Aprendamos sobre Learnability

Pun intended

Guillermo (Billy) Mosse

Exactas, UBA

¿Qué queremos definir?

¿Se puede inferir una función de manera computable a partir de finitas entradas?

¿Qué necesitamos para responder esa pregunta?

- ¿De qué bolsa estamos sacando la función?
  Por ejemplo, si queremos aprender una función f que **sabemos** que es lineal, se puede aprender con solo 2 datos.

¿Qué necesitamos para responder esa pregunta?

- ¿De qué bolsa estamos sacando la función?
  Por ejemplo, si queremos aprender una función f que **sabemos** que es lineal, se puede aprender con solo 2 datos.
- ¿Cuándo/Cuánto queremos aprender a la función? No es lo mismo aprenderla luego de n pasos con n prefijado que en el límite, o de manera probabilística.

Learnability in the limit

Notación: Dada f una función, notamos f^n a la n-upla de pares $[< x_1, f(x_1) >, \cdots, < x_n, f(x_n) >]$, es decir, a un sample de tamaño n de la función f.

Learnability in the limit

Notación: Dada f una función, notamos f^n a la n-upla de pares $[< x_1, f(x_1) >, \cdots, < x_n, f(x_n) >]$, es decir, a un sample de tamaño n de la función f.

Definición

Una clase C de funciones total computables se dice que **se puede aprender en el límite** ("learnable in the limit") $\Leftrightarrow$ existe una función computable total $g : \mathbb{N} \to \mathbb{N}$ llamada **aprendedora** ("~~Alejandro~~ Learner") tal que $\forall\ f \in C\ \exists\ n_f \in \mathbb{N}$ tal que:

Learnability in the limit

<u>Notación</u>: Dada f una función, notamos f^n a la n-upla de pares $[< x_1, f(x_1) >, \cdots, < x_n, f(x_n) >]$, es decir, a un sample de tamaño n de la función f.

Definición

Una clase C de funciones total computables se dice que **se puede aprender en el límite** ("learnable in the limit") $\Leftrightarrow$ existe una función computable total $g : \mathbb{N} \to \mathbb{N}$ llamada **aprendedora** ("~~Alejandro~~ Learner") tal que $\forall\ f \in C\ \exists\ n_f \in \mathbb{N}$ tal que:

- $\phi_{g(f^{n_f})}$ computa a f

Learnability in the limit

<u>Notación</u>: Dada f una función, notamos f^n a la n-upla de pares $[< x_1, f(x_1) >, \cdots, < x_n, f(x_n) >]$, es decir, a un sample de tamaño n de la función f.

Definición

Una clase C de funciones total computables se dice que **se puede aprender en el límite** ("learnable in the limit") $\Leftrightarrow$ existe una función computable total $g : \mathbb{N} \to \mathbb{N}$ llamada **aprendedora** ("~~Alejandro~~ Learner") tal que $\forall\ f \in C\ \exists\ n_f \in \mathbb{N}$ tal que:

- $\phi_{g(f^{n_f})}$ computa a f
- $g(f^n) = g(f^{n_f})\ \forall n\ \geq n_f$

Learnability in the limit

Notación: Dada f una función, notamos f^n a la n-upla de pares $[< x_1, f(x_1) >, \cdots, < x_n, f(x_n) >]$, es decir, a un sample de tamaño n de la función f.

Definición

Una clase C de funciones total computables se dice que **se puede aprender en el límite** ("learnable in the limit") $\Leftrightarrow$ existe una función computable total $g : \mathbb{N} \to \mathbb{N}$ llamada **aprendedora** ("~~Alejandro~~ Learner") tal que $\forall f \in C \; \exists \; n_f \in \mathbb{N}$ tal que:

- $\phi_{g(f^{n_f})}$ computa a f
- $g(f^n) = g(f^{n_f}) \; \forall n \geq n_f$

Denotamos $\mathcal{LIM}$ como el conjunto de las clases que se pueden aprender en el límite.

Learnability in the limit

Notación: Dada f una función, notamos f^n a la n-upla de pares $[< x_1, f(x_1) >, \cdots, < x_n, f(x_n) >]$, es decir, a un sample de tamaño n de la función f.

Definición

Una clase C de funciones total computables se dice que **se puede aprender en el límite** ("learnable in the limit") $\Leftrightarrow$ existe una función computable total $g : \mathbb{N} \to \mathbb{N}$ llamada **aprendedora** ("~~Alejandro~~ Learner") tal que $\forall\ f \in C\ \exists\ n_f \in \mathbb{N}$ tal que:

- $\phi_{g(f^{n_f})}$ computa a f
- $g(f^n) = g(f^{n_f})\ \forall n \geq n_f$

Denotamos $\mathcal{LIM}$ como el conjunto de las clases que se pueden aprender en el límite.

Obs: podríamos pedir que el sample no venga ordenado.

¿Qué hay en $\mathcal{LIM}$?

Trivialmente, $\forall f$ computable, $\{f\} \in \mathcal{LIM}$ vía $g(n) \equiv e$, donde e es el número de programa de f.

¿Qué hay en $\mathcal{LIM}$?

Trivialmente, $\forall f$ computable, $\{f\} \in \mathcal{LIM}$ vía $g(n) \equiv e$, donde e es el número de programa de f.

Ejercicio: Si C es un conjunto finito de funciones computables, $C \in LIM$

¿Qué hay en $\mathcal{LIM}$?

Trivialmente, $\forall f$ computable, $\{f\} \in \mathcal{LIM}$ vía $g(n) \equiv e$, donde e es el número de programa de f.

Ejercicio: Si C es un conjunto finito de funciones computables, $C \in LIM$

¡Lo interesante es ver clases infinitas de funciones computables!

Ajustemos la definición

Definición

Vamos a decir que una clase de funciones totales C se puede t-aprender en el límite si se puede aprender en el límite vía una función g (total, como antes) tal que $\phi_{g(n)}$ es total $\forall$ n. Llamamos $\mathcal{RTOTAL}$ al conjunto de clases que se pueden t-aprender.

Ajustemos la definición

Definición

Vamos a decir que una clase de funciones totales C se puede t-aprender en el límite si se puede aprender en el límite vía una función g (total, como antes) tal que $\phi_{g(n)}$ es total $\forall$ n. Llamamos $\mathcal{RTOTAL}$ al conjunto de clases que se pueden t-aprender.

Obs: $\mathcal{RTOTAL} \subset \mathcal{LIM}$ (vía la misma g)

Caracterización de $\mathcal{RTOTAL}$

Definición

C es computablemente enumerable como conjunto de funciones si existe una función "enumeradora" computable ψ, tal que $\forall\ f \in C\ \exists\ i$ tal que $f = \psi_i$

Caracterización de $\mathcal{RTOTAL}$

Definición

C es computablemente enumerable como conjunto de funciones si existe una función "enumeradora" computable ψ, tal que $\forall\ f \in C\ \exists\ i$ tal que $f = \psi_i$

Teorema

Dada una clase de funciones totales C, $C \in \mathcal{RTOTAL} \Leftrightarrow C$ es un subconjunto de **una familia de funciones totales** computablemente enumerable.

Caracterización de $\mathcal{RTOTAL}$

Definición

C es computablemente enumerable como conjunto de funciones si existe una función "enumeradora" computable ψ, tal que $\forall\ f \in C\ \exists\ i$ tal que $f = \psi_i$

Teorema

Dada una clase de funciones totales C, $C \in \mathcal{RTOTAL} \Leftrightarrow C$ es un subconjunto de **una familia de funciones totales** computablemente enumerable.

dem: ejercicio (?) Idea:

- Si C se puede aprender, la función g learner me genera el superset enumerable de C.
- Si C está generado por ψ, la hipótesis que devuelvo para entrada será la primera función que me genera ψ compatible con esa entrada. Eso casi que funciona.

Imagen que le pedí prestada a Gaby Senno

Para la vuelta:

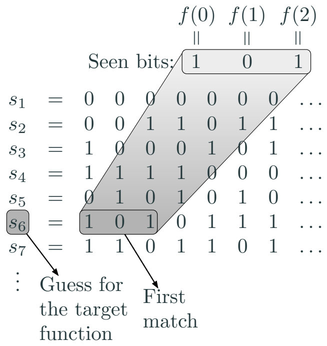

Figure 1: Luego de ver $f(0) = 0, f(1) = 0, f(2) = 1$, la función learner g devuelve $\psi(i)$,i.e, $\phi_{g(f^3)}(x) = \phi_{\psi(i)}(x)$

Funciones acotadas temporalmente por una T

Definición

Dada T una función computable (total), decimos que f es computable en tiempo $O(T(n))$ si para casi todo n f tarda a lo sumo $T(n)$ pasos en computar $T(n)$.

Funciones acotadas temporalmente por una T

Definición

Dada T una función computable (total), decimos que f es computable en tiempo $O(T(n))$ si para casi todo n f tarda a lo sumo $T(n)$ pasos en computar $T(n)$.

Proposición

Dada T como arriba, la clase de funciones C_T computable en tiempo $O(T(n))$ es computablemente enumerable (¡en el sentido de funciones!)

Funciones acotadas temporalmente por una T

Definición

Dada T una función computable (total), decimos que f es computable en tiempo $O(T(n))$ si para casi todo n f tarda a lo sumo $T(n)$ pasos en computar $T(n)$.

Proposición

Dada T como arriba, la clase de funciones C_T computable en tiempo $O(T(n))$ es computablemente enumerable (¡en el sentido de funciones!)

dem: la idea es dar para cada i una función f que para casi toda entrada n tarde menos de $T(n)$ pasos en computarla.

Funciones acotadas temporalmente por una T

Definición

Dada T una función computable (total), decimos que f es computable en tiempo $O(T(n))$ si para casi todo n f tarda a lo sumo $T(n)$ pasos en computar $T(n)$.

Proposición

Dada T como arriba, la clase de funciones C_T computable en tiempo $O(T(n))$ es computablemente enumerable (¡en el sentido de funciones!)

dem: la idea es dar para cada i una función f que para casi toda entrada n tarde menos de $T(n)$ pasos en computarla.

Idea:

$\psi(i)(x)$ va a hacer lo siguiente: corre el programa ϕ_i con entrada 1 hasta tiempo $T(1)$. Si no termina, devuelve 0 (algún número de programa de

(sigue la demo)

Si termina, corre ϕ_i con entrada k hasta tiempo $T(k)$ $\forall$ $k \leq x$. Si en algún momento tardo se cumple que $\phi_i(k)$ no terminó a tiempo $T(k)$ devuelve 0.

Si $\forall$ k $\phi_i(k)$ tarda el tiempo correcto, devuelve (el nro de) $\phi_i(x)$.

¿Por qué esto es (casi)* correcto?

$\forall$ i, si $\phi_i \notin C_T$, entonces ψ_i es 0 ctp (y por lo tanto está en C_T). Y si $\phi_i \in C_T$, entonces $\psi_i \equiv \phi_i$.

*Una sutileza que no estoy tratando: por simplicidad, no genero las funciones que en finitas entradas se pasan de tiempo. Eso se arregla cambiando $\psi(i)$ por una $\psi(i, j)$ acorde. $\square$

Resultado fuerte

Teorema

Teorema: Si T es una función computable, la clase de funciones (totales) computable en tiempo O(T(n)) está en $\mathcal{RTOTAL}$.

Resultado fuerte

Teorema

Teorema: Si T es una función computable, la clase de funciones (totales) computable en tiempo O(T(n)) está en $\mathcal{RTOTAL}$.

dem: recién probamos que es computablemente enumerable. $\square$

Corolarios copados

Las siguientes clases de funciones están en $\mathcal{RTOTAL}$:

- PR (porque son más lentas que Ackerman)
- P, acotando por cualquier exponencial
- NP (¡en serio!):
  $NP \subsetneq EXSPACE = O(2^{p(n)})space \subsetneq O(2^{e^n})space \subset O(2^{(2^{e^n})})time$
- BQP*: $BQP \subset EXPTIME = O(2^{p(n)}) \subsetneq O(2^{e^n})$
- PSPACE

*donde BQP es la clase de problemas de decisión que se pueden resolver con una computadora cuántica en tiempo polinomial con una probabilidad de error siempre a lo sumo 1/3.

Existen clases que no se pueden aprender

Teorema

Teorema: la clase de funciones computables totales TOT no está en $\mathcal{RTOTAL}$

Existen clases que no se pueden aprender

Teorema

Teorema: la clase de funciones computables totales TOT no está en $\mathcal{RTOTAL}$

dem: argumento diagonal! Supongamos que TOT está en $\mathcal{RTOTAL}$ vía la función $g : \mathbb{N} \to \mathbb{N}$.

Defino $f : \mathbb{N} \to \mathbb{N}$ como:

$$f(0) := 0$$

$$f(n+1) := \phi_{g(f^n)}(n+1) + 1$$

Claramente f y $\phi_{g(f^n)}$ nunca van a coincidir, para ninguna entrada f^n (basta mirar la entrada $n+1$ para cada n).

Existen clases que no se pueden aprender

Teorema

Teorema: la clase de funciones computables totales TOT no está en $\mathcal{RTOTAL}$

dem: argumento diagonal! Supongamos que TOT está en $\mathcal{RTOTAL}$ vía la función $g : \mathbb{N} \to \mathbb{N}$.

Defino $f : \mathbb{N} \to \mathbb{N}$ como:

$$f(0) := 0$$

$$f(n+1) := \phi_{g(f^n)}(n+1) + 1$$

Claramente f y $\phi_{g(f^n)}$ nunca van a coincidir, para ninguna entrada f^n (basta mirar la entrada $n+1$ para cada n).

f es total computable, es decir, está en R y f y $\phi_{g(f^n)}$ siempre van a diferir en el valor $n+1$. $\square$

¿Y qué pasa con la unión?

Si dos clases se pueden aprender, ¿se podrá aprender la unión?

Otra manera de aprender

Definición

Una clase C de funciones totales se dice predecible ~~o aburrida~~ si existe una función computable total S a la que vamos a llamar "eStrategia" tal que $S(f^n) = f(n+1) \forall f \in C$ y para casi todo n (le permitimos finitos errores).

O sea, pedimos poder predecir casi siempre el siguiente valor de una función.

Denotamos por NV (por "Next Value")

¡Teorema!

Teorema

$NV = \mathcal{RTOTAL}$

Otra manera más de aprender

Definición

Convergencia semántica en el límite con anomalías: decimos que una clase C se puede aprender correctamente en el límite, pero con a anomalías si $\exists g / \forall\ f\ \in\ C$,

- $\forall\ n\ \in\ \mathbb{N} g(n)$ está definida (esto no cambia)
- $\exists\ j \in \mathbb{N}$ tal que $\phi_{S(f^n)} = f\ \forall n \geq n_0$.

Definición

Convergencia semántica en el límite con anomalías: decimos que una clase C se puede aprender correctamente en el límite, pero con a anomalías si $\exists g / \forall\ f\ \in\ C$,

- $\forall\ n\ \in\ \mathbb{N} g(n)$ está definida (esto no cambia)
- $\exists\ j \in \mathbb{N}$ tal que $\phi_{S(f^n)} = f\ \forall n \geq n_0$.

Notación: $C \in \mathcal{BC}$

Otra manera más de aprender

Definición

Convergencia semántica en el límite con anomalías: decimos que una clase C se puede aprender correctamente en el límite, pero con a anomalías si $\exists g / \forall\ f\ \in\ C$,

- $\forall\ n\ \in\ \mathbb{N} g(n)$ está definida (esto no cambia)
- $\exists\ j \in \mathbb{N}$ tal que $\phi_{S(f^n)} = f\ \forall n \geq n_0$.

Notación: $C \in \mathcal{BC}$

Comentario: TOT$\notin \mathcal{BC}$. Además no es cerrado por uniones finitas.

Última

Notación: Dado $a \in \mathbb{N}, f =^a g$ sii $f(x) = g(x)\ \forall\ x \geq a$. Para a $=$ *, pedimos igualdad para todo x salvo finitos.

Última

Notación: Dado $a \in \mathbb{N}, f =^a g$ sii $f(x) = g(x) \; \forall \; x \geq a$. Para a = *, pedimos igualdad para todo x salvo finitos.

Definición

Convergencia en el límite con anomalías: decimos que una clase C se puede aprender correctamente en el límite, pero con a anomalías si $\exists \; g / \; \forall \; f \; \in \; C$,

- $\forall \; n \; \in \; \mathbb{N} g(n)$ está definida (esto no cambia)
- $\exists j \in \mathbb{N}/\phi_j =^a f$ y $S(f^n)$ converge a j.

Última

Notación: Dado $a \in \mathbb{N}, f =^a g$ sii $f(x) = g(x) \; \forall \; x \geq a$. Para a $=$ *, pedimos igualdad para todo x salvo finitos.

Definición

Convergencia en el límite con anomalías: decimos que una clase C se puede aprender correctamente en el límite, pero con a anomalías si $\exists \; g / \; \forall \; f \;\; \in \;\; C$,

- $\forall \; n \;\; \in \;\; \mathbb{N} g(n)$ está definida (esto no cambia)
- $\exists j \in \mathbb{N} / \phi_j =^a f$ y $S(f^n)$ converge a j.

Más notación: $C \in \mathcal{LIM}^a$.

Última

Notación: Dado $a \in \mathbb{N}, f =^a g$ sii $f(x) = g(x)\ \forall\ x \geq a$. Para a = *, pedimos igualdad para todo x salvo finitos.

Definición

Convergencia en el límite con anomalías: decimos que una clase C se puede aprender correctamente en el límite, pero con a anomalías si $\exists\ g/\ \forall\ f\ \in\ C$,

- $\forall\ n\ \in\ \mathbb{N} g(n)$ está definida (esto no cambia)
- $\exists j \in \mathbb{N}/\phi_j =^a f$ y $S(f^n)$ converge a j.

Observación: $\mathcal{LIM}^0 = \mathcal{LIM}$

Me encanta decir cosas sin demostrarlas

Teorema

$$\mathcal{LIM} \subsetneq \mathcal{LIM}^1 \subsetneq \mathcal{LIM}^2 \subsetneq \mathcal{LIM}^3 \subsetneq \ldots \bigcup_{a\in\mathbb{N}} \mathcal{LIM}^a \subsetneq \mathcal{LIM}^* \subsetneq BC$$

Me encanta decir cosas sin demostrarlas

Teorema

$$\mathcal{LIM} \subsetneq \mathcal{LIM}^1 \subsetneq \mathcal{LIM}^2 \subsetneq \mathcal{LIM}^3 \subsetneq \ldots \bigcup_{a \in \mathbb{N}} \mathcal{LIM}^a \subsetneq \mathcal{LIM}^* \subsetneq \mathcal{BC}$$

Teorema

$$\mathcal{BC} \subsetneq \mathcal{BC}^1 \subsetneq \mathcal{BC}^2 \subsetneq \mathcal{BC}^3 \subsetneq \ldots \bigcup_{a \in \mathbb{N}} \mathcal{BC}^a \subsetneq \mathcal{BC}^*$$

Me encanta decir cosas sin demostrarlas

Teorema

$$\mathcal{LIM} \subsetneq \mathcal{LIM}^1 \subsetneq \mathcal{LIM}^2 \subsetneq \mathcal{LIM}^3 \subsetneq \ldots \bigcup_{a\in\mathbb{N}} \mathcal{LIM}^a \subsetneq \mathcal{LIM}^* \subsetneq BC$$

Teorema

$$\mathcal{BC} \subsetneq \mathcal{BC}^1 \subsetneq \mathcal{BC}^2 \subsetneq \mathcal{BC}^3 \subsetneq \ldots \bigcup_{a\in\mathbb{N}} \mathcal{BC}^a \subsetneq \mathcal{BC}^*$$

Teorema

$TOT \in BC^*$

Fin

Preguntas?

 If Alan was alive, he'd be so happy about gay marriage that he would buy Turings! 

Right? Right?! ... I'll see myself out.