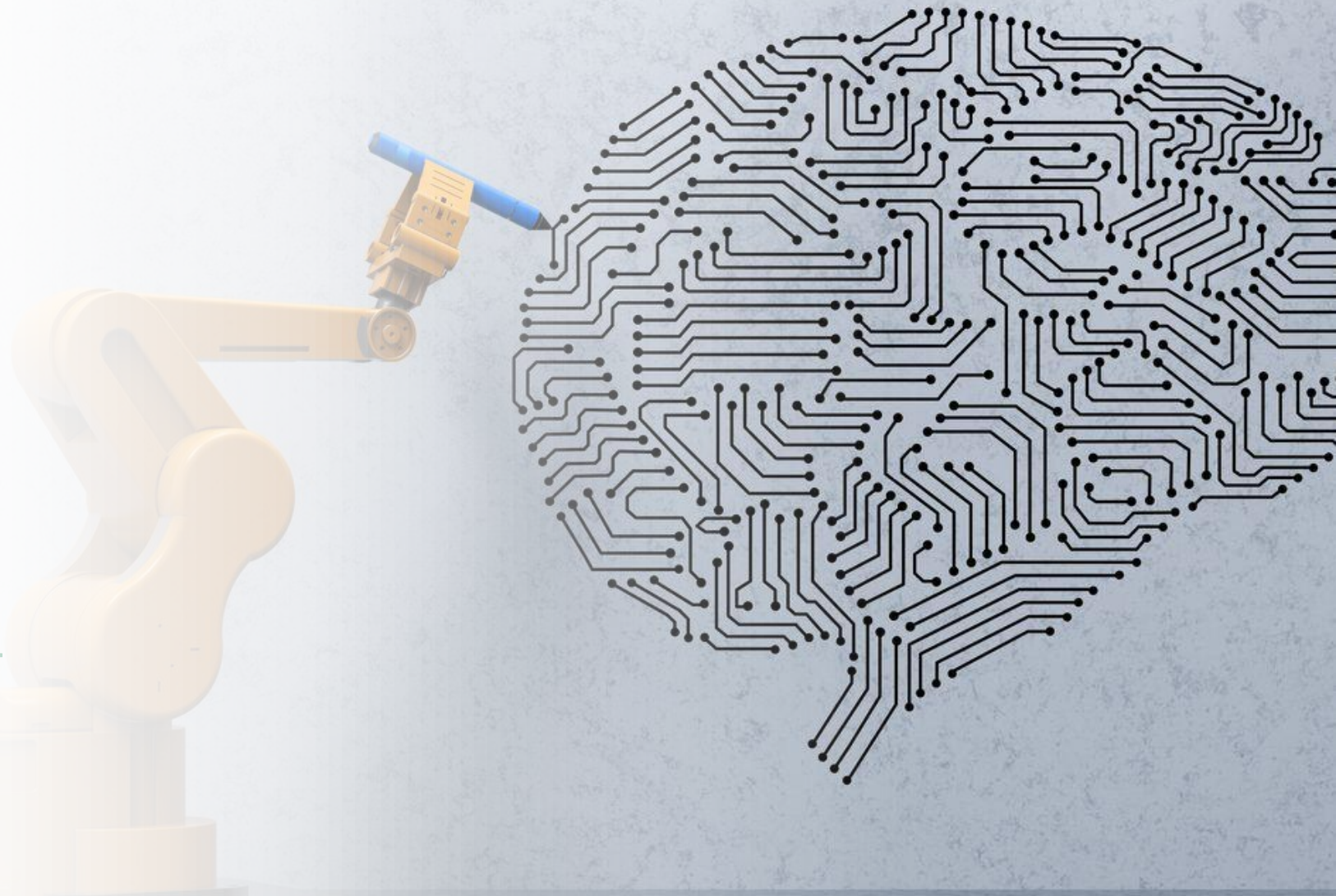
Aprendizaje por refuerzo

Clase 5: Predicción libre de modelo



Antes de empezar…

- Dudas de tarea 1

Para el día de hoy…

- Aprendizaje de Monte Carlo
- Aprendizaje de diferencia temporal (TD)
- TD(λ)

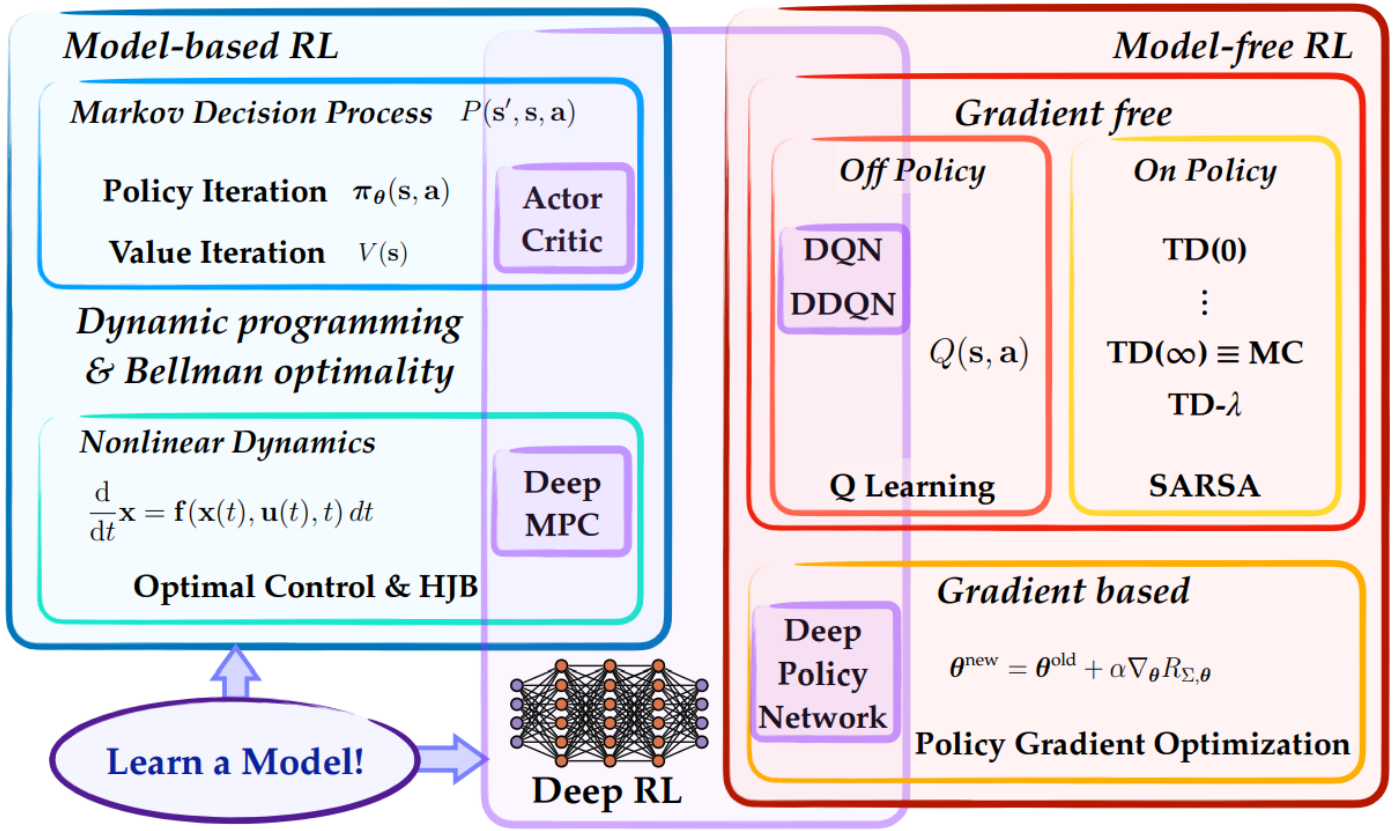
Lo que hemos visto…

- Evaluación de política
  - $v_{k+1}^{\pi}(s) \leftarrow \sum_{s}' p(s, \pi(s), s')[r(s, \pi(s), s') + \gamma v_{k}^{\pi}(s')]$
- Iteración de política
  - Evaluación: $v_{k+1}^{\pi_i}(s) \leftarrow \sum_{s}' p(s, \pi(s), s')[r(s, \pi(s), s') + \gamma v_{k}^{\pi_i}(s')]$
  - Mejora: $\pi_{i+1}(s) = argmax_a \sum_{s'} p(s, a, s') \, [r(s, a, s') + \gamma v^{\pi_i}(s')]$
- Iteración de valor
  - $v_{k+1} \leftarrow \max_a \sum_{s'} p(s, a, s') \, [r(s, a, s') + \gamma v_k(s')]$

Aprendizaje por refuerzo libre de modelo

- Hasta ahora hemos supuesto que alguien nos da el MDP de forma explicita
- Y que podemos explorarlo…
- Ahora trataremos de evaluar una política sin que alguien nos diga como funciona el mundo

Mapa


Model-based RL
Markov Decision Process $P(\mathrm{s}', \mathrm{s}, \mathrm{a})$
Policy Iteration $\pi_\theta(\mathrm{s}, \mathrm{a})$
Value Iteration $V(\mathrm{s})$
Actor Critic
Dynamic programming & Bellman optimality
Nonlinear Dynamics
$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)\, dt$
Deep MPC
Optimal Control & HJB
Learn a Model!
Deep RL
Model-free RL
Gradient free
Off Policy
DQN
DDQN
$Q(\mathrm{s}, \mathrm{a})$
Q Learning
On Policy
TD(0)
⋮
TD(∞) ≡ MC
TD-λ
SARSA
Gradient based
Deep Policy Network
$\theta^{\text{new}} = \theta^{\text{old}} + \alpha \nabla_\theta R_{\Sigma,\theta}$
Policy Gradient Optimization

Aprendizaje de Monte-Carlo (MC)

- Métodos que aprenden de episodios de experiencia
- Es libre de modelo. No hay conocimiento de las transiciones/recompensas
- Aprende de episodios completos
- Solo se puede aplicar a MDPs de episodios. Todos los episodios deben terminar

Evaluación de Monte Carlo de política

- Objetivo: aprender v_π de episodios de experiencia siguiendo la política π

$$S_1, A_1, R_2 \ldots, S_k \sim \pi$$

- El retorno es la suma total de recompensas

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-t-1} R_T$$

- La evaluación de la función es el valor esperado

$$v_\pi = \mathbb{E}[G_t | S_t = s, \pi]$$

- La evaluación de Monte Carlo usa la valor medio empírico en lugar del retorno esperado

Evaluación de Monte Carlo de Política en primera visita

- Para evaluar un estado s
- **La primera vez** que se visita s en el episodio
  - Se incrementa un contador $N(s) \leftarrow N(s) + 1$
  - Se actualiza el retorno $S(s) \leftarrow S(s) + G_t$
- Se estima la media del retorno $V(s) = \frac{S(s)}{N(s)}$
- Por la ley de números grandes $V(s) \rightarrow v_\pi(s)$ para $N(s) \rightarrow \infty$

El algoritmo

First-visit MC prediction, for estimating $V \approx v_\pi$

Input: a policy π to be evaluated

Initialize:

$V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$

$Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Loop forever (for each episode):

Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

Unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$:

Append G to $Returns(S_t)$

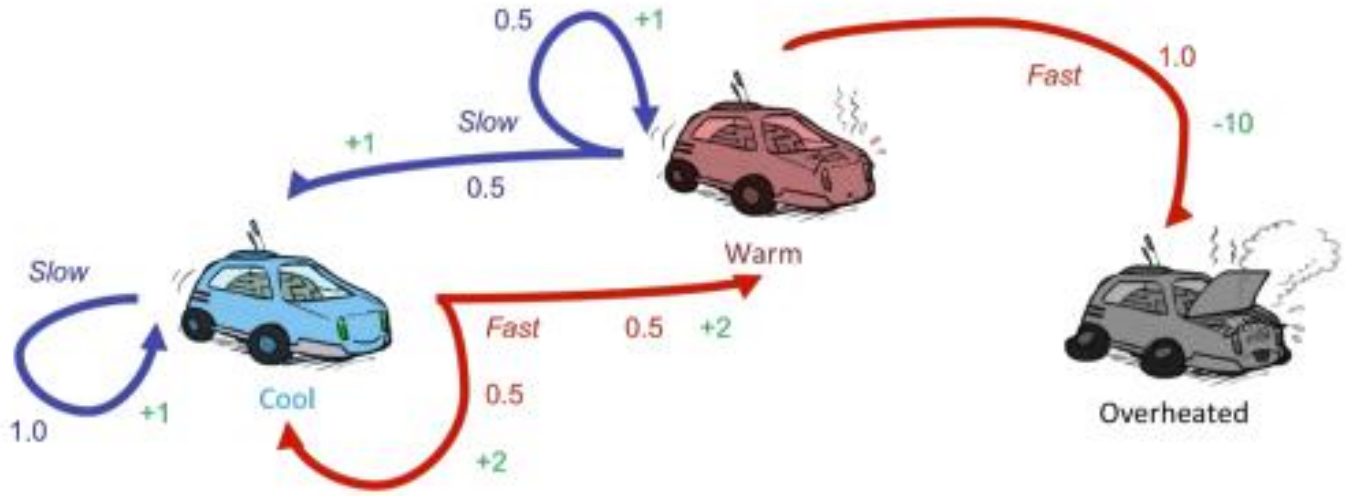
$V(S_t) \leftarrow \text{average}(Returns(S_t))$

Evaluación de Monte Carlo de Política en cada visita

- Para evaluar un estado s
- **Cada vez** que se visita s en el episodio
- Se incrementa un contador $N(s) \leftarrow N(s) + 1$
- Se actualiza el retorno $S(s) \leftarrow S(s) + G_t$
- Se estima la media del retorno $V(s) = \frac{S(s)}{N(s)}$
- Por la ley de números grandes $V(s) \rightarrow v_\pi(s)$ para $N(s) \rightarrow \infty$

Ejemplo 1

- $\pi(C) \rightarrow F; \pi(W) \rightarrow F$
- 3 turnos del juego
- Muestras
  - C,F,2,C,F,2,C,F,2
  - W,F,-10
  - C,F,2,W,F,-10
  - C,F,2,C,F,2,W,F,-10
- $N(s) =¿?$
- $S(s) =¿?$
- $V(s) =¿?$

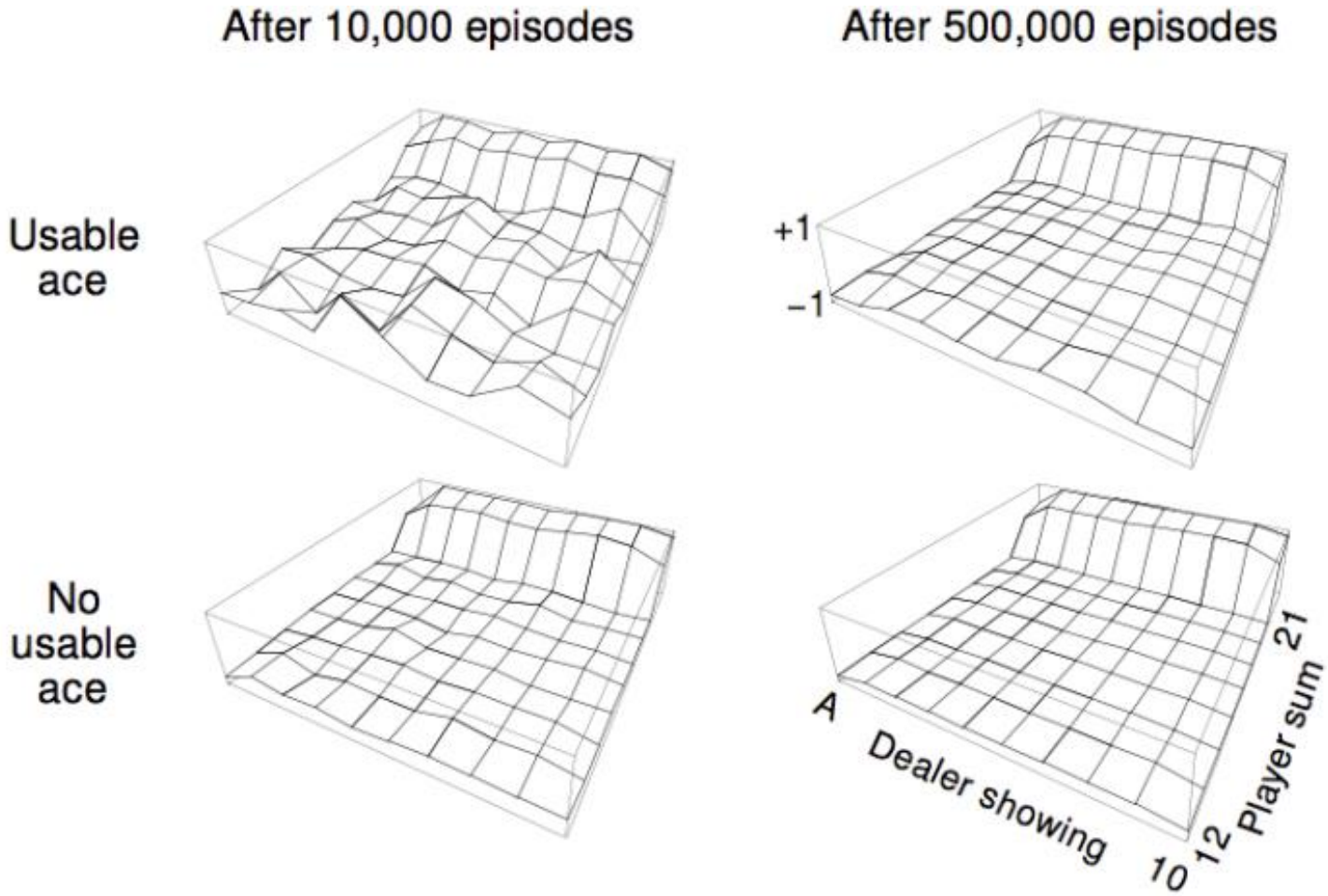


Ejemplo 2

- Estados (200):
  - La suma actual (12-21)
  - La carta del dealer
  - Tengo un as utilizable
- Acción stick: no quiero otra carta (y se termina el juego)
- Acción draw: pedir otra carta
- Suma de recompensa para stick:
  - +1 si la suma de cartas > suma del dealer
  - 0 si la suma de cartas = suma del dealer
  - -1 si la suma de cartas < suma del dealer
- Recompensa para draw:
  - -1 si la suma de cartas > 21 (y se termina el juego)
  - 0 de lo contrario
- Transiciones: automáticamente draw si suma de cartas < 12

Evaluación de la función de aprendizaje de Monte Carlo en Blackjack

Politica: stick si la suma de cartas es ≥ 20 de lo contrario twist



Media incremental

- La media $\mu_1, \mu_2, \ldots$ de una secuencia $x_1, x_2, \ldots$ puede ser calculada de forma incremental
- $\mu_k = \frac{1}{k}\sum_{j=1}^{k} x_j$
  - $= \frac{1}{k}(x_k + \sum_{j=1}^{k-1} x_j)$
  - $= \frac{1}{k}(x_k + (k-1)\mu_{k-1})$
  - $= \mu_{k-1} + \frac{1}{k}(x_k - \mu_{k-1})$

Incrementos de Monte Carlo

- Actualizar $V(s)$ incrementalmente después del episodio $S_1, A_1, R_2, \dots, S_T$
- Para cada estado S_t con retorno G_t
  - $N(S_t) \leftarrow N(S_t) + 1$
  - $V(S_t) \leftarrow V(S_t) + \frac{1}{N(S_t)}(G_t - V(S_t))$
- Para problemas no estacionarios, puede ser útil utilizar una media móvil

$$V(S_t) \leftarrow V(S_t) + \alpha\big(G_t - V(S_t)\big)$$

$$V(S_t) \leftarrow V(S_t) + \alpha(G_t - V(S_t) + \alpha(G_t - V(S_t)))$$

Aprendizaje de diferencia temporal (TD)

- Aprenden directamente de episodios de experiencia
- Es libre de modelo, no requieren conocimiento del MDP
- TD puede aprender de episodios incompletos por medio de bootstrapping
- Actualiza una conjetura hacía otra conjetura

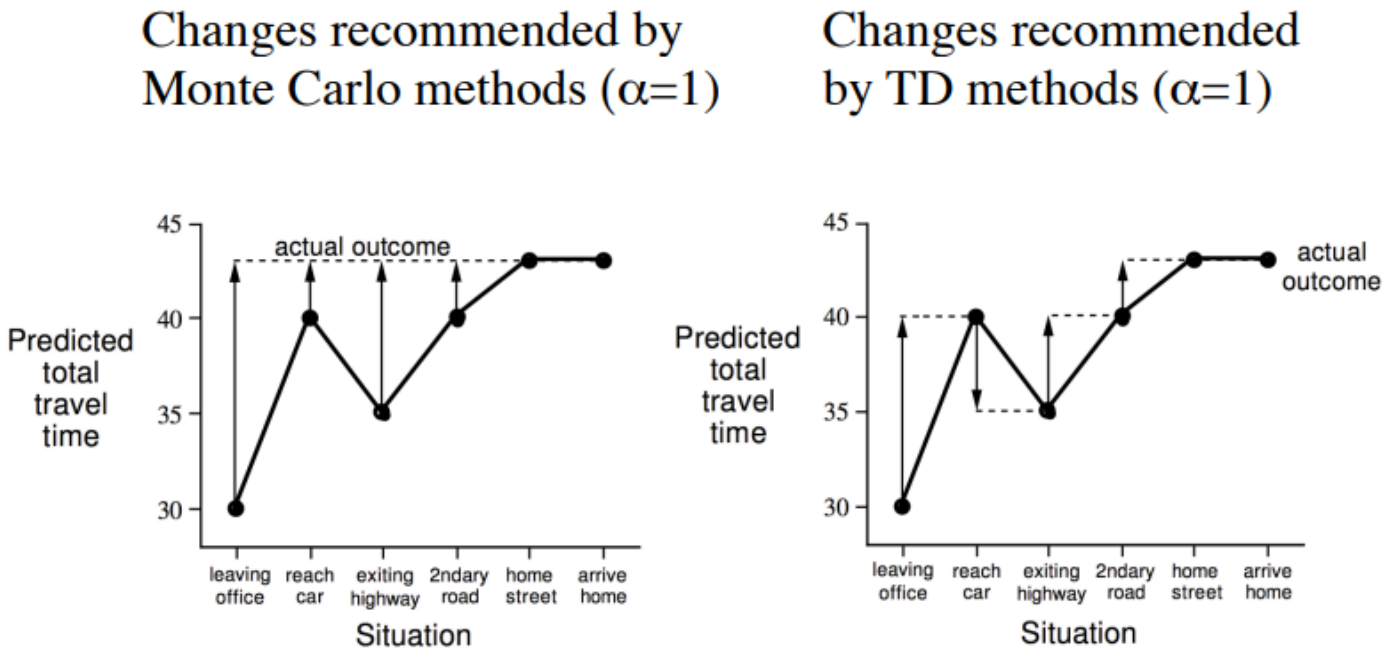
MC y TD

- Objetivo: aprender v_π en línea a partir de la experiencia bajo la política π
- Actualización en cada visita incremental de Monte-Carlo
  - $v(S_t) \leftarrow v(S_t) + \alpha(G_t - v(S_t))$
- El modelo más simple de diferencia temporal TD(0)
  - Actualizar el valor $V(S_t)$ hacia el retorno estimado $R_{t+1} + \gamma V(S_{t+1})$
  - $v(S_t) \leftarrow v(S_t) + \alpha\big(R_{t+1} + \gamma v(S_{t+1}) - v(S_t)\big)$
  - TD target: $R_{t+1} + \gamma v(S_t + 1)$
  - TD error: $\delta_t = R_{t+1} + \gamma v(S_{t+1}) - v(S_t)$

Un ejemplo

State	Elapsed Time (minutes)	Predicted Time to Go	Predicted Total Time
leaving office	0	30	30
reach car, raining	5	35	40
exit highway	20	15	35
behind truck	30	10	40
home street	40	3	43
arrive home	43	0	43

MC vs TD

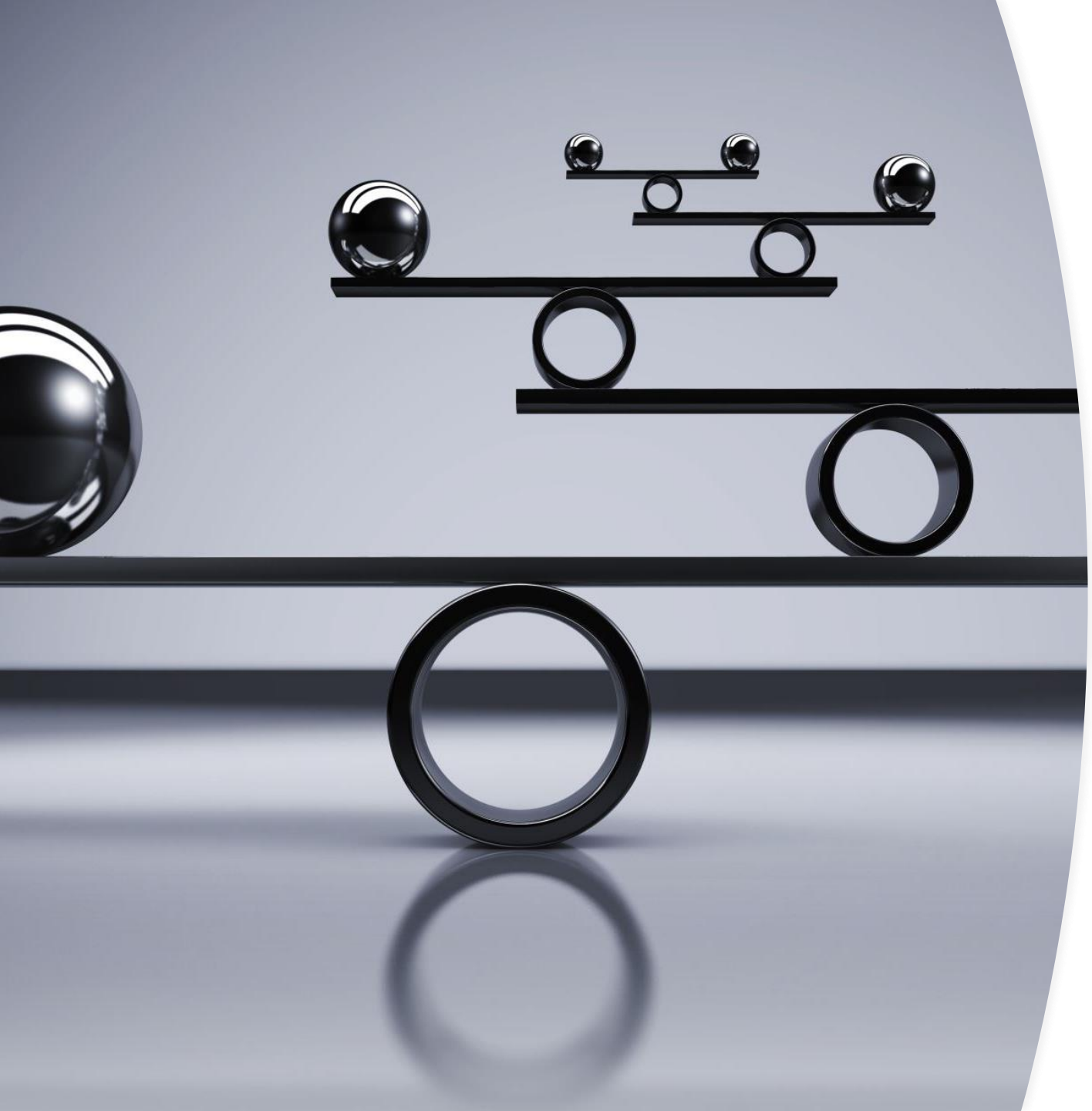

Changes recommended by Monte Carlo methods (α=1)
Changes recommended by TD methods (α=1)
Predicted total travel time
45
40
35
30
actual outcome
leaving office
reach car
exiting highway
2ndary road
home street
arrive home
Situation
Predicted total travel time
45
40
35
30
actual outcome
leaving office
reach car
exiting highway
2ndary road
home street
arrive home
Situation

Ventajas y desventajas de MC y TD

- TD puede aprender antes de conocer el resultado final
  - TD puede aprender en línea después de cada paso
  - MC debe esperar al final de episodio
- TD Puede sin saber el resultado final
  - TD puede aprender de secuencias incompletas
  - MC solo puede aprender de secuencias completas
  - TD trabaja con ambientes que no terminan
  - MC solo trabaja con ambientes que tienen final

Compromiso entre sesgo y varianza

- El retorno $G_t = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-1} R_T$ es un estimado sin sesgo de $v_\pi(S_t)$
- El verdadero TD target $R_{t+1} + \gamma v_\pi(S_{t+1})$ es un estimado sin sesgo de $v_\pi(S_t)$
- TD target $R_{t+1} + \gamma v(S_{t+1})$ es un estimado con sesgo de $v_\pi(S_t)$
- TD target tiene mucha menor varianza que el retorno



Ventajas y desventajas de MC vs TD

- MC tiene alta varianza y cero sesgo
  - Buenas propiedades de convergencia
  - No es sensible al valor inicial
  - Muy simple de entender y utilizar
- TD tiene baja varianza y algo de sesgo
  - Usualmente más eficiente que MC
  - TD(0) converge a $v_\pi(s)$
  - Más sensible al valor inicial

MC y TD en batch

- MC y TD convergen: $v_t \rightarrow v_\pi$ mientras la experiencia $\rightarrow \infty$
- ¿Qué sucede con experiencia finita?
- Consideremos experiencia finita
  - $S_1^1, A_1^1, R_2^1, \ldots, S_{T_1}^1$
  - $\vdots$
  - $S_1^K, A_1^K, R_2^K, \ldots, S_{T_1}^K$
- Muestrear repetidamente el episodio $k \in [1, K]$
- Aplicar MC o TD(0) al episodio k

Ejemplo AB

- Existen dos estados A, B no hay descuento y tenemos 8 episodios de experiencia
- ¿Cuál es el valor de $v(A)$ y $v(B)$?

A, 0, B, 0

B, 1

B, 1

B, 1

B, 1

B, 1

B, 1

B, 0

Diferencias en soluciones de batch

- MC converge minimiza el error cuadrático medio de los retornos observados. En el ejemplo AB, $v(A) = 0$

$$\sum_{k=1}^{K} \sum_{t=1}^{T_k} \left(G_t^k - v(S_t^k) \right)^2$$

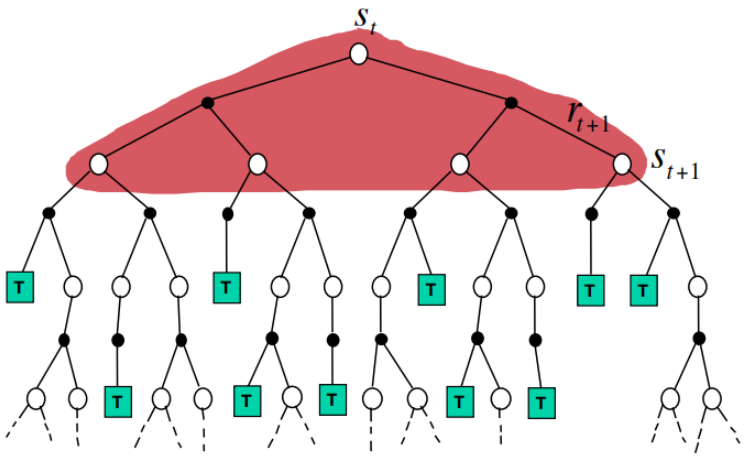
- TD converge a la solución de máxima verosimilitud del modelo de Markov dados los datos
  - Solución al MDP empírico $(\mathcal{S}, \mathcal{A}, \hat{p}, \gamma)$ que mejor explican los datos
  - En el ejemplo: $\hat{p}(S_{t+1} = B | S_t(A)) = 1$ y por lo tanto $v(A) = v(B) = 0.75$

Más comentarios

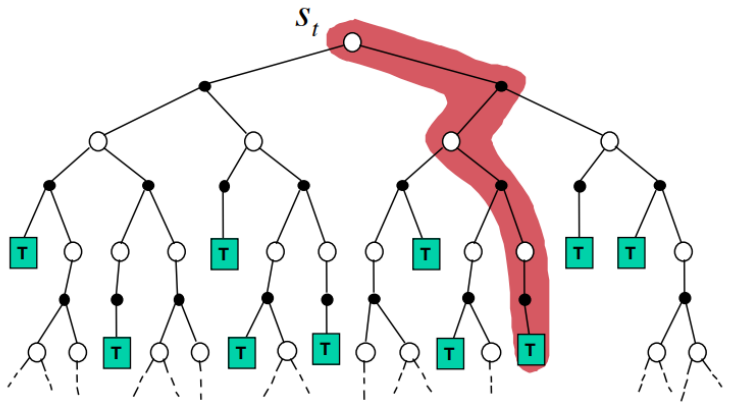
- TD explota la propiedad de Markov
  - Puede ayudar en ambientes totalmente observables
- MC no explota la propiedad de Markov
  - Puede ayudar en espacios parcialmente observables
- Con datos finitos, la solución puede diferir

La comparación

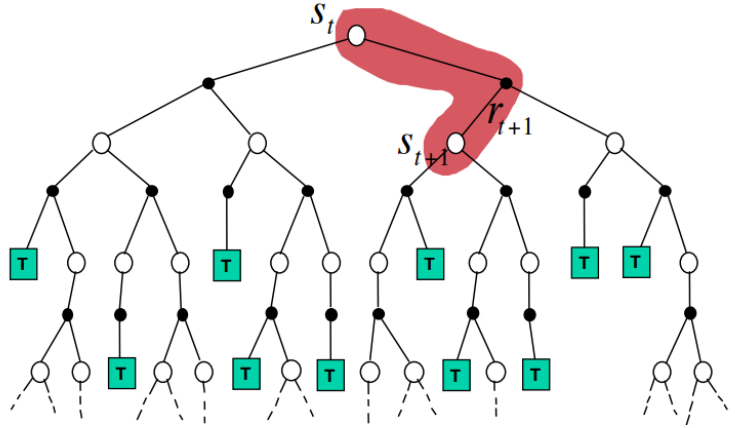
$$V(S_t) \leftarrow \mathbb{E}_\pi \left[R_{t+1} + \gamma V(S_{t+1}) \right]$$



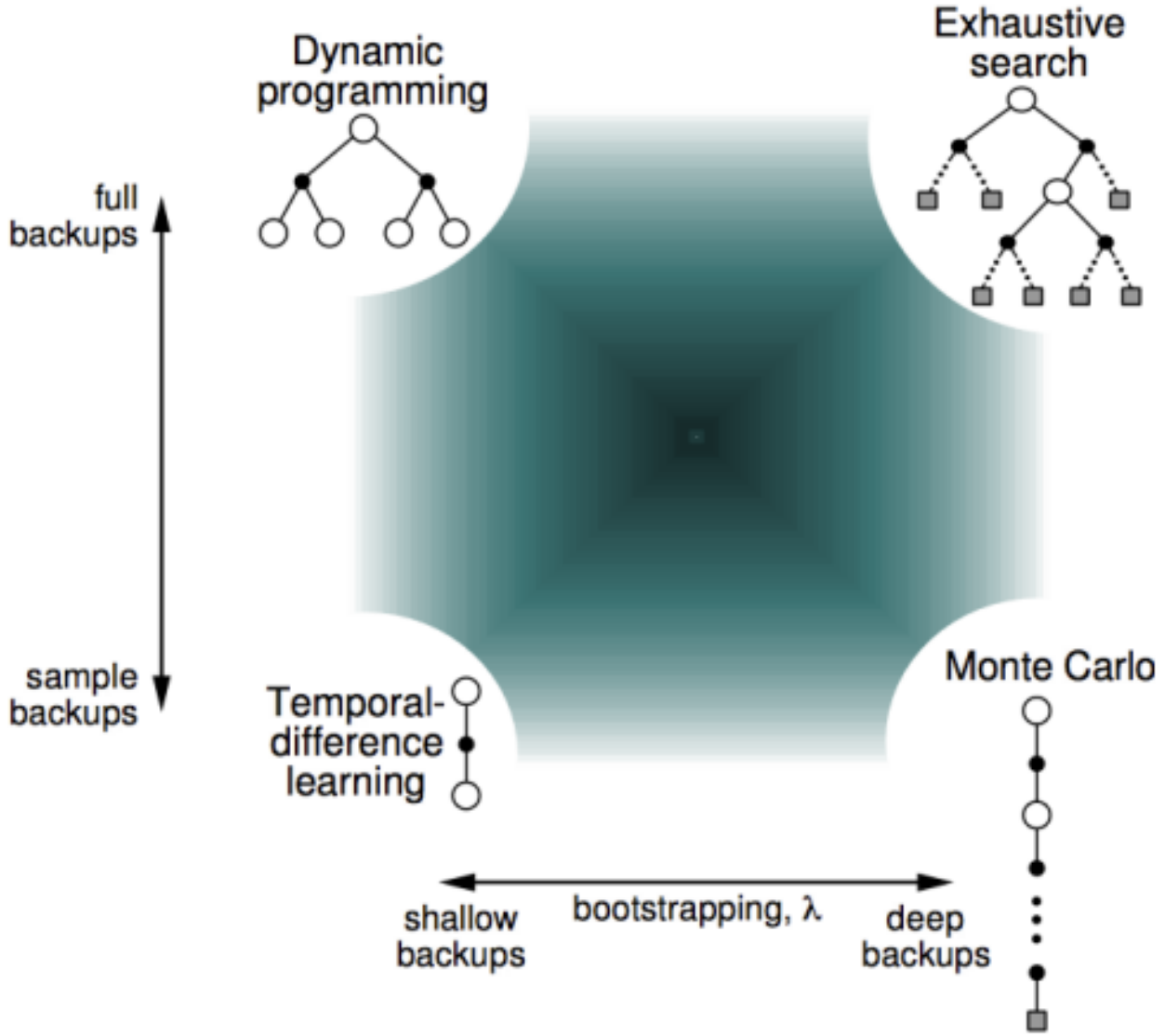
$$V(S_t) \leftarrow V(S_t) + \alpha \left(G_t - V(S_t) \right)$$

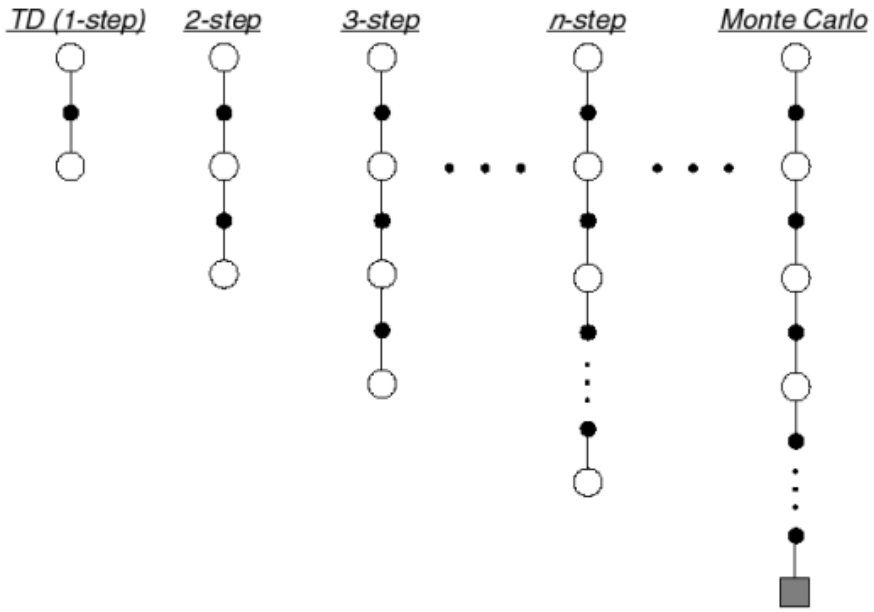


$$V(S_t) \leftarrow V(S_t) + \alpha \left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \right)$$



Visión unificada de aprendizaje por refuerzo



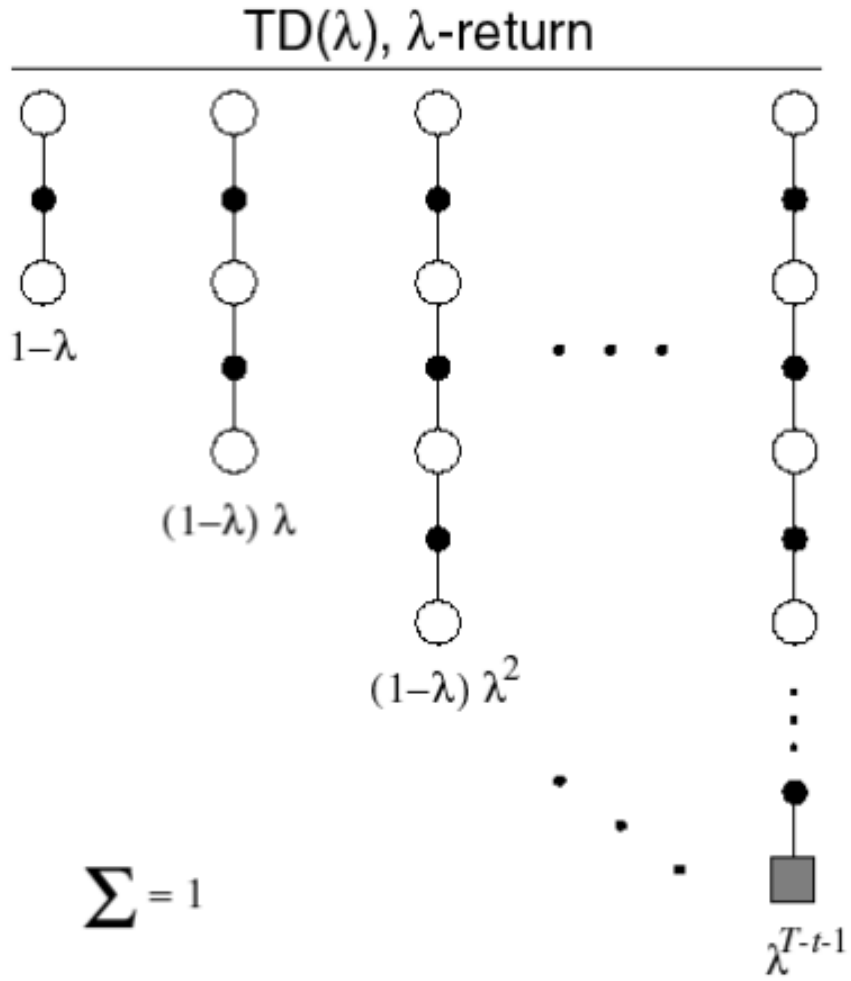


Predicción de n pasos

- Consideremos los retornos para n pasos para $n = 1,2, \dots, \infty$
  - $n = 1$ (TD) $G_t^{(1)} = R_{t+1} + \gamma v(S_{t+1})$
  - $n = 2 \qquad G_t^{(2)} = R_{t+1} + \gamma R_{t+2} + \gamma v(S_{t+1})$
  - $\vdots$
  - $n = \infty$ (MC) $G_t^{(\infty)} = R_{t+1} + \cdots + \gamma^{T-t-1} R_T$
- En general, el retorno de n pasos
- $G_t^n = R_{t+1} + \cdots + \gamma^n v(S_{t+n})$
- El aprendizaje de diferencia temporal de n pasos
- $v(S_t) \leftarrow v(S_t) + \alpha(G_t^{(n)} - v(S_t))$

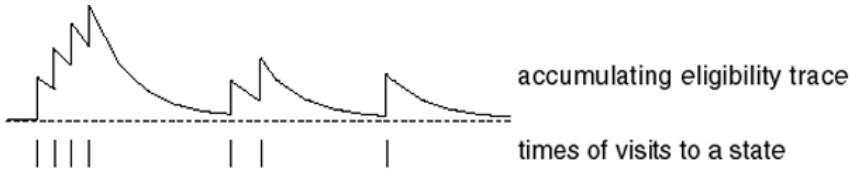
Retorno λ

- El retorno λ combina los retornos de los n pasos $G_t^{(n)}$
- Utiliza los pesos $(1-\lambda)\lambda^{n-1}$
- $G_t^{\lambda} = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}G_t^{(n)}$
- Visión hacia adelante $TD(\lambda)$
- $v(S_t) \leftarrow V(S_t) + \alpha(G_t^{\lambda} - v(S_t))$



Rastros de elegibilidad

- Problema de asignación de crédito: ¿qué causo el la descarga eléctrica?
- Frecuencia: asignar crédito a los estados más frecuentes
- Reciente: asignar crédito a lo más reciente
- Rastros de elegibilidad: combinar ambas
  - $e_0(s) = 0$
  - $e_t(s) = \gamma\lambda e_{t-1}(s) + 1(S_t = s)$
- Con ello, actualizar $v(s)$
  - $v(s) \leftarrow v(s) + \alpha\delta_t e_t(s)$



Para la otra vez…

- Control libre de modelo