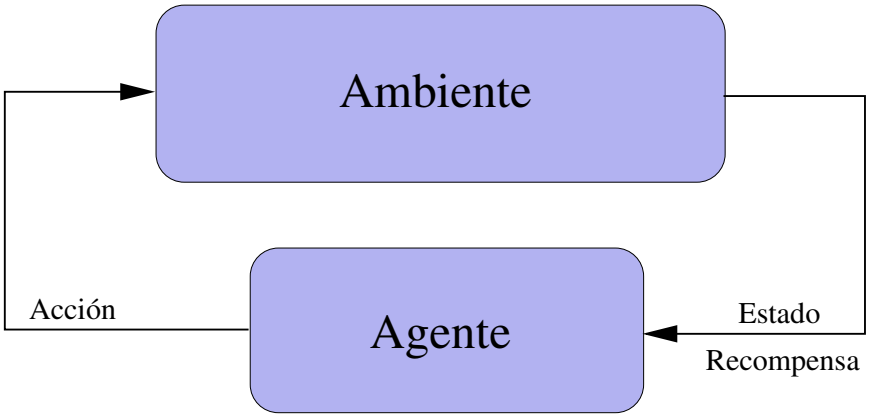
Aprendizaje reforzado en espacio de estado continuo

JULIO WAISSMAN VILANOVA

Ideas básicas

1. Problema



- $t = 1, 2, \ldots$ tiempo discreto, se asume una ventana de tiempo suficientemente grande,
- Espacio de estados X finito (x_t, estado en el instante t),
- Conjunto de acciones A finitas, $A(x) \subseteq A$, acciones admisibles para $x \in X$,
- Ambiente markoviano discreto (MDP),

$$P_{i,j}(a) = \Pr(x_{t+1} = x_j | x_t = x_i, a_t = a),$$

- $r : X \times A \to \mathbb{R}$, refuerzo (positivo: estimulo, negativo: castigo).

El objetivo del agente es maximizar su recompensa, definida como

$$R_t = \sum_{k=t}^{\infty} \gamma^{k-t} r(x_{k+1}, a_k),$$

a través de una política determinista y estacionaria $\pi : X \to A$,

$$a_t = \pi(x_t), \quad a_t \in A(x_t).$$

2. Función de valor de estado y estado–acción

- Función de valor de estado:

$$\begin{aligned} V^{\pi}(x_i) &= E\Big[\sum_{t=0}^{\infty}\gamma^t r(x_{t+1},\pi(x_t))|x_0 = x_i\Big], \\ &= \sum_{x_j\in X} P_{i,j}(\pi(x_i))\Big(r(x_j,\pi(x_i)) + \gamma E\Big[\sum_{t=1}^{\infty}\gamma^{t-1} r(x_{t+1},\pi(x_t))|x_1 = x_j\Big]\Big), \\ &= \sum_{x_j\in X} P_{i,j}(\pi(x_i))\Big(r(x_j,\pi(x_i)) + \gamma V^{\pi}(x_j)\Big) \end{aligned}$$

- Función de valor estado–acción:

$$\begin{aligned} Q^{\pi}(x_i,a_k) &= E\Big[\sum_{t=0}^{\infty}\gamma^t r(x_{t+1},a_t)|x_0 = x_i, a_0 = a_k\Big], \\ &= \sum_{x_j\in X} P_{i,j}(a_k)\Big(r(x_j,a_k) + \gamma E\Big[\sum_{t=1}^{\infty}\gamma^{t-1} r(x_{t+1},\pi(x_t))|x_1 = x_j\Big]\Big), \\ &= \sum_{x_j\in X} P_{i,j}(a_k)\Big(r(x_j,a_k) + \gamma V^{\pi}(x_j)\Big), \\ &= \sum_{x_j\in X} P_{i,j}(a_k)\Big(r(x_j,a_k) + \gamma Q^{\pi}(x_j,\pi(x_j))\Big). \end{aligned}$$

- Valores óptimos

$$\begin{aligned} V^*(x_i) &= \max_{\pi} V^{\pi}(x_i), \\ Q^*(x_i,a_k) &= \max_{\pi} Q^{\pi}(x_i,a_k) \end{aligned}$$

- Política óptima

$$\pi^*(x_i) = \arg\max_{a} Q^*(x_i,a)$$

¿Es que el agente puede estimar la política óptima (o algo parecido) interactuando con el Ambiente?

De ser así, entonces se puede *aprender* una política cercana a la óptima, simulando un agente y el ambiente.

3. Diferencias temporales

Durante la simulación:

1. El ambiente en el tiempo t se encuentra en el estado x, y se tiene un valor nominal para $V^\pi(x)$.
2. El agente decide aplicar la acción $\pi(x)$ al ambiente.
3. El ambiente en el tiempo $t+1$ se encuentra en el estado x', y se tiene un valor nominal para $V^\pi(x')$.

Entonces, se puede estimar en el instante $t+1$ la función $\hat{V}^\pi(x)$ como

$$\hat{V}^\pi(x) = r(x', \pi(x)) + \gamma V^\pi(x'),$$

debido a que en el instante $t+1$ no existe incertidumbre.

Idealmente,

$$\Delta V^\pi(x) = \hat{V}^\pi(x) - V^\pi(x) = 0.$$

Si se inicializan $V^\pi(x)$, $\forall x \in X$ al azar, habrá una diferencia entre $\hat{V}^\pi(x)$ y $V^\pi(x)$. Un método sencillo para encontrar $V^\pi(x)$ es minimizar $|\Delta V^\pi(x)|$.

Si se simula el sistema, entonces, en un instante $t+1$ se puede modificar el valor de la función de velor del estado x en que el ambiente se encontraba en el instante t como

$$V^\pi(x) \leftarrow V^\pi(x) + \alpha \Delta V^\pi(x)$$

donde $\alpha \in (0,1)$ se conoce como factor de aprendizaje. Este método de optimización por búsqueda directa se conoce como *descenso de gradiente estocástico*. El descenso de gradiente estocástico tiende a la vecindad de un punto crítico, el cual no es un máximo.

A este método se le conoce como **diferencias temporales**, o TD(0).

Algoritmo 1 TD(0)

Inicializa E, episodios
Inicializa T, pasos por episodio
para todo $x \in X$ **hacer**
  Inicializa $V(x)$
fin para
$e \leftarrow 1$
repetir
  $t \leftarrow 0$
  Inicializa $x \in X_0$, donde $X_0 \subseteq X$ conjunto de estados iniciales
  repetir
    $a \leftarrow \pi(x)$
    Aplicar a al entorno
    Observar estado siguiente x'
    $V(x) \leftarrow V(x) + \alpha\big(r(x', a) + \gamma V(x') - V(x)\big)$
    $t \leftarrow t + 1$
  hasta $t > T$
  $e \leftarrow e + 1$
hasta $e > E$

4. Diferencias temporales para valor estado–acción

¿Y si se quiere aprender una política cercana a la óptima?
Se puede estimar las funciones de valor estado–acción con TD(0).

1. Al instante t el sistema se encuentra en el estado x.
2. Se selecciona $a_M = \arg\max_{a \in A(x)} Q(x,a)$.
3. El agente aplica la acción a_M.
4. Al instante $t+1$ el sistema se encuentra en el estado x'.
5. Se selecciona $a'_M = \arg\max_{a \in A(x')} Q(x',a)$.
6. $\hat{Q}(x,a_M) = r(x',a_M) + \gamma Q(x',a'_M)$.
7. $\Delta Q(x,a_M) = \hat{Q}(x,a_M) - Q(x,a_M)$.
8. Se actualiza $Q(x,a_M)$ por descenso de gradiente

$$Q(x,a_M) \leftarrow Q(x,a_M) + \alpha \Delta Q(x,a_M).$$

¿Es que $Q(s,a_M)$ tiende a $Q^*(s,a_M)$?

El descenso de gradiente asegura solo un punto crítico de la politica *ávida* $\pi(x) = \max_a Q(s,a)$. Q(s,a) puede tender a un mínimo local.

5. El dilema de Exploración Explotación (EEP)

Si durante el aprendizaje se utiliza una política *ávida*, entonces se tiende rápidamente a un mínimo local que depende en gran medida de los valores iniciales de $Q(x,a)$. Si por otra parte, se utiliza una política que explora el espacio de estados, se tiene a un problema tan lento (o más) como la programación dinámica, pero sin sus ventajas.

- Política ϵ – ávida: $\pi_\epsilon(x) = \arg\max_a Q(x,a)$ con probabilidad $1-\epsilon$, de lo contrario $\pi_\epsilon(x) = a$, donde $a \in A(x)$ se escoge de forma aleatoria.
- Política *Softmax*: $\pi_\tau(x) = a$, donde $a \in A(x)$ es una variable aleatoria con

$$\Pr(a) = \frac{\exp(Q(s,a)/\tau)}{\sum_{b \in A(x)} \exp(Q(s,b)/\tau)}$$

6. Métodos *on–policy* y *off–policy*

La función de actualización de las diferencias temporales puede calcularse como

$$Q(x_t, a_t) \leftarrow Q(x_t, a_t) + \alpha\Big(r(x_{t+1}, a_t) + \gamma Q(x_{t+1}, a_{t+1}) - Q(x_t, a_t)\Big),$$

lo que se conoce como métodos *on policy*. Éste, en particular, se conoce como SARSA(0).

Pero... ¡El valor estado–acción $Q(x, a)$ que queremos estimar no es la de la política de exploración–explotación, si no el de la política óptima! Entonces podemos hacer dos cosas:

1. Modificar la política de exploración explotación, para reducir en forma suave la exploración, hasta terminar con una política *ávida*. Fácil para el análisis de convergencia del algoritmo, pero muy lento en la práctica.
2. Utilizar una política de exploración–explotación para la simulación, pero aprender el valor de estado–acción de una política *ávida* que no se aplica.

La función de actualización de las diferencias temporales puede calcularse como

$$Q(x_t, a_t) \leftarrow Q(x_t, a_t) + \alpha\Big(r(x_{t+1}, a_t) + \gamma \max_{b \in A(x_{t+1})} Q(x_{t+1}, b) - Q(x_t, a_t)\Big),$$

lo que se conoce como métodos *off policy*. Éste, en particular, se conoce como Q–Learning(0).

Algoritmo 2 SARSA(0)

Inicializa $Q(s,a)$ arbitrariamente $\forall x \in X, a \in A(x)$
para cada episodio **hacer**
  $t \leftarrow 0$
  Inicializa $x \in X_0$, donde $X_0 \subseteq X$ conjunto de estados iniciales
  $a \leftarrow \pi_{EEP}(x)$
  repetir
    Aplicar a al entorno, observar estado siguiente x'
    $a' \leftarrow \pi_{EEP}(x')$
    $Q(x,a) \leftarrow Q(x,a) + \alpha\big(r(x',a) + \gamma Q(x',a') - Q(x,a)\big)$
    $a \leftarrow a'$, $x \leftarrow x'$, $t \leftarrow t+1$
  hasta $t > T$
fin para

Algoritmo 3 $Q(0)$

Inicializa $Q(s,a)$ arbitrariamente $\forall x \in X, a \in A(x)$
para cada episodio **hacer**
  $t \leftarrow 0$
  Inicializa $x \in X_0$, donde $X_0 \subseteq X$ conjunto de estados iniciales
  repetir
    $a \leftarrow \pi_{EEP}(x)$
    Aplicar a al entorno, observar estado siguiente x'
    $Q(x,a) \leftarrow Q(x,a) + \alpha\big(r(x',a) + \gamma \max_{b \in A(x')} Q(x',b) - Q(x,a)\big)$
    $x \leftarrow x'$, $t \leftarrow t+1$
  hasta $t > T$
fin para

7. AR para ambientes con espacios de estados y acciones infinitos

- X, A y $A(x)$ son espacios de dimensión infinita, o muy grande para almacenarlo en forma tabular.
- La función de valor de estado siguiendo una política π será entonces:

$$\begin{aligned} V^{\pi}(x_i) &= E\Big[\sum_{t=0}^{\infty} \gamma^t r(x_{t+1}, \pi(x_t)) | x_0 = x_i\Big], \\ &= \int_{x_j \in X} p(x_1 = x_j | x_0 = x_i, a_0 = \pi(x_i)) \Big(r(x_j, \pi(x_i)) + \gamma V^{\pi}(x_j)\Big) \end{aligned}$$

- La función de valor de estado–acción:

$$\begin{aligned} Q^{\pi}(x_i, a_k) &= E\Big[\sum_{t=0}^{\infty} \gamma^t r(x_{t+1}, a_t) | x_0 = x_i, a_0 = a_k\Big], \\ &= \int_{x_j \in X} p(x_1 = x_j | x_0 = x_i, a_0 = a_k) \Big(r(x_j, a_k) + \gamma Q^{\pi}(x_j, \pi(x_j))\Big). \end{aligned}$$

- Si podemos simular el sistema, entonces podemos ajustar una política cercana a la óptima (al menos en una región del espacio de estados que sea interesante) ¡Sin necesidad de resolver el problema de integración numérica!

¡Pero $Q(x, a)$ no se puede almacenar en forma tabular!

La solución es aproximar $Q(x, a)$ a través de una función de aproximación, tal que

$$Q(x, a) \approx f(x, a, \theta),$$

donde $\theta = [\theta_1, \ldots, \theta_n]^T$,es un vector de parámetros.

El problema de aprendizaje consiste entonces a estimar los valores de θ a partir de la idea de *diferencias temporales*. Por ejemplo, utilizando SARSA(0)

$$
\begin{aligned}
Q(x_t, a_t) &\leftarrow Q(x_t, a_t) + \alpha\Big(r(x_{t+1}, a_t) + \gamma Q(x_{t+1}, a_{t+1}) - Q(x_t, a_t)\Big), \\
f(x_t, a_t, \theta) &\leftarrow f(x_t, a_t, \theta) + \alpha\Big(r(x_{t+1}, a_t) + \gamma f(x_{t+1}, a_{t+1}, \theta) - f(x_t, a_t, \theta)\Big), \\
\theta &\leftarrow \theta + \alpha\Big(r(x_{t+1}, a_t) + \big(\gamma f(x_{t+1}, a_{t+1}, \theta) - f(x_t, a_t, \theta)\big)\nabla_\theta f(x_t, a_t, \theta)\Big),
\end{aligned}
$$

donde $\nabla_\theta f(x_t, a_t\theta)$ es el gradiente de $f(x, a, \theta)$ respecto a θ y evaluado para los valores de $x = x_t$, $a = a_t$ y θ.

- Es importante que sea sencillo encontrar $\sup_u f(x, u, \theta)$ para x y θ dadas.
- El aprendizaje es muy sensible a la política de exploración–explotación.

Ejemplo:

- $X = \mathbb{R}^n$, $A = \mathbb{R}^m$, $A(x) = A$, para todo $x \in X$.
- $x_t = [x_{t,1}, \ldots, x_{t,n}]^T$, $a_t = [a_{t,1}, \ldots, a_{t,m}]^T$.
- Con el fin de encontrar fácilmente el valor de la acción ávida se selecciona una función de aproximación cuadrática

  $$f(x_t, a_t, \Theta) = z_t^T \Theta z_t,$$

  donde $z = [x_t^T, a_t^T]^T$ y Θ es una matriz de $n + m \times n + m$ parámetros de la forma

  $$\Theta = \begin{bmatrix} \Theta_{xx} & \Theta_{xa} \\ \Theta_{xa}^T & \Theta_{aa} \end{bmatrix},$$

  tal que la matriz Θ_{aa} de $m \times m$ es definida negativa.
- La política ávida se encuentra entonces como

  $$a_M = \pi_{\text{ávida}}(x_t, \Theta) = \arg\max_{a \in A} f(x_t, a, \Theta) = -\Theta_{uu}^{-1} \Theta_{xa}^T x_t.$$

- Los parámetros se pueden actualizar utilizando una representación matricial definiendo

  $$\nabla_\theta f(x_t, a_t\theta) = z_t^T z_t.$$

- La actualización de los parámetros será entonces

  $$\Theta \leftarrow \Theta + \alpha\Big(r(x_{t+1}, a_t) + \big(\gamma z_{t+1}^T \Theta z_{t+1} - z_t^T \Theta z_t\big) z_t^T z_t\Big).$$