MODELO PAC GENERALIZADO Y PREDICTORES LINEALES PROBLEMAS

ESTUARDO A. DÍAZ H., 16110

- **(Cap 3, Ejercicio 4)** Notemos que nuestra clase de hipótesis consiste de todas las posibles conjunciones de d variables. Cada una de estas variables puede estar en la conjunción, estar negada en la conjunción o no estar en la conjunción. Por lo tanto, $|H| = 3^d$, i.e. $|H|$ es finita. Ya demostramos en clase que todas las clases de hipótesis finitas son PAC aprendibles, por lo que H es PAC aprendible.
  Ahora, el algoritmo propuesto es el siguiente: Seleccionamos los elementos $(x_i, y_i) \in S$ tales que $y_i = 1$, $1 \leq i \leq m$. Luego, para cada variable x_{*_j}, $1 \leq j \leq d$, revisamos si $x_{i_j} = 1, \forall i$ y se se cumple, entonces agregamos x_{*_j} en la conjunción. Notemos que a lo sumo, revisamos m elementos de la muestra, y como revisamos cada una de las d coordenadas, a lo sumo realizamos dm pasos.

- **(Cap 3, Ejercicio 7)**

  Demostración. Recordemos la definición de Bayes,

$$f(x) = \begin{cases} 1 & \mathbb{P}(y = 1|x) \geq \frac{1}{2} \\ 0 & \mathbb{P}(y = 1|x) < \frac{1}{2} \end{cases}$$

  A probar: $\forall g, g : X \to Y$, $L_D(f_D) \leq L_D(g)$.
  Sabemos que $L_D(h) = \underset{(x,y)\sim D^m}{\mathbb{P}}[h(x) \neq y] = D(\{(x, y) : h(x) \neq y\}) = \mathbb{P}[h(x) \neq y|X = x]\mathbb{P}[X = x]$
  Luego,

$$\mathbb{P}[h(x) \neq y|X = x] = \mathbb{P}[y = 1|X = x]\mathbb{P}[h(x) = 0|X = x] + \mathbb{P}[y = 0|X = x]\mathbb{P}[h(x) = 1|X = x]$$

  Para Bayes,

$$\mathbb{P}[f_D(x) \neq y|X = x] = \mathbb{P}[y = 1|X = x]\mathbb{P}[f_D(x) = 0|X = x] + \mathbb{P}[y = 0|X = x]\mathbb{P}[f_D(x) = 1|X = x]$$

  entonces $\mathbb{P}[f_D(x) \neq y|X = x] = min\{\mathbb{P}[y = 1|X = x], 1 - \mathbb{P}[y = 1|X = x]\}$.
  Para cualquier otro predictor,

$$\begin{aligned}
&\mathbb{P}[h(x) \neq y|X = x] \\
&= \mathbb{P}[y = 1|X = x]\mathbb{P}[h(x) = 0|X = x] + \mathbb{P}[y = 0|X = x]\mathbb{P}[h(x) = 1|X = x] \\
&\geq min\{\mathbb{P}[y = 1|X = x], 1 - \mathbb{P}[y = 1|X = x]\}[\mathbb{P}[h(x) = 0|X = x] + \mathbb{P}[h(x) = 1|X = x]] \\
&= min\{\mathbb{P}[y = 1|X = x], 1 - \mathbb{P}[y = 1|X = x]\} \\
&= L_D(f_d)
\end{aligned}$$

  □

- **(Cap 4, Ejercicio 1)** Dada una distribución D, y un algoritmo A, queremos demostrar que los siguientes son equivalentes:
  1. Dados $\epsilon, \delta > 0, \exists m(\epsilon, \delta) \ni \forall m > m(\epsilon, \delta)$,

$$\underset{S\sim D^m}{\mathbb{P}}[L_D(A(S)) > \epsilon] < \delta$$

  2.

$$\lim_{n\to\infty} \underset{S\sim D^m}{\mathbb{E}}[L_D(A(S))] = 0$$

Demostración. ($\Longrightarrow$) $\mathop{\mathbb{E}}_{S\sim D^m}[L_D(A(S))] = \mathop{\mathbb{P}}_{S\sim D^m}[L_D(A(S)) > \epsilon]\mathop{\mathbb{E}}_{S\sim D^m}[L_D(A(S))|L_D(A(S)) > \epsilon] + \mathop{\mathbb{P}}_{S\sim D^m}[L_D(A(S)) \leq \epsilon]\mathop{\mathbb{E}}_{S\sim D^m}[L_D(A(S))|L_D(A(S)) \leq \epsilon] \leq \epsilon + \delta \leq 2max\{\epsilon, \delta\}$.
Dado $\epsilon' > 0$, tomamos $\epsilon = \delta = \epsilon'/2$. Entonces, $\exists m(\epsilon'/2, \epsilon'/2) \ni$ si $m > m(\epsilon'/2, \epsilon'/2) \Longrightarrow$

$$\mathop{\mathbb{E}}_{S\sim D^m}[L_D(A(S))] < \epsilon'$$

($\Longleftarrow$) Sean $\epsilon, \delta > 0$. Sea $\epsilon' = \epsilon\delta$. Entonces $\exists M \ni$ si $m > M$,

$$\mathop{\mathbb{E}}_{S\sim D^m}[L_D(A(S))] < \epsilon\delta$$

Supongamos que $\mathop{\mathbb{P}}_{S\sim D^m}[L_D(A(S)) > \epsilon] > \delta$ entonces

$$\mathop{\mathbb{E}}_{S\sim D^m}[L_D(A(S))] > \epsilon\delta(\rightarrow\leftarrow)$$

□

- **(Cap 9, Ejercicio 3)** Perceptrones: Queremos demostrar que para cada $m \in \mathbb{Z}^+, \exists w* \in \mathbb{R}^d$ y una secuencia de ejemplos $\{(x_1, y_1), \dots (x_m, y_m)\}$ se cumple lo siguiente:
  - $R = max_i||x_i|| \leq 1$
  - $||w*||^2 = m$ y para todo $i \leq m$, $y_i < x_i, w* >\geq 1$.
  - Cuando realizamos el algoritmo del perceptron, en esta secuencia de ejemplos, se actualiza m veces.

  Consideremos $d = m$ y para cada i, sea $x_i = e_i$. Entonces $R = max_i||x_i|| = max_i||e_i|| = max_i\{1\} = 1 \leq 1$.
  Luego, obtenemos $w*$ de la siguienta manera: Sea $w*_0 = (e_1, \dots, e_d)$, luego para cada $i = 1, \dots, m$, si $y_i < x_i, w*_i >= -1$ entonces $w*_i$ es igual a $w*_{i-1}$ con la i-ésima coordenada reemplazada por $-e_i$. Entonces $||w*||^2 =< w*, w* >= 1 + \cdots + 1 = m$. Para $i \leq m$, $y_i < x_i, w*_i >= 1 \geq 1$ por construcción. Entonces tenemos que $min\{||w|| : \forall i \in [m], y_i < w, x_i >\geq 1\} \leq \sqrt{m}$. Finalmente, el algoritmo propuesto solo necesita m pasos para converger.