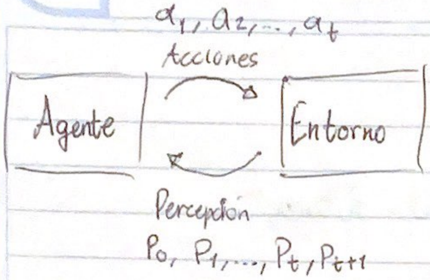


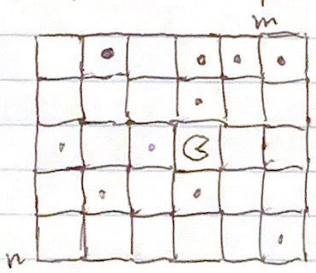
P Performance — Estado S

E Entorno — $S = (S_1, S_2, \dots, S_n) \in D_1 \times D_2 \times \dots \times D_n$

A Actuadores — esto significa

S Sensores — Mi espacio es un vector con diferentes valores que pertenecen a diferentes dominios.

Ejemplo de espacio estado (Pac-man)



$S = (f, c, i_0, \dots, i_{nm-1})$

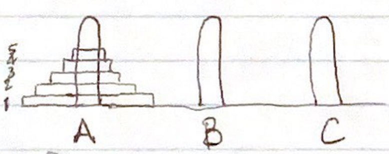
i_k = indicador si hay punto en la posición $\quad c = k // m$, $f = k \div m$

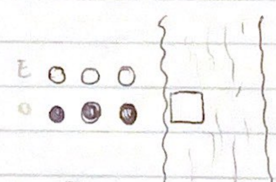
$f \in \{0, 1, \dots, n-1\}$

$c \in \{0, 1, \dots, m-1\}$

$i_k \in \{0, 1\} \ \forall k = 0, \dots, m \times n - 1$

Actividad en clase (Juegos)





$S = [i_{11}, \dots, i_{15}, i_{21}, \dots, i_{25}, i_{31}, \dots, i_{35}]$

$i_{a,b} \in \{0, 1\}$

$|S| = 2^{15} = 32 \times 10^3$

$S = [S_1, \dots, S_5]$

$S_i \in \{A, B, C\}$

$|S| = 3^5 = 243$

$S = [E, O, L]$

$E = \{0, 1, 2, 3\}$

$O = \{0, 1, 2, 3\}$

$L = \{0, 1\}$

$|S| = 4 \times 4 \times 2 = 32$

~~Tipos de estados~~ (~~características~~)

Determinista, Estático, Observable, Discreto

$$\underset{\text{estado}}{S} = f(\underset{\text{acción}}{a})$$

Podemos tener...

$a \in A$ ← conjunto de acciones

o también...

$a \in A(S)$ acciones legales en S

Determinista, Dinamico, Observable, Discreto

$$S_{K+1} = f(S_K, A_K)$$

Determinista, Dinamico, Parcialmente observable, Discreto

$$S_{K+1} = f(S_K, A_K)$$

$$P_K = g(S_K)$$

Espacio continuo:

$$\dot{x}_t = f(x_t, a_t)$$
$$P_t = g(x_t)$$

Cuando se tiene un problema donde no hay certeza de lo que va a pasar es estocástico...

Estocástico, Discreto, Estatico

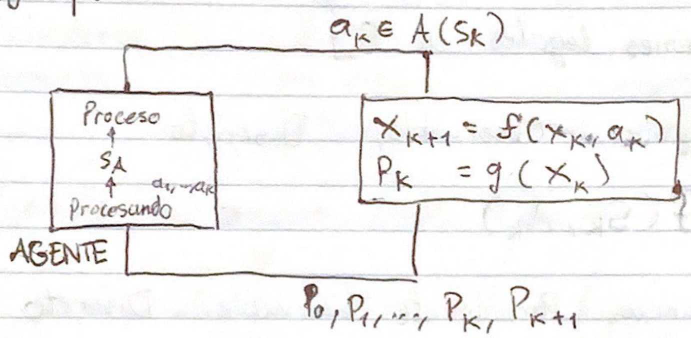
$$P_r[S/a] = \begin{bmatrix} P_r[S = s^{[1]} \mid a] \\ \vdots \\ P_r[S = s^{[a]} \mid a] \end{bmatrix}$$

Estocástico, Discreto, Dinamico

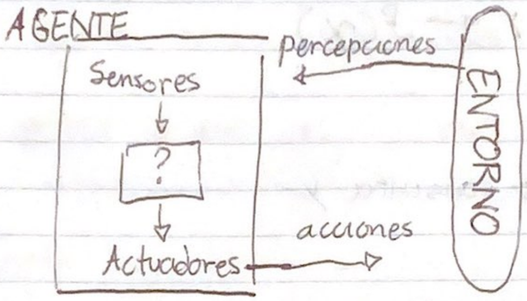
$$\Pr[S_{k+1} \mid s_k, a_k] = \begin{bmatrix} \Pr[S_{k+1} = s^{(1)} \mid s_k, a_k] \\ \vdots \\ \Pr[S_{k+1} = s^{(n)} \mid s_k, a_k] \end{bmatrix}$$

$$-\min(-a, -b) = \max(a, b)$$

Ejemplo:

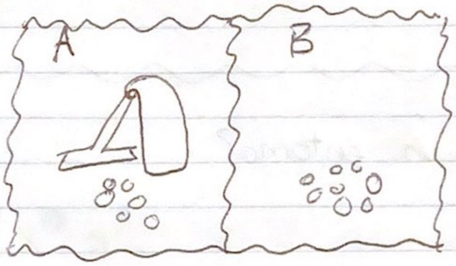


Intelligent Agents



Agente incluye humanos, robots, etc.

Ejemplo: Aspiradora



- Percepciones: Ubicación y contenidos
- Acciones: Izquierda, derecha, aspira y no-operar

$S = (A, B, R)$

actuadores

acciones

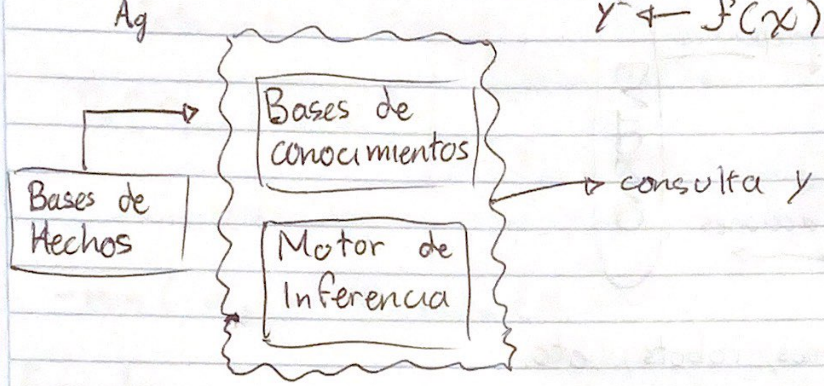
$A \in \{L, S\}$ $\quad$ $A = \{\leftarrow, \rightarrow, \text{aspira}, \text{nada}\}$

$B \in \{L, S\}$ $\quad$ percepciones

$R \in \{A, B\}$ $\quad$ $P = (R, S[R])$

Secuencia de percepción	Acción
A, Limpio	Derecha
A, Sucio	Aspira
B, Limpio	Izquierda
B, Sucio	Aspira
[A, Limpio], [A, Limpio]	
[A, Limpio], [A, Sucio]	
⋮	

$Y \leftarrow f(x)$



Esto ayuda para resolver problemas en sistemas reactivos con estados.

¿Qué se necesita para crear un entorno?

Entorno

S = Estado

$S = (S_1, \dots, S_n) \in D_1 \times \dots \times D_n = S$

$a_t \in A = \{a_1, \dots, a_m\}$

$a_t \in A(S_t) \rightarrow$ acciones legales

$p_t \in P \longrightarrow$ espacio de percepciones.

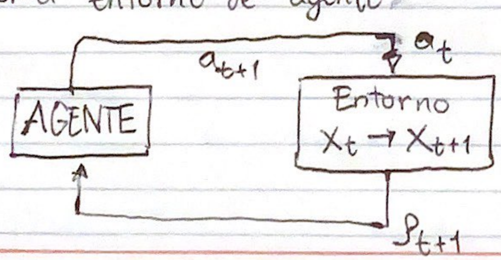
P_t = percepción (S_t)

percepción: $S \rightarrow P$

S_{t+1} = transición (S_t, a_t)

costo (S_t, a_t, S_{t+1})

Sistema entorno de agente:



El agente recibe percepciones y retorna acciones.

[En python se puede usar "_" para variables anonimas]

Agente reactivo

f_A es desconocida

$f_A : X \to Y$ ---- Y es el espacio de salida

$p \to \boxed{f_A(p)} \to a$

tipicamente $X = \mathbb{R}^n$

$x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$

si $y = \mathbb{R}$ --- Regresión.

si $y = \{-1, 1\}$ --- Clasificación Binaria.

si $y = \{C1, C2, \dots, C_K\}$ Clasificación en varias clases.

$\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ una muestra de X de dist. desconocida

$$D = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$$

donde $y^{(i)} = f_A(x^{(i)}) + \hat{e}$ v. A. dist. desconocida

El problema es encontrar una función $h^* : X \to Y$ tal que $h^* \approx f_A$

$h^* \in \mathcal{H}$ hipótesis posibles

$$\mathcal{H} = \{h_\alpha \mid h_\alpha : \mathbb{R} \to \mathbb{R} \text{ donde } h_\alpha(x) = \alpha x, \ \forall \alpha \in \mathbb{R}\}$$

$X = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}$

$$\underbrace{\begin{bmatrix} x_1^{(1)} & \cdots & x_R^{(1)} \\ x_1^{(2)} & \cdots & x_R^{(2)} \\ \vdots & & \\ x_1^{(m)} & \cdots & x_R^{(m)} \end{bmatrix}}_{X} \quad \underbrace{\begin{bmatrix} \\ \\ \\ \\ \end{bmatrix}}_{Y}$$

$h: X \times H \to Y$

parámetros: $h(x^{(i)}, \theta) = \hat{y}^{(i)}$

o también se puede escribir como (es equivalente):

$h_\theta(x^{(i)}) = \hat{y}^{(i)}$

Si θ es un vector de parámetros fijo

$h_\theta : X \to Y$

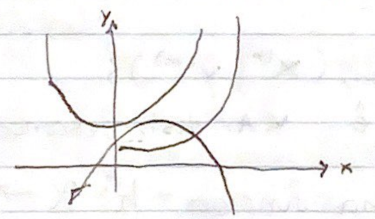
Ejemplo:

$h_w(x) = \sum_{j=1}^{T} w_j x^j + w_0, \quad x^J \in \mathbb{R}$

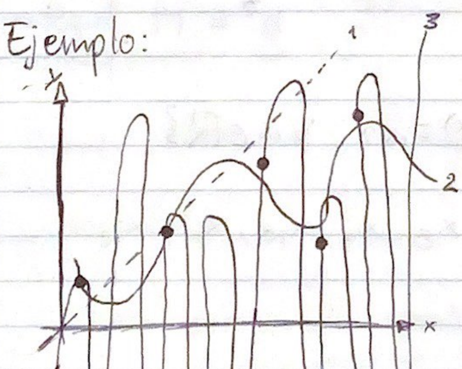
$h_w : \mathbb{R} \to \mathbb{R}$

$w = (w_0, w_1, \dots, w_T) \in \mathbb{R}^{T+1}$

$\hat{y} = w_1 x + w_0 \qquad \hat{y} = w_2 x^2 + w_1 x + b$



Ejemplo:



El 2 es el mejor ya que se aproxima lo más posible a todos los puntos sin ser muy complejo, como el 3 que aunque toque exactamente todos los puntos es demasiado complejo.

$\circledH = \Theta$

Hipótesis

1.- $f: X \to Y$ existe y es desconocida.

2.- Tengo $\{X^{(1)}, \dots, X^{(m)}\} \subseteq X$ una muestra de m datos con distribución desconocida.

3.- $D = \{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$, donde

$y^{(i)} = f(x^{(i)}) + e^{(i)}$ donde $e^{(i)}$ son VA

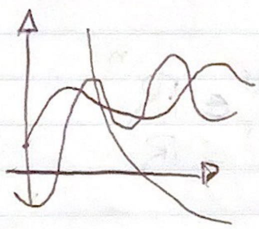
4.- Tenemos una función "parametrizada" $h: X \times \Theta \to Y$, típicamente $\Theta = \mathbb{R}^p$

5.- Para un valor específico de Θ, $h_\Theta: X \to Y$

6.- $h_\Theta \in \mathcal{H}$ conjunto de hipótesis

7.- El aprendizaje supervisado consiste en encontrar $h^* \in \mathcal{H}$ tal que

$$h^* \approx f$$



$L: Y \times Y \to \mathbb{R}$ función de perdida

$L(y, \hat{y}) = L(f(x), h^*(x))$

$L(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$ $\quad y \in \mathbb{R}$

$= |y - \hat{y}|$

$\left|\frac{y - \hat{y}}{y}\right|$

$$L(y, \hat{y}) = \begin{cases} 0 & \text{si } y = \hat{y} \\ 1 & \text{si } y \neq \hat{y} \end{cases} = \vec{1}\{y \neq \hat{y}\} \quad y \in \{-1, 1\}$$

$$E_{out}(h^*) = E_{x \in X}[L(f(x), h^*(x))]$$

E_{out} = Es el error fuera de muestra

$f \approx h^*$ si y solo si $E_{out}(h^*) \approx 0$

$$E_{in}(h^*) = \frac{1}{M} \sum_{i=1}^{M} L(y^{(i)}, h(x^{(i)}))$$

E_{in} = Es el error en muestra

$f \approx h^*$ si $\nearrow E_{in}(h^*) \approx 0$, $\searrow E_{in}(h^*) \approx E_{out}(h^*)$

Nota = Θ es un vector de parametros, ej:

$h_\Theta(x) = w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + b \qquad \Theta = (w_1, w_2, w_3, b) \in \mathbb{R}^4$

Desigualdad de Hueffding (PAC- Learning)

$$P_{\sigma}(|E_{out}(h^*) - E_{in}(h^*)| > \varepsilon) \leq \frac{1}{2} e^{-2\varepsilon^2 M}$$

donde M es el tamaño de la muestra

$d_{VC}(\mathcal{H}) \approx \#$ parámetros INDEPENDIENTES

EL APRENDIZAJE ES POSIBLE SI

$$10 * d_{VC}(\mathcal{H}) << M \leftrightarrow E_{in}(h^*) \approx E_{out}(h^*)$$

$$\underset{\text{Instancias}}{\begin{bmatrix} X^{(1)T} \\ X^{(2)T} \\ \vdots \\ X^{(m)T} \end{bmatrix}} \rightarrow \underset{\substack{X \\ (m,n)}}{\begin{array}{cccc} a_1 & a_2 & \cdots & a_n \\ \hline (X_1^{(1)} & X_2^{(1)} & \cdots & X_n^{(1)}) \\ (X_1^{(2)} & X_2^{(2)} & \cdots & X_n^{(2)}) \\ \vdots & \vdots & & \vdots \\ (X_1^{(m)} & X_2^{(m)} & \cdots & X_n^{(m)}) \end{array}} \qquad \underset{\substack{Y \\ (m,1)}}{\begin{array}{c} y \\ \hline y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{array}} \qquad \begin{array}{l} X^{(i)} \in \mathbb{R}^n \\ y^{(i)} \in \mathbb{R} \end{array}$$

$D = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$

conjunto de aprendizaje

$$h_\theta(x) = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b = \sum_{j=1}^{n} w_j x_j + b = w^T x + b$$
$$= x^T w + b$$

si $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$
$w = (w_1, \ldots, w_n) \in \mathbb{R}^n$

$\theta = (w_1, \ldots, w_n, b) \in \mathbb{R}^{n+1}$

Aprendizaje:

$E_{in}(h^*) \approx E_{out}(h^*)$

$E_{in}(h^*) \approx 0$

$$h^* = \arg\min_{h \in \mathcal{H}} E_{in}(h) \leftrightarrow \theta^* = w^*, b^* = \arg\min_{\substack{w \in \mathbb{R}^n \\ b \in \mathbb{R}}} E_{in}(h_{w,b})$$

$$\omega^*, b^* = \arg \min_{\substack{\omega \in \mathbb{R}^n \\ b \in \mathbb{R}^n}} \frac{1}{M} \sum_{i=1}^{m} \frac{1}{2} \left(y^{(i)} - h_{\omega, b}(x^{(i)})\right)^2 \quad \text{MSE}$$

Mean Square Error

$$\omega^*, b^* = \arg \min_{\substack{\omega \in \mathbb{R}^n \\ b \in \mathbb{R}}} \frac{1}{M} \sum_{i=1}^{m} \frac{1}{2} \left(y^{(i)} - \underbrace{\omega^T x^{(i)} - b}_{= x^{(i)T} \omega - b}\right)^2$$

$$= \arg \min_{\substack{\omega \in \mathbb{R}^n \\ b \in \mathbb{R}}} \frac{1}{2M} \left[(y^{(1)} - x^{(1)T}\omega - b) \ldots (y^{(m)} - x^{(m)T}\omega - b) \right] \begin{bmatrix} y^{(1)} - x^{(1)T}\omega - b \\ y^{(2)} - x^{(2)T}\omega - b \\ \vdots \\ y^{(m)} - x^{(m)T}\omega - b \end{bmatrix}$$

$$= \arg \min_{\omega \in \mathbb{R}^n} \frac{1}{2M} \left[Y - [X, \vec{1}]\, \theta \right]^T \left[Y - [X, \vec{1}]\, \theta \right]$$

$$\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix} - \begin{bmatrix} x^{(1)T} \\ x^{(2)T} \\ \vdots \\ x^{(m)T} \end{bmatrix} \omega - \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} b$$

Para encontrar un mínimo de una función: Se deriva y se iguala a cero

En este caso como tenemos dos variables tenemos que derivar parcialmente, para encontrar el gradiente.

Ejemplo: $f(x_1, \ldots, x_n)$

$f(y) \quad x \in \mathbb{R}^n, \quad f: \mathbb{R}^n \to \mathbb{R}$

$$\nabla_x f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix}$$

Regresamos al problema...

$$[Y - X_e\theta]^T [Y - X_e\theta]$$

$$\frac{\cancel{2}}{\cancel{2}M}[Y - X_e\theta]_{m,1}$$

$$\frac{\partial (Y - Kx)^2}{\partial x} = -2(Y - Kx)K$$

$$\frac{1}{M} X_e^T [Y - X_e\theta] = \vec{0}$$

$$X_e^T Y - X_e^T X_e \theta = \vec{0}$$

$$X_e^T X_e \theta = X_e^T Y$$

$$\boxed{\theta = \underbrace{[X_e^T X_e]^{-1} X_e^T}_{\substack{\text{pinv}(X_e) \\ \text{pseudo - inversa}}} Y}$$

← Este es el método mínimo de cuadrados.