	-2.1494	-1.2227	-3.9286	-3.7500	-2.1494
-2.3450	-3.2083	-3.5357	-4.2045	-2.3450	-3.2083	-3.5357	-4.2045
-3.4133	-3.9167	-3.9125	-3.6591	-3.4133	-3.9167	-3.9125	-3.6591
-2.7903	-2.3072	-2.6190	-4.9231	-2.7903	-2.3072	-2.6190	-4.9231
-2.4418	-3.6379	-6.0000	-3.2364	-2.4418	-3.6379	-6.0000	-3.2364
-3.4693	-5.3231	-4.4634	-3.8750	-3.4693	-5.3231	-4.4634	-3.8750
-3.3110	-4.3438	-2.3405	-3.2424	-3.3110	-4.3438	-2.3405	-3.2424
-4.7273	-1.1579	-1.2815	-4.2500	-4.7273	-1.1579	-1.2815	-4.2500
-3.5394	-5.6563	-6.5455	-5.7447	-3.5394	-5.6563	-6.5455	-5.7447
-4.5319	-6.0000	-5.9676	-5.9286	-4.5319	-6.0000	-5.9676	-5.9286
-3.5473	-4.8049	-5.0173	-7.2857	-3.5473	-4.8049	-5.0173	-7.2857
0	0	0	0	0	0	0	0

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

170

## MCM: *Off-policy*

### “Evaluar una política siguiendo otra”

- Política comportamental:  $\pi$  (debe ser *soft*).
- Política de estimación:  $\pi'$ .
- Se puede hacer siempre y cuando: toda acción tomada bajo  $\pi$  sea también tomada, al menos ocasionalmente, bajo  $\pi'$ .
- Esto es:
$$\pi(s, a) > 0 \Rightarrow \pi'(s, a) > 0$$
- Ventaja: se puede estimar una política determinística (*greedy*).
- Desventaja: no se puede aprender de partes del episodio donde se tomaron acciones *non-greedy*.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

171

## MCM: Incremental

- Hace algún tiempo vimos como se pueden cambiar los promedios por actualizaciones incrementales. **VER**
- Es más cómodo ... no tiene tantos requerimientos de memoria.
- **¿Será estacionario el problema?**
- **... Observe que los retornos calculados con una política pueden no tener nada que ver con los calculados con otra!!**

$$Q_{k+1}(s, a) = Q_k(s, a) + \alpha (R_{k+1}(s, a) - Q_k(s, a))$$

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

172

## MCM: Incremental (2)

### Ejemplo ... El mismo *gridworld*

- Valores  $Q$  luego de 5000 episodios:

```

-1.9127  -3.5854  -5.2277  -1.0435
-2.4830  -6.5563  -4.3936  -2.3013
-4.6893  -7.0611  -4.8294  -3.5176
-1.1774  -2.8361  -2.3869  -1.1932
-2.5234  -3.0371  -3.2524  -2.1019
-3.3093  -5.6780  -4.1302  -7.7986
-4.6226  -2.1007  -3.1580  -8.9933
-2.3271  -4.1276  -7.7500  -4.9830
-3.5446  -4.7237  -5.3676  -4.5957
-3.0307  -5.6748  -2.4364  -4.9165
-7.3896  -1.0403  -1.3271  -8.3094
-8.6453  -8.0540  -4.1014  -7.1285
-5.8276  -5.5687  -2.4448  -6.3756
-3.7119  -4.9151  -1.0926  -6.6958
      0         0         0         0
    
```

Valores de Estado

	-1.04	-2.03	-3.51
-1.18	-2.10	-3.31	-2.10
-2.32	-3.54	-2.43	-1.04
-4.01	-2.44	-1.09	

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

173

## MCM: Incremental (3)

- Política *greedy* aprendida:

VER COD. MATLAB

	←	←	←
↑	←	↑	↓
↑	↑	→	↓
→	→	→	

- Como de costumbre este método es: mas rápido ... más inestable.
- Puede que no converja a los valores exactos ... lo que importan son las diferencias relativas.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

174

## Contenido

- Introducción
- Elementos del RL
- Acercamiento al RL
- El problema del RL

### Métodos elementales

- *Dynamic Programming*
- *Monte Carlo Methods*
- ***Temporal Difference learning***

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

175

## Bibliografía

- Aljibury, HALIM. Improving the performance of Q-learning with Locally Weighed Regression. Miami, USA 53p. Tesis (Master of Engineering). Graduate School of the University of Florida. 2001.
- Appl, MARTIN. Model-Based Reinforcement Learning in Continuous Environments. München, 149 p. Tesis (Doktors der Naturwissenschaften). Institut für Informatik der Technischen Universität München. Septiembre 2000.
- ASADPOUR, Masoud y SIEGWART, Roland. Compact Q-Learning for Microrobots with Processing Constraints. En: European Conference on Mobile Robots ECMR 2003. Radziejowice, Polonia. (Sep. 2003). 6p.
- BAIRD III, Leemon C. Advantage Updating. Technical report WL-TR-93-1146. Wright-Patterson Air Force Base. (1993). 48p.
- BAIRD III, Leemon C., KLOPF A. Harry. Reinforcement Learning with High-Dimensional, Continuous Actions. Technical Report WL-TR93-1147. Wright Laboratory. (1993). 19p.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

176

## Bibliografía (2)

- [10] BAIRD III, Leemon C. Residual Algorithms: Reinforcement Learning with Function Approximation. En: Prieditis, A., Russell, S., eds.: Machine Learning: Proceedings of the Twelfth International Conference (ICML95). San Mateo, USA, Morgan Kaufmann. (1995). p. 30–37.
- BERENJI, Hamid R., VENGEROV, David. A Convergent Actor-Critic-Based FRL Algorithm with Application to Power Management of Wireless Transmitters. En: IEEE Transactions on Fuzzy Systems. Piscataway, USA. Vol. 11, No. 4 (Ago. 2003). p. 478–485.
- BOYAN Justin A. y MOORE, Andrew W. Generalization in Reinforcement Learning: Safely Approximating the Value Function. En: G. Tesauro, D. S. Touretzky y T. K. Leen eds, Advanced in Neural Information Processing Systems. Cambridge, USA. Vol. 7 (1995). 8p.
- CASTRILLÓN, Jerónimo, GIRALDO, Daniel y PEÑA, Jorge A. Aprendizaje por Refuerzo en Espacios Continuos para la Evasión de Obstáculos en un Robot Móvil". Tesis (Ingeniería Electrónica). UPB, Medellín 2004, 365p.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

177

## Bibliografía (3)

- GASKETT, Chris, WETTERGREEN, David y ZELINSKY, Alexander. Reinforcement learning for a vision based mobile robot. En: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2000). Takamatsu, Japón. (2000). 12p.
- GLORENNEC, Pierre Yves. Reinforcement learning: an overview. En: en memorias del ESIT'00. Aachen, Alemania. (Sep. 2000); p. 17-35.
- GLORENNEC, Pierre Yves y JOUFFE, Lionel. A reinforcement learning method for an autonomous robot. En: Proceedings of EUFIT'96, Fourth European Congress on Intelligent Techniques and Soft Computing. Aachen, Alemania. Vol. 4, (Sep. 1996); p.1100-1104.
- GLORENNEC, Pierre Yves y JOUFFE, Lionel. Fuzzy Q-Learning. En: Proceedings of Fuzz-IEEE'97, Sixth International Conference on Fuzzy Systems. Barcelona, España. Vol 6, (Jul. 1997); p. 659-662.
- GORDON, Geoffrey J. Stable Function Approximation in Dynamic Programming. En: Machine Learning (proceedings of the twelve international conference). San Francisco, USA. (1995). 8p.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

178

## Bibliografía (4)

- Gordon, GEOFFREY J. Approximate Solutions to Markov Decision Processes. Pittsburgh, 140p. Tesis (Doctor of philosophy). School of Computer Science, Carnegie Mellon University. 1999.
- Kaelbling, Leslie Pack Kaelbling, LITTMAN Michael L. y MOORE, Andrew W. Reinforcement learning: A survey. En: Journal of Artificial Intelligence Research. Washington, USA. Vol. 4 (May. 1996); p. 236-285.
- KONDA, Vijay R., TSITSIKLIS, John N. Actor Critic Algorithms. En: Advances in Neural Information Processing Systems. MIT Press, Cambridge, USA. Vol. 12 (2000). p. 1008-1014.
- Lucidarme, PHILIPPE. Apprentissage et adaptation pour des ensembles de robots réactifs coopérants. Montpellier, 127 p. Tesis (docteur). Université de Montpellier II, école: Information, Structures et Systèmes. 2003.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

179

## Bibliografía (5)

- PENG, Jing y WILLIAMS, Ronald J. Incremental multi-step Q-learning. en Machine Learning. En: Proceedings of the Eleventh International Conference (ML94). Morgan Kaufmann, New Brunswick, USA. (Jul. 1994); p. 226-234.
- Rummery, GAVIN ADRIAN. Problem solving with reinforcement learning. Cambridge, 107 p. Tesis (Doctor of Philosophy). Cambridge University Engineering Department, 1995.
- SINGH, Satinder P., JAAKKOLA Tommi y JORDAN Michael I. Reinforcement Learning with Soft State Aggregation. En: G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, Advances in Neural Information Processing Systems. Boston, USA. Vol. 7 (1995). p. 361-368.
- Smart, WILLIAM DONALD, Making Reinforcement Learning Work on Real Robots. Rhode Island, 149p. Tesis (Doctor of Philosophy). Department of Computer Science, Brown University, May 2002.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

180



## Bibliografía (6)

- SUTTON, Richard S. et al. Policy Gradient Methods for Reinforcement Learning with Function Approximation. En: Advances in Neural Information Processing Systems. MIT Press, Cambridge, USA. (2000). p. 1057–1063.
- **SUTTON, Richard. y BARTO, Andrew., Reinforcement Learning: An Introduction. Cambridge, Massachusetts- USA. Bradford Book, MIT Press. 1998. 320p.**
- TAN, Ming. Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents. En: Proceedings of the Tenth International Conference on Machine Learning. Amherst, USA. (1993). p. 330-337.
- TESAURO, G. Practical Issues in Temporal Difference Learning. En: Machine Learning. Irvine, USA. Vol. 8 (1992); p. 257-277.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

181



## Bibliografía (7)

- TOUZET, Claude. Neural Networks and Q-Learning for Robotics. En: Proceedings International Joint Conference on Neural Networks IJCNN '99. Washington, USA. (Jul. 1999); p. 1-80.
- TOUZET, Claude. Neural Reinforcement Learning for Behavior Synthesis. En: Special issue on Learning Robot: the New Wave, N. Sharkey (guest ed.), Robotics and Autonomous Systems. Marsella, Francia. Vol. 22, No. 3-4 (Sep. 1997); p. 251-281.
- TSITSIKLIS, John N. y VAN ROY, Benjamin. An analysis of temporal-difference learning with function approximation. En: IEEE Transactions on Automatic Control. Boston, USA. Vol. 42 (1995); p. 674–690.

Feb-Jun 2005

RL. Diplomado en Computación Inteligente.  
UPB. jeronimocm@yahoo.com

182