Aprendizaje Automático

Cuestionario 2

David Gil Bautista

45925324M

Cuestiones

1. Identificar de forma precisa dos condiciones imprescindibles para que un problema de predicción puede ser aproximado por inducción desde una muestra de datos. Justificar la respuesta usando los resultados teóricos estudiados.

En un problema de predicción aprendemos de unos datos y tratamos de predecir las etiquetas de unas nuevas entradas, usando la aproximación por inducción aprendemos reglas generales a partir de unos datos específicos.

Para usar esta aproximación han de cumplirse las siguientes condiciones:

1) $E_{in} \approx E_{out}$

2) $E_{in}(g) \approx 0$

Usando la desigualdad de Hoeffding sabemos que

$$\mathbb{P}[\mathcal{D} : |v - \mu| > \epsilon) \leq 2e^{-2\epsilon^2 N}$$
$$v = E_{in}, \quad \mu = E_{out}$$
$$\mathbb{P}[\mathcal{D} : |E_{in} - E_{out}| > \epsilon) \leq 2e^{-2\epsilon^2 N}$$

Para poder aplicar esto tenemos que tener una muestra de datos ($\mathcal{D}$) independiente e idénticamente distribuida. A pesar de que teniendo dicha muestra pueda parecer que no vamos a conseguir generalizar reglas que aplicaremos no es así, aunque tengamos datos distintos, podremos establecer relaciones con otros datos que se parecen más o menos.

Al tener un muestra independiente garantizamos que E_{in} alcanza todos los valores posibles, por lo que se cumple que $E_{in} \approx E_{out}$.

Para la segunda condición buscamos garantizar $E_{in} \approx 0$, para que esto se cumpla, la complejidad de $\mathcal{H}$ determina la factibilidad de esta condición. Cuando más compleja sea $\mathcal{H}$, más probable será que nuestro modelo ajuste bien, lo que garantizará un error mínimo. Sin embargo, para asegurar la primera condición, $\mathcal{H}$ no debe ser muy compleja.

Ahora tenemos que la desigualdad de Hoeffding garantiza la aproximación de los errores y que la complejidad del algoritmo garantiza obtener unos resultados de error cercanos a 0 en el train. Sabiendo ambas cosas podemos asegurar que obtendremos un bajo error en el test ($E_{out} \approx 0$).

2. El jefe de investigación de una empresa con mucha experiencia en problemas de predicción de datos tras analizar los resultados de los muchos algoritmos de aprendizaje usados sobre todos los problemas en los que la empresa ha trabajado a lo largo de su muy dilatada existencia, decide que para facilitar el mantenimiento del código de la empresa van a seleccionar un único algoritmo y una única clase de funciones con la que aproximar todas las soluciones a sus problemas presentes y futuros. ¿Considera que dicha decisión es correcta y beneficiará a la empresa? Argumentar la respuesta usando los resultados teóricos estudiados.

No, el teorema "No Free Lunch" nos dice que no hay un modelo que funcione mejor para cada problema. Cuando nos encontramos ante un problema y debemos encontrar un modelo que nos permita estudiarlo y solucionarlo, buscamos un modelo o una serie de modelos que se ajustan mejor al problema. A la hora de escoger el modelo nos basamos en la validación para elegir aquel que nos ofrece un mejor resultado. El restringir el el modelo a una única clase de funciones puede hacer que incluso para algunos problemas ni siquiera seamos capaces de encontrar una solución, debido a la complejidad del problema.

Una vez escogido el modelo, (en caso de que este nos permita encontrar la solución) debemos escoger el algoritmo con el que entrenar nuestro modelo. Varios algoritmos pueden ofrecer la misma solución pero variando la velocidad, precisión y complejidad. Por tanto, el escoger un único algoritmo puede hacer que para un problema muy sencillo se tarde una eternidad en encontrar la solución.

Concluyendo podemos afirmar que **si la empresa elige un sólo modelo y un algoritmo para resolver todos los problemas presentes y futuros, con muy alta probabilidad irá a pique.**

---

3. Supongamos un conjunto de datos D de 25 ejemplos extraidos de una funcion desconocida $f : X \to Y$, donde $X = R$ e $Y = -1, +1$. Para aprender f usamos un conjunto simple de hipótesis $\mathcal{H} = \{h_1, h_2\}$ donde h_1 es la función constante igual a +1 y h_2 la función constante igual a -1. Consideramos dos algoritmos de aprendizaje, **S**(smart) y **C**(crazy). **S** elige la hipótesis que mejor ajusta los datos y **C** elige deliberadamente la otra hipótesis.

a) ¿Puede S producir una hipótesis que garantice mejor comportamiento que la aleatoria sobre cualquier punto fuera de la muestra? Justificar la respuesta.

No, sin saber que la muestra es iid ni siquiera **podemos asegurar que el algoritmo S clasifique bien sobre la muestra de entrenamiento**. En el caso de la muestra aleatoria podría llegar a acertar en algún caso, por lo que sería más probable clasificar correctamente con esta.

---

4. Con el mismo enunciado de la pregunta.3:

a) Asumir desde ahora que todos los ejemplos en D tienen $y_n = +1$. ¿Es posible que la hipotesis que produce C sea mejor que la hipótesis que produce S ?. Justificar la respuesta.

Si, **dado** que ahora nuestro algoritmo S no aprende bien, ya **que solo ha aprendido con una parte de la muestra**, al tener que clasificar una muestra con otra etiqueta que aún no conoce lo clasificaría según lo ya aprendido. Por tanto, es posible que el algoritmo C clasifique mejor.

Al contrario que en el ejercicio anterior, aquí nos dicen que el **conjunto de datos no es iid**, por lo que ya sabemos que un **algoritmo inteligente no va a obtener un modelo correcto**.

5. Considere la cota para la probabilidad del conjunto de muestras de error D de la hipótesis solución g de un problema de aprendizaje, a partir de la desigualdad de Hoeffding, sobre una clase finita de hipótesis,

$$\mathbb{P}[|E_{out}(g) - E_{in}(g)| > \epsilon) < \delta(\epsilon, N, |\mathcal{H}|)$$

a) Dar una expresión explícita para $\delta(\epsilon, N, |\mathcal{H}|)$

$$\delta\left(\epsilon, N, |\mathcal{H}|\right) = 2|\mathcal{H}|e^{-2e^2N}$$

b) Si fijamos $\epsilon = 0,05$ y queremos que el valor de δ sea como máximo $0,03$ ¿cual será el valor más pequeño de N que verifique estas condiciones cuando $\mathcal{H} = 1$?.

Usamos la fórmula del algoritmo PAC que viene dada en función de ϵ, δ y $\mathcal{H}$ para calcular el número mínimo de muestras necesarias para esos valores.

$$N = \frac{1}{\epsilon}\left(ln(|\mathcal{H}|) + ln\left(\frac{1}{\delta}\right)\right)$$
$$\epsilon = 0.05, \quad \mathcal{H} = 1, \quad \delta = 0.03$$
$$N = \frac{1}{0.05}\left(ln(1) + ln\left(\frac{1}{0.03}\right)\right)$$

Input:

$$N = \frac{1}{0.05}\left(\log(1) + \log\left(\frac{1}{0.03}\right)\right)$$

Open code

$\log(x)$ is the natural logarithm

Result:

$$N = 70.1312$$

$$N = \frac{1}{0.05}\left(ln(1) + ln\left(\frac{1}{0.03}\right)\right) = 71$$

c) Repetir para $\mathcal{H} = 10$ y para $\mathcal{H} = 100$

H = 10 ----------

$$N = \frac{1}{\epsilon}\left(ln(|\mathcal{H}|) + ln\left(\frac{1}{\delta}\right)\right)$$
$$\epsilon = 0.05, \quad \mathcal{H} = 10, \quad \delta = 0.03$$
$$N = \frac{1}{0.05}\left(ln(10) + ln\left(\frac{1}{0.03}\right)\right)$$

Input:

$$N = \frac{1}{0.05}\left(\log(10) + \log\left(\frac{1}{0.03}\right)\right)$$

Open code

$\log(x)$ is the natural logarithm

Result:

$$N = 116.183$$

$$N = \frac{1}{0.05}\left(ln(10) + ln\left(\frac{1}{0.03}\right)\right) = 117$$

H = 100 ----------

$$N = \frac{1}{\epsilon}\left(ln(|\mathcal{H}|) + ln\left(\frac{1}{\delta}\right)\right)$$

$$\epsilon = 0.05, \quad \mathcal{H} = 100, \quad \delta = 0.03$$

$$N = \frac{1}{0.05}\left(ln(100) + ln\left(\frac{1}{0.03}\right)\right)$$

Input:

$$N = \frac{1}{0.05}\left(\log(100) + \log\left(\frac{1}{0.03}\right)\right)$$

Open code

$\log(x)$ is the natural logarithm

Result:

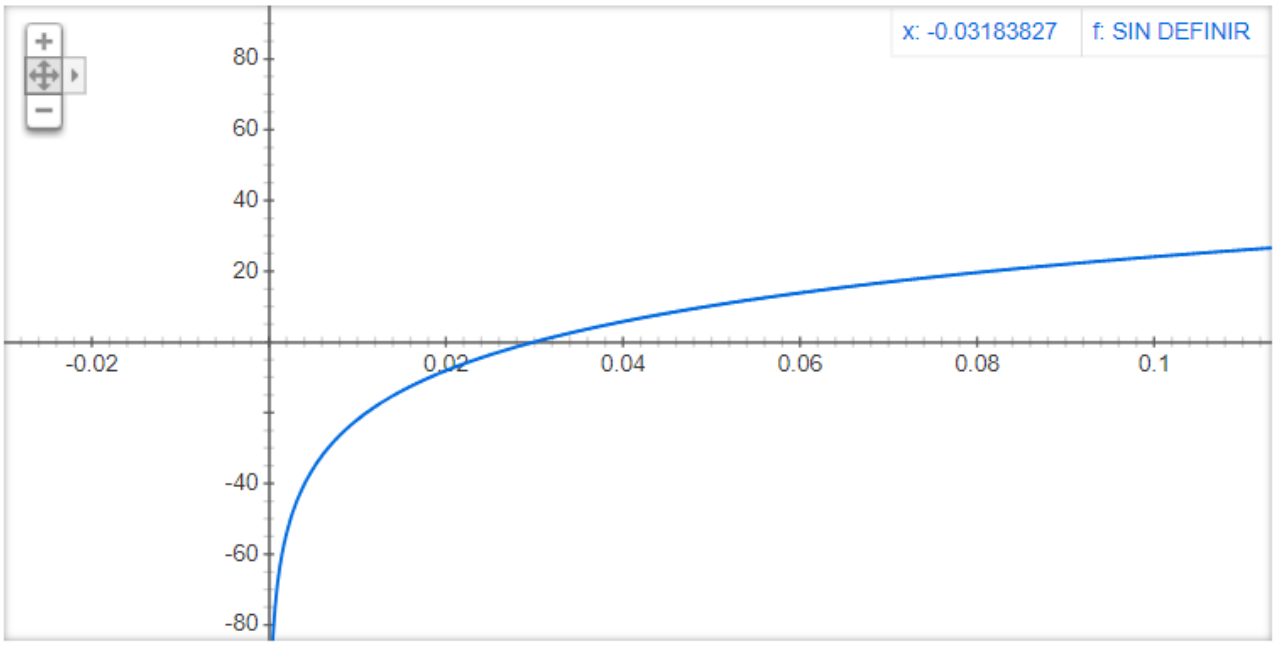
$$N = 162.235$$

$$N = \frac{1}{0.05}\left(ln(100) + ln\left(\frac{1}{0.03}\right)\right) = 163$$

¿Qué conclusiones obtiene?

Dado un δ fijo, el tamaño de la muestra necesaria para aprender con una tasa de error (ϵ) fijada crece linealmente con el logaritmo natural del tamaño del conjunto $\mathcal{H}$. Para comprobar esto con el ejemplo dado se puede observar en la siguiente gráfica cómo la función aumenta logarítmicamente en función de $|\mathcal{H}|$.

Gráfico de 1/0.05*(ln(x)+ln(1/0.03))



6. Considere la cota para la probabilidad del conjunto de muestras de error $\mathcal{D}$ de la hipótesis solución g de un problema de aprendizaje, a partir de la desigualdad de Hoeffding, sobre una clase finita de hipótesis,

$$\mathbb{P}[|E_{out}(g) - E_{in}(g)| > \epsilon) < \delta$$

a) ¿Cuál es el algoritmo de aprendizaje que se usa para elegir g?

Aquel que cumple con la regla **ERM** (Empirical Risk Minimization), es decir, **se escoge** aquel que halle con una función g **que minimice el error**.

b) Si elegimos g de forma aleatoria ¿seguiría verificando la desigualdad?

No, al escoger una función de forma aleatoria es **probable que verifique la igualdad**, depende del azar. Incluso escogiendo una g que ajuste bien el conjunto es probable que nunca lleguemos a una probabilidad menor que δ.

c) ¿Depende g del algoritmo usado?.

Sí, antes de conocer el conjunto de datos debemos fijar una g, después **será el algoritmo el encargado de** usar la muestra de entrenamiento para **buscar la función** g.

d) ¿Es una cota ajustada o una cota laxa?

Es una **cota laxa** ya que es posible que la diferencia entre errores sea menor que el error de generalización, y al no cumplirse esto la probabilidad no será menor que la cota δ.

Justificar las respuestas.

7. ¿Por qué la desigualdad de Hoeffding no es aplicable de forma directa cuando el número de hipótesis de $\mathcal{H}$ es mayor de 1? Justificar la respuesta.

A la hora de usar la desigualdad de Hoeffding debemos fijar nuestra h antes de conocer el conjunto de datos. Si tras usar la desigualdad no obtenemos un resultado aceptable podemos añadir más hipótesis hasta encontrar una solución.

Al tener varias $\mathcal{H}$ hemos ido comprobando que $\mathbb{P}[|E_{out}(h_m) - E_{in}(h_m)| > \epsilon)$ no es pequeño, si no que va decrecindo hasta que se cumple $\mathbb{P}[|E_{out}(g) - E_{in}(g)| > \epsilon)$ siendo ahora g la última hipótesis de $\mathcal{H}$.

8. Si queremos mostrar que k^* es un punto de ruptura para una clase de funciones $\mathcal{H}$ cuales de las siguientes afirmaciones nos servirían para ello:

a) Mostrar que existe un conjunto de k^* puntos $x_1, \ldots, x_{k^*}$ que $\mathcal{H}$ puede separar ("shatter").

No, debido a que en el caso de tener una $\mathcal{H}$ capaz de separar el conjunto $k*$, $k*$ no es un punto de ruptura.

> *Definition 2.3. If no data set of size k can be shattered by $\mathcal{H}$, then k is said > to be a break point for $\mathcal{H}$ - Learning from data*

b) Mostrar que $\mathcal{H}$ puede separar cualquier conjunto de k^* puntos.

No ya que si $\mathcal{H}$ puede separar cualquier conjunto de $k*$ puntos, $k*$ nunca será un punto de ruptura de esa $\mathcal{H}$.

c) Mostrar un conjunto de k^* puntos $x_1, \ldots, x_{k^*}$ que $\mathcal{H}$ no puede separar.

Un punto de ruptura es aquel que nos indica el tamaño de la muestra en el que nuestra $\mathcal{H}$ no es capaz de separar al conjunto. Que $\mathcal{H}$ no sea capaz de separar ese conjunto de puntos $k*$ no significa que no pueda separar otro $k*$ distinto, por lo que esta afirmación **no** nos vale.

d) Mostrar que $\mathcal{H}$ no puede separar ningún conjunto de k^* puntos.

Si encontramos una $\mathcal{H}$ tal que no sea capaz de separar a ningún conjunto de puntos, se cumple que $m_{\mathcal{H}}(k) < 2^{k^*}$, por tanto se cumple que $k*$ es un punto de ruptura.

e) Mostrar que $m_{\mathcal{H}}(k) = 2^{k^*}$.

No valdría puesto que el que $k*$ sea un punto de ruptura verifica $m_{\mathcal{H}}(k) < 2^{k^*}$.

9. Para un conjunto $\mathcal{H}$ con $d_{VC} = 10$, ¿que tamaño muestral se necesita (según la cota de generalización) para tener un 95 % de confianza (δ) de que el error de generalización (ϵ) sea como mucho 0.05?

$$\sqrt{\frac{8}{N} ln\left(\frac{4m_{\mathcal{H}}(2N)}{\delta}\right)} \leq \epsilon$$
$$\frac{8}{N} ln\left(\frac{4m_{\mathcal{H}}(2N)}{\delta}\right) \leq \epsilon^2$$
$$\frac{8}{\epsilon^2} ln\left(\frac{4m_{\mathcal{H}}(2N)}{\delta}\right) \leq N$$
$$\frac{8}{\epsilon^2} ln\left(\frac{4((2N)^{d_{VC}} + 1)}{\delta}\right) \leq N$$

En el último paso reemplazamos $m_{\mathcal{H}}(2N)$ por su límite superior polinómico basado en la dimensión VC, de esta forma obtenemos $((2N)^{d_{VC}} + 1)$. Una vez tenemos esto podemos sustituir para obtener el tamaño de la muestra.

$$\frac{8}{\epsilon^2}\ ln\left(\frac{4((2N)^{d_{VC}} + 1)}{\delta}\right) \leq N$$
$$\frac{8}{0.05^2}\ ln\left(\frac{4((2N)^{10} + 1)}{0.05}\right) \leq N$$
$$\frac{8}{0.025}\ ln\left(\frac{4((2N)^{10} + 1)}{0.05}\right) \leq N$$
$$\frac{8}{0.025}\ ln\left(\frac{4((2N)^{10} + 1)}{0.05}\right) \leq N$$
$$\frac{8}{0.025}\left(ln\left(4((2N)^{10} + 1)\right) - ln(0.05)\right) \leq N$$
$$\frac{8}{0.025}\left(ln(4(2N)^{10} + 4) - ln(0.05)\right) \leq N$$
$$\frac{8}{0.025}\left(ln(2^{12}N^{10} + 4) - ln(0.05)\right) \leq N$$
$$\frac{8\left(ln(2^{12}N^{10} + 4) - ln(0.05)\right)}{0.025} \leq N$$
$$320\left(ln(2^{12}N^{10} + 4) - ln(0.05)\right) \leq N$$

Como no somos capaces de despejar la N hacemos uso de un software de análisis matemático (*wolfram alpha*) para aproximar la solución.

Podemos ver que obtenemos un tamaño muestral de 37306.5 , dado que el tamaño de nuestra muestra debe ser un número entero tomamos como **solución** $N = 37307$.

10. Considere que le dan una nuestra de tamaño N de datos etiquetados {−1, +1} y le piden que encuentre la función que mejor ajuste dichos datos. Dado que desconoce la verdadera función f, discuta los pros y contras de aplicar los principios de inducción **ERM** y **SRM** para lograr el objetivo. Valore las consecuencias de aplicar cada uno de ellos.

En caso de usar **ERM**: La reducción del riesgo empírico consiste en reducir el error de clasificación de la muestra de entrenamiento esperando que minimice de igual forma en cualquier muestra que cumpla con la misma distribución uniforme.

Al buscar minimizar el error debemos tener en cuenta que en **caso de que los datos no sean *i.i.d.***(independientes e idénticamente distribuidos), **minimizar el error no garantiza** que encontremos el **modelo que mejor ajusta** a todos los datos, si no el modelo que ajusta ese conjunto con un error mínimo lo cual tiende a **sobreajustar** el conjunto, es decir, busca la función que mejor ajusta esa muestra, lo cual puede no servir para otra distinta.

En caso de usar **SRM**: En este caso buscamos minimizar el riesgo estructural, lo que nos permite **solucionar el problema del sobreajuste**. Con este principio se busca una estructura sólida que minimice el E_{out}, por lo que aunque tengamos una muestra pequeña, se hallará la mejor función del conjunto de hipótesis que divida los datos.