Parte02

October 21, 2018

1 Introducción al Aprendizaje por Refuerzos

1.0.1 Curso Aprendizaje por Refuerzos, Diplomatura en Ciencia de Datos, Aprendizaje Automático y sus Aplicaciones

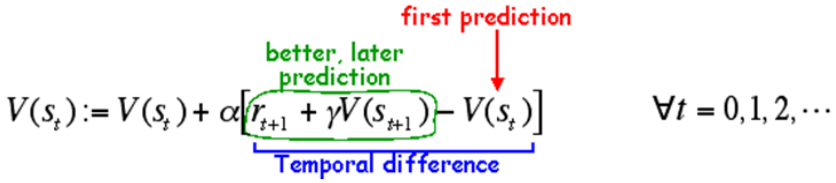
1.0.2 FaMAF, 2018

Agenda Clase 1 Parte 2

- Aprendizaje por Diferencias Temporales (TD Learning).
- SARSA: On Policy TD Learning.
- Q-Learning: Off-Policy TD Learning
- Eligibility Traces: TD(λ)
  - Uso de Trazas para la actualización de la Función de Valor.
    - Sarsa(λ) para predicción de Valor.
    - Sarsa(λ) para predicción de Acción-Valor.

1.1 Aprendizaje por Diferencias Temporales (TD Learning).

- La idea principal es actualizar una predicción de la función de valor en base al cambio que existe en la misma de un momento al siguiente (**Diferencia Temporal o Temporal Difference**)



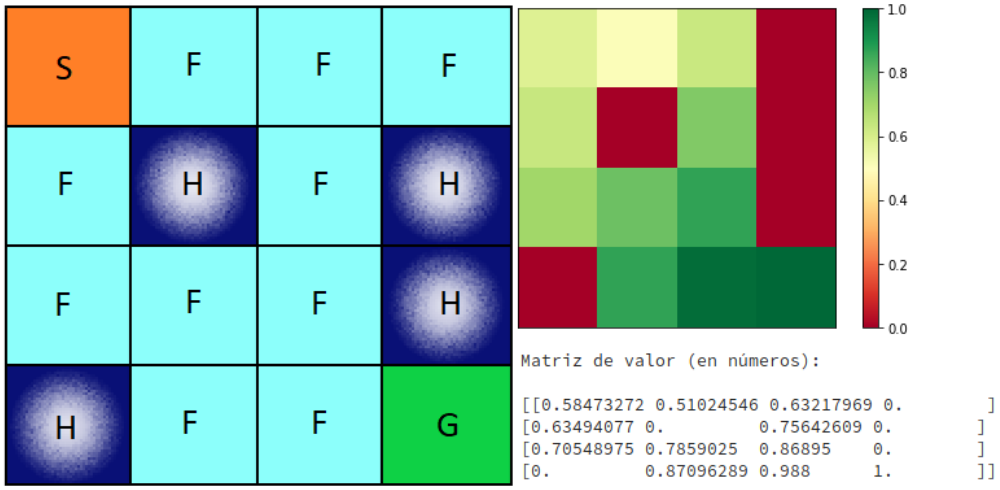
- Los algoritmos de aprendizaje basados en TD se emplean en mayor medida para realizar el CONTROL respecto de las acciones que ejecuta un agente que interactúa con su entorno. De esa manera, en lugar de aprender la función de estado-valor **V**, se orientan al aprendizaje de la función de acción-valor **Q**.

- En particular, existen dos enfoques principales para realizar el aprendizaje de funciones **Q**, ambos considerando el trade-off exploración/explotación: **on-policy** y **off-policy**. En particular, los métodos **on-policy** estiman $Q(s, a)$ para la política que el agente se encuentra ejecutando, para todos los estados s y acciones a.

- Dicha estimación puede ser realizada empleando el mismo método TD descripto anteriormente para actualizar V, an base a:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\Big]$$

1.2 Ejemplo Función de Valor: Frozen Lake



donde S= starting point (safe), F= frozen surface (safe), H=hole (fall to your doom), G= goal (where the frisbee is located)

(imagen de https://www.analyticsindiamag.com/openai-gym-frozen-lake-beginners-guide-reinforcement-learning/)

1.3 SARSA: On-Policy TD Learning

- El agente y su entorno interactúan a través de la ejecución de acciones, observación de estados y señales rewards.
  La inteligencia tendrá efecto sólo si el agente tiene claramente definidos objetivos o metas que persigue activamente mientras ocurre la interacción.

Sarsa (on-policy TD control) for estimating $Q \approx q_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Choose A from S using policy derived from Q (e.g., ε-greedy)
  Loop for each step of episode:
    Take action A, observe R, S'
    Choose A' from S' using policy derived from Q (e.g., ε-greedy)
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma Q(S', A') - Q(S, A)\big]$
    $S \leftarrow S'$; $A \leftarrow A'$;
  until S is terminal

1.4 Q-Learning: Off-Policy TD Learning

- Uno de los más importantes avances en RL fue el desarrollo del algoritmo **off-policy** conocido como Q-learning (Watkins, 1989).
  En su forma más simple, en Q-learning la actualización de la función de acción-valor realizada se define por:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\Big]$$

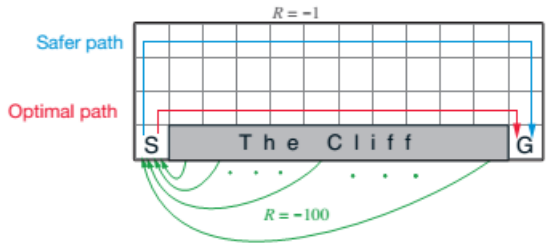
- Es este caso, la función de acción-valor Q aproxima directamente q, es decir, la función de acción-valor correspondiente a la política óptima, independientemente de la política seguida por el agente (de ahí su clasificación como **off-policy**), la cuál tiene efecto en el proceso de selección de acciones y por lo tanto determina cuales pares estado-acción se actualizan.

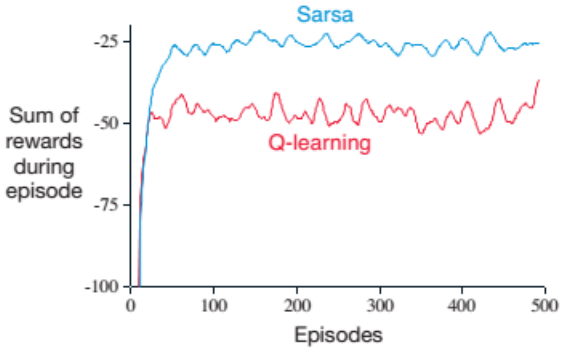
Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

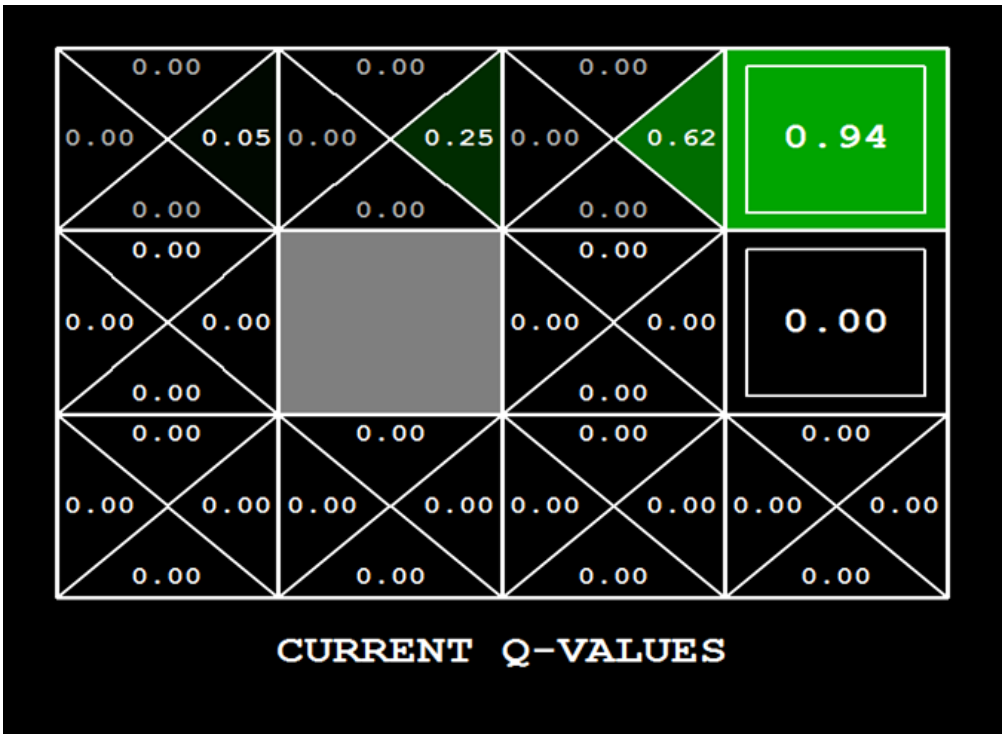
Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
    $S \leftarrow S'$
  until S is terminal

1.5 Ejemplo On-Policy vs. Off-Policy: The Cliff





1.6 Propagación de Valores en Q-Learning

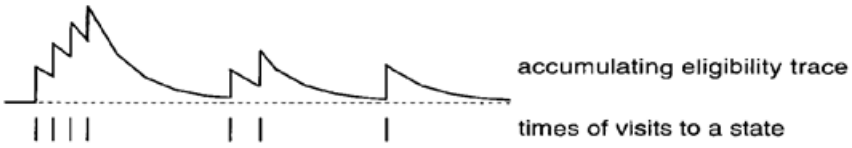


1.7 Elegibility Traces: TD(λ)

- Las Elegibility Traces (trazas) constituyen uno de los mecanismos básicos del Aprendizaje por Refuerzos.
- Prácticamente cualquier método de aprendizaje basado en diferencias temporales puede ser combinado con trazas para obtener un algoritmo más general que permite aprender de manera más eficiente.
- A nivel conceptual, una traza es un registro temporal de la ocurrencia de un evento, tal como la visita de un estado o la ejecución de una acción.
  La traza determina que dicho estado o acción es elegible en el momento de realizar una actualización de valor.
- Cuando ocurre una actualización basada en una diferencia temporal, sólo a los estados o acciones marcadas como elegibles se les asigna el crédito correspondiente.
- En TD(λ), existe una variable adicional asociada o bien a un estado, o bien a un par estado-acción (traza del estado/estado-acción).
- La traza de un estado s en t se denota como $e_t(s) \in R^+$.
  En cada paso, la traza para todos los estados decae su valor de acuerdo a $\gamma\lambda$, y la traza asociada al estado visitado se incrementa en 1.

$$e_t(s) = \begin{cases} \gamma\lambda\, e_{t-1}(s) & \text{if } s \neq s_t; \\ \gamma\lambda\, e_{t-1}(s) + 1 & \text{if } s = s_t, \end{cases}$$

- La actualización anterior ocurre para todos los estados $s \in S$, en donde γ es el factor de descuento y λ es el parámetro de decaimiento de traza.
  Este tipo de traza se denomina **traza de acumulación** porque acumula la cantidad de veces que el estado es visitado, disminuyendo gradualmente dicho valor a medida que el estado no recibe visitas, como se ilustra debajo:



- De esa manera, las trazas registran cuáles estados han sido visitados recientemente; **recientemente**, se define en términos de $\gamma\lambda$.

1.7.1 Uso de Trazas para la actualización de la Función de Valor

- La información provista por la traza se emplea en el proceso de actualización de la función de valor (V o Q).
  Si se considera que el valor de actualización por diferencia temporal es:

$$\delta_t = r_{t+1} + \gamma V_t(s_{t+1}) - V_t(s_t)$$

entonces dicha actualización se realiza en todos los estados de manera proporcional al valor de su traza.

$$\Delta V_t(s) = \alpha \delta_t e_t(s), \quad \text{for all } s \in \mathcal{S}.$$

1.7.2 Sarsa(λ) para predicción de Valor.

Initialize $V(s)$ arbitrarily and $e(s) = 0$, for all $s \in \mathcal{S}$
Repeat (for each episode):
  Initialize s
  Repeat (for each step of episode):
    $a \leftarrow$ action given by π for s
    Take action a, observe reward, r, and next state. s'
    $\delta \leftarrow r + \gamma V(s') - V(s)$
    $e(s) \leftarrow e(s) + 1$
    For all s:
      $V(s) \leftarrow V(s) + \alpha \delta e(s)$
      $e(s) \leftarrow \gamma \lambda e(s)$
    $s \leftarrow s'$
  until s is terminal

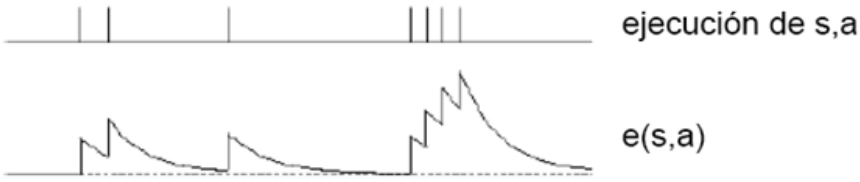
1.7.3 Sarsa(λ) para predicción de Acción-Valor.

$$e_t(s,a) = \begin{cases} \gamma\lambda e_{t-1}(s,a) + 1 & \text{if } s = s_t \text{ and } a = a_t \\ \gamma\lambda e_{t-1}(s,a) & \text{otherwise} \end{cases}$$

$$Q_{t+1}(s,a) = Q_t(s,a) + \alpha\delta_t e_t(s,a)$$

$$\delta_t = r_{t+1} + \gamma Q_t(s_{t+1},a_{t+1}) - Q_t(s_t,a_t)$$

Eligibility trace



Initialize $Q(s,a)$ arbitrarily and $e(s,a) = 0$, for all s, a
Repeat (for each episode):
  Initialize s, a
  Repeat (for each step of episode):
    Take action a, observe r, s'
    Choose a' from s' using policy derived from Q (e.g., ε-greedy)
    $\delta \leftarrow r + \gamma Q(s',a') - Q(s,a)$
    $e(s,a) \leftarrow e(s,a) + 1$
    For all s, a:
      $Q(s,a) \leftarrow Q(s,a) + \alpha\delta e(s,a)$
      $e(s,a) \leftarrow \gamma\lambda e(s,a)$
    $s \leftarrow s'$; $a \leftarrow a'$
  until s is terminal

In [ ]: