Cuestionario de teoría 1

José Carlos Martínez Velázquez

3 de Abril de 2016

1.Identificar, para cada una de las siguientes tareas, que tipo de aprendizaje es el adecuado (supervisado, no supervisado, por refuerzo) y los datos de aprendizaje que deberiamos usar. Si una tarea se ajusta a más de un tipo, explicar como y describir los datos para cada tipo.

Vamos a explicar los diferentes tipos de aprendizaje que tenemos:

- A. Supervisado: dado un conjunto de ejemplos etiquetados anteriormente, el aprendizaje supervisado devuelve una función que trata de clasificar ejemplos nuevos, es decir, que imita a la función objetivo. Este tipo de aprendizaje se usa para resolver el problema de clasificación.
- A. No supervisado: este tipo de aprendizaje no dispone de ejemplos etiquetados anteriormente, sino un conjunto de entradas al sistema. Para tener éxito, el sistema debe ser capaz de detectar patrones y aplicarlos. Es lo que se usa, por ejemplo, en data mining.
- A. Por refuerzo: En refuerzo, el sistema tampoco dispone inicialmente de ejemplos etiquetados. La forma de aprender es lo más parecido al sistema humano, debe ser capaz de aprender de sus errores. Cuando el sistema realiza una acción, se retroalimenta para no volver a realizar esta acción, por lo tanto es un aprendizaje a ensayo y error. El aprendizaje por refuerzo podría usarse para que una máquina aprendiese a jugar, por ejemplo, al ajedrez.

a) Categorizar un grupo de animales vertebrados en pajaros, mamíferos, reptiles, aves y anfibios.

Las entradas que debería tener este sistema serían:

- booleano tienePlumas: Clasificará entre pájaros y aves cuando sea true. En caso de ser false, se quedará con mamíferos, reptiles y anfibios.
- float ó int tamanio: Podríamos definir el tamaño de un vertebrado mediante dos sistemas: su altura exacta, por ejemplo, para lo que utilizaríamos un número en coma flotante o bien un sistema del tipo 0:muy pequeño, 1: pequeño, . . . , 5: muy grande.
- int[] color: Los colores se pueden definir de muchas maneras, pero nos aprovecharemos del sistema RGB para representarlos. Un vector de 3 enteros entre 0 y 255.

Este problema podría ajustarse a un esquema de aprendizaje supervisado, pues el ser humano es capaz de clasificar muy fácilmente a los vertebrados sin saber cómo lo hace, por lo que sería muy fácil conseguir ejemplos etiquetados.

Podría también ajustarse al aprendizaje no supervisado, pues por ejemplo un animal sin plumas, muy pequeño y de color verde, casi seguro que es un anfibio, así que probablemente un sistema de aprendizaje no supervisado fuese capaz de extraer patrones de un conjunto de entradas.

Este problema no puede ajustarse al aprendizaje por refuerzo, pues no hay nada que le diga que ha fallado cuando clasifica.

b) Clasificación automática de cartas por distrito postal

Al igual que un código postal se divide en partes, podemos dar al sistema un origen y destino codificados con números enteros.

El problema se ajusta a un esquema de aprendizaje supervisado, pues es capaz de extraer la probabilidad del destino dado un origen mediante ejemplos previos.

No puede ajustarse al aprendizaje no supervisado, pues no podría extraerse la probabilidad del destino sin ejemplos previos. Además es prácticamente imposible extraer patrones de un conjunto donde sólo se proporciona el origen.

No puede ajustarse ni de lejos a un problema de aprendizaje por refuerzo, pues además de que no hay nada que le diga que ha fallado, sería desastroso poner en manos de un aprendizaje por refuerzo este problema, pues la forma de clasificación sería prácticamente aleatoria.

c) Decidir si un determinado índice del mercado de valores subirá o bajará dentro de un periodo de tiempo fijado.

- float periodo_tiempo: saber si va a subir o bajar en este periodo (dias, meses,...)
- float valor_actual: valor actual del índice
- float media_valor: media del valor durante el último año.
- int periodo_tiempo_subida: cuanto tiempo lleva en subida desde la última bajada
- int periodo_tiempo_bajada: cuanto tiempo lleva en bajada desde la última subida
- booleano valor_estable: hay valores que son estables por definición

También influyen otros aspectos no codificables como la situación política, financiera, etc del país en cuestión. Suponiendo que hemos encontrado alguna manera de hacerlo, valoremos los tipos de aprendizaje:

Podría ajustarse a un aprendizaje supervisado, pero dado el caracter cambiante del entorno del mercado bursátil, algo que tuvo éxito en un momento determinado, podría no tenerlo en el momento, entonces, el porcentaje de acierto podría no ser excelente, ya que depende de los ejemplos.

Podría ajustarse también al aprendizaje no supervisado, pueden extraerse patrones de un conjunto de ejemplos y decidir en consecuencia.

También podríamos ajustar el problema a un aprendizaje por refuerzo, aunque no es lo mejor al principio, pues el modo de decidir es practicamente aleatorio. Cuando gana varias veces aprenderá cuáles son los factores propicios para ganar y cuáles para perder, independientemente de la situación temporal.

2. ¿Cuales de los siguientes problemas son más adecuados para una aproximación por aprendizaje y cuales más adecuados para una aproximación por diseño? Justificar la decisión

Debemos separar aprendizaje de diseño. Una aproximación por diseño pretende fijar unos parámetros para un problema específico. La aproximación por aprendizaje pretende aprender una hipótesis (función) que se usará para clasificar un conjunto de datos correctamente.

a) Determinar el ciclo óptimo para las luces de los semáforos en un cruce con mucho tráfico.

Es un problema algorítmico que puede resolverse mediante una aproximación por diseño. Podría medirse el flujo de tráfico que circula por cada calle que converge al cruce y determinar un grado de descongestión del cruce dando más tiempo de luz verde a la calle en la que el flujo sea mayor.

b) Determinar los ingresos medios de una persona a partir de sus datos de nivel de educación, edad, experiencia y estatus social.

Este problema es muy claro. Imaginemos que queremos determinar los ingresos medios de una persona sólo con dos de éstos parámetros, entonces podríamos incluso dibujarlo en una gráfica sobre ejes X e Y. Lo único que cambia con varios parámetros es que tendríamos (si se pudiera representar) una gráfica n-dimensional. Está claro que existe una función (que no conocemos) que nos dice los ingresos medios de una persona sabiendo esos parámetros. Si conocemos algunos ejemplos, seríamos capaz de buscar otra hipótesis que aproximara ésta función que no conocemos. Sería un problema abordable desde una aproximación por aprendizaje.

c) Determinar si se debe aplicar una campaña de vacunación contra una enfermedad.

En mi opinión este problema es abordable desde las dos aproximaciones. Mediante la aproximación de diseño, fijaríamos parámetros sobre a cuánta gente afecta la enfermedad, su gravedad, etc. y asignar en base a los parámetros una probabilidad de éxito de dicha campaña. Mediante la aproximación de aprendizaje, si tenemos ejemplos de otras campañas y cómo han resultado, podríamos definir una hipótesis que fuese capaz de separar los casos en que fueron productivas y los que no, en base a eso, podemos decidir si la campaña que nos planteamos aplicar lo será o no.

3. Construir un problema de aprendizaje desde datos para un problema de selección de fruta en una explotación agraria (ver transparencias de clase). Identificar y describir cada uno de sus elementos formales. Justificar las decisiones.

Este problema es una generalización del de los mangos de la página 13 de las transparencias de teoría. Para saber si una fruta debe ser seleccionada, lo más razonable sería tener en cuenta, al menos, los siguientes parámetros.

- Color: Por el color podemos saber datos acerca del grado de madurez y la especie a la que pertenece una determinada fruta. Podríamos representarlo mediante una escala, por ejemplo del 0 al 10, pues un array de enteros sería demasiado.
- Tamaño: El tamaño nos puede dar pistas acerca de si la fruta es deseable o no, tambien podemos saber datos sobre la variedad. Se puede representar con un float del peso exacto o con otra escala.
- Textura: La textura nos dará pistas sobre el estado del consumo. Una fruta demasiado dura no podría ser consumida pues estará verde, una fruta excesivamente blanda podría estar podrida. Será representado con una escala del 1 al 5.
- Temporada de recoleccion: La temporada de recolección nos dará todos los datos que nos faltan sobre la especie a la que pertenece una fruta. Será representado por un rango de meses, es decir, un vector de int, donde el primer elemento será `mes_desde` y el segundo `mes_hasta`. Si la temporada sólo es de un mes, ambos valores coincidirán.

Esto es el espacio de dominio `X`.

El conjunto de etiquetas será binario: 1 la fruta es seleccionada, 0 la fruta no es seleccionada.

Sabemos que existe una función objetivo que no conocemos, pero tendremos un conjunto de ejemplos para los que tenemos un valor para cada parámetro y si la fruta fue seleccionada o no.

El resultado debería ser una función $g(X)$ que fuese capaz de clasificar nuevos ejemplos correctamente.

4. Suponga un modelo PLA y un dato x(t) mal clasificado respecto de dicho modelo. Probar que la regla de adaptación de pesos del PLA es un movimiento en la dirección correcta para clasificar bien x(t).

Vamos a ver, en pseudocódigo la parte que compensa el error de un punto:

```
punto=(x,y,1)#En una columna
error=sumar(etiqueta_verdadera-(traspuesta(pesos)%*%punto))

if(signo(trasupesta(pesos)%*%punto)!=etiqueta_punto){
  umbral_anterior<-ultimo_componente_pesos
  pesos<-pesos+(etiqueta_verdadera*punto)
  ultimo_componente_pesos<-umbral_anterior+error
}
```

Como vemos, lo que se hace es actualizar el umbral (a partir del cual se considera un signo u otro). Fijémonos en el caso más fácil y los componentes de los pesos. Cuando los pesos son el vector $(0,0,0)$ y encontramos el primer punto mal clasificado $(x,y,1)$, la etiqueta verdadera sólo puede ser -1, pues al 0 se le considera positivo. Los pesos se actualizan a $(-x,-y,-1)$. Como sólo consideramos los signos de la función en el algoritmo perceptrón, si un punto está mal clasificado es lo mismo que decir que se le ha considerado cambiado de signo, entonces, si los pesos se convierten en el punto cambiado de signo, el resultado de la multiplicación $pesos^T punto$ va a ser negativo seguro. En este caso se ve muy claro.

Cuando los pesos son distintos de $(0,0,0)$, es porque ya se ha encontrado algún (o algunos) puntos mal clasificados, entonces, estaríamos ante los valores (a,b,c) del vector de pesos. Si encontramos un nuevo punto $(x',y',1')$ mal clasificado, entonces los pesos se actualizarán a $(a-(x'*etiqueta_verdadera), b-(y'*etiqueta_verdadera), c+error)$. Lo único extraño es que a c se le suma $\pm error$ ¿por qué?. Cuando los pesos están inicializados a 0, el error sólo puede ser ± 1 $(0-1=-1)$, sin embargo, cuando los pesos han tenido alguna actualización, el error sólo puede ser ± 2. La componente c de los pesos es el umbral, luego en un caso genérico, siempre actualizamos los pesos compensando el error en el umbral, de este modo, si un punto tenía cambiado el signo, al mover el hiperplano compensando el error, si multiplicamos de nuevo $pesos^T punto$, tendremos el signo correcto.

Para entender cómo afecta el umbral al hiperplano, hay que recordar la ecuación de la recta. Dicha ecuación es de la forma $ax+by+c=\pm 1$, resultado de la multiplicación $(a,b,c)\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$, justo la traspuesta de los pesos por el punto. Supongamos que la etiqueta verdadera del punto es 1, entonces hemos cometido un error de $1-(-1)=2$. Si el punto estuviera mal clasificado, al final del proceso de compensación, tendríamos $(a+((1)x), b+((1)y), c+2)\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$, entonces la nueva ecuación de la recta ha cambiado a $(a+x)x+(b+y)y+(c+2)=1$. El umbral afecta al punto de corte con el eje y, pues despejando tenemos que: $y=-\frac{(a+x)}{(b+y)}x-\frac{(c+2)+1}{(b+y)}$. Donde la pendiente de la recta es $-\frac{(a+x)}{(b+y)}$ y el punto de corte con el eje y es $-\frac{(c+2)+1}{(b+y)}$. Entonces vemos cómo el error cometido juega un papel importante en cómo la recta es movida hacia la dirección correcta para compensarlo.

Básicamente, si tenemos un punto mal etiquetado, quiere decir que $signo(w^T X) \neq signo(Y)$, entonces, tenemos que $w^T XY < 0$. Lo que buscamos es que $signo(w^T X) = signo(Y)$, para que $w^T XY > 0$. La compensación de los pesos y el error explicados anteriormente es lo que nos permite conseguir este objetivo.

Supongamos un ejemplo concreto. Tenemos los pesos a $(0,0,0)$ y un punto mal clasificado $(3,5,1)$, entonces la etiqueta verdadera del punto es -1. Hemos cometido un error de $-1-0=-1$. Entonces, los nuevos pesos serán $(0+(-1)*3, 0+(-1)*5, 0+(-1))=(-3,-5,-1)$. Si volvemos a multiplicar los pesos por el punto,

ahora sí está bien clasificado, pues $(-3,-5,-1)\begin{pmatrix}3\\5\\1\end{pmatrix} < 0 \Rightarrow etiqueta = -1$. El error del primer punto ha sido corregido.

Supongamos un nuevo punto $(4,8,1)$ cuya etiqueta verdadera es 1. Si multiplicamos $(-3,-5,-1)\begin{pmatrix}4\\8\\1\end{pmatrix} < 0 \Rightarrow etiqueta = -1$, lo cual es un error de $1-(-1)=2$, entonces hay que actualizar los pesos a $(-3+(1*4), -5+(1*8), -1+2) = (1,3,1)$. Si volvemos a multiplicar el punto por los nuevos pesos, nos damos cuenta cómo el error ha sido corregido: $(1,3,1)\begin{pmatrix}4\\8\\1\end{pmatrix} > 0 \Rightarrow etiqueta = 1$.

Esto no asegura que la buena clasificación del primer punto no se pierda, pues el algoritmo PLA corrige un punto pero puede dejar mal clasificados a muchos anteriores. Por eso, para dar por terminado (se ha conseguido una convergencia) el algoritmo, debe haber una iteración donde todos los puntos estén bien clasificados, es decir, no haya cambios.

5. Considere el enunciado del ejercicio 2 de la sección FACTIBILIDAD DEL APRENDIZAJE de la relación apoyo

Recordemos qué decia el ejercicio mencionado:

Supongamos que tenemos 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 $H = \{h_1, h_2\}$ 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. Vamos a estudiar cómo estos algoritmos se comportan fuera de la muestra desde un punto de vista determinístico y probabilístico. Suponga en el caso probabilístico que hay una distribución de probabilidad sobre X , y sea $P[f(x) = +1] = p$

a) Si p = 0,9 ¿Cual es la probabilidad de que S produzca una hipótesis mejor que C?

Lo primero es ver qué hipótesis elegiría S, que elegiría h_1, pues se ajusta más a los datos. De hecho, la probabilidad de que h_1 cometa un error sería $1-p = 0.1 = 10\%$. Dado que $P[f(x) = +1] = p = 0.9$, entonces sabemos que la probabilidad de que S acierte es también $p = 0.9$.

Si C es la hipótesis contraria a S, entonces sabemos que la probabilidad de que C acierte es $p = 0.1$ y de que falle $1-p = 0.9$.

Para que S sea mejor que C (o viceversa), a la probabilidad de que uno acierte hay que quitarle la probabilidad de que el otro acierte, pues si los dos acertaran con la misma probabilidad, la probabilidad de que uno fuese mejor que otro, sería 0. Entonces, la probabilidad de que S sea mejor C es de $p-(1-p) = 0.9-0.1 = 0.8 = 80\%$. Con una probabilidad del 80% S produciría una hipótesis mejor que C.

b) ¿Existe un valor de p para el cual es más probable que C produzca una hipótesis mejor que S?

Dado que S siempre elige la hipótesis que mejor se ajuste a los datos, no habrá un p donde haya una probabilidad mayor de que C sea mejor que S. Sólo hay un caso donde C podría producir una hipótesis igual de buena que S, las probabilidades de acertar deberían ser iguales. Este es el caso aleatorio, donde $P[f(x) = +1] = 0.5$. Por tanto, no existe un p para el cual C sea mejor que S, lo máximo que podría conseguirse es que C produjese una hipótesis tan buena como la de S.

6. La desigualdad de Hoeffding modificada nos da una forma de caracterizar el error de generalización con una cota probabilística $P[|E_{out}(g) - E_{in}(g)| > \epsilon] \leq 2Me^{-2N\epsilon^2}$ para cualquier $\epsilon > 0$. Si fijamos ϵ = 0,05 y queremos que la cota probabilística $2Me^{-2N^2\epsilon}$ sea como máximo 0.03 ¿cual será el valor más pequeño de N que verifique estas condiciones si M = 1?. Repetir para M = 10 y para M = 100

Lo primero que haremos será despejar N para aplicarlo de forma automática a todos los tamaños de M.

Tenemos un límite para la cota probabilística 0.03, que podría ser cualquier otro, a esto lo llamaremos δ. Entonces, tenemos que:

$$2Me^{-2N\epsilon^2} \leq \delta$$

Tomaremos logaritmo neperiano en ambos lados para bajar el exponente:

$$\ln(2Me^{-2N\epsilon^2}) \leq \ln(\delta) \Rightarrow \ln(2M) + \ln(e^{-2N\epsilon^2}) \leq \ln(\delta) \Rightarrow -2N\epsilon^2 \ln(e) \leq \ln(\delta) - \ln(2M) \Rightarrow$$

$$\Rightarrow -2N\epsilon^2 \leq \ln(\delta) - \ln(2M)$$

Ahora basta con despejar N^2 de la multiplicación del miembro de la izquierda

$$N \geq \frac{\ln(\delta) - \ln(2M)}{-2\epsilon^2}$$

Ya tenemos N despejado, ahora vamos a aplicarlo para cada caso.

- M=1

$$N \geq \frac{\ln(0.03) - \ln(2)}{-2 * 0.05^2} = 840$$

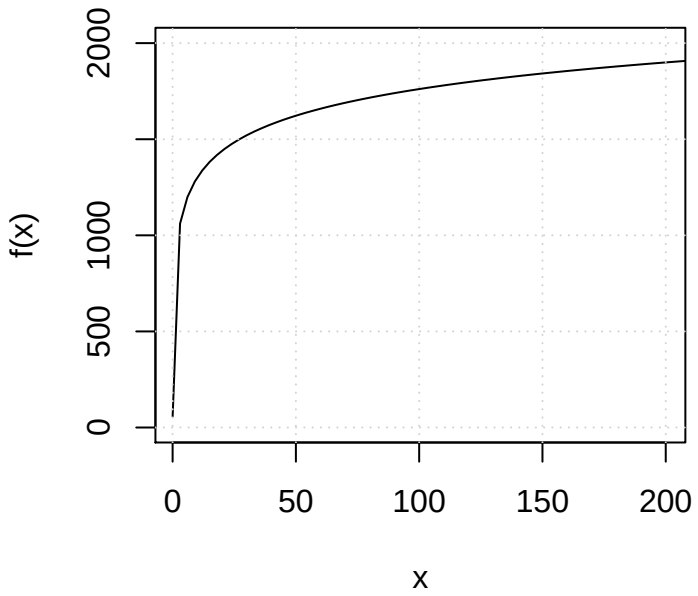
- M=10

$$N \geq \frac{\ln(0.03) - \ln(2 * 10)}{-2 * 0.05^2} = 1301$$

- M=100

$$N \geq \frac{\ln(0.03) - \ln(2 * 100)}{-2 * 0.05^2} = 1761$$

Vemos cómo al aumentar M también aumenta el tamaño de la muestra, pero, como muestra la función $f(M) = \frac{\ln(0.03) - \ln(2M)}{-2*0.05}$, lógicamente, a mayor complejidad de la función objetivo, mayor tamaño de la muestra, es decir, cuando M tiende a infinito, el tamaño de la muestra también:



7.Consideremos el modelo de aprendizaje "M-intervalos" donde $h : R \to -1, +1$, y $h(x) = +1$ si el punto está dentro de cualquiera de m intervalos arbitrariamente elegidos y -1 en otro caso. ¿Cual es el más pequeño punto de ruptura para este conjunto de hipótesis?

Lo primero de todo es saber qué es un punto de ruptura. Sea $mH(k)$ las dicotomías (todas las formas posibles de separar k puntos) que puede implementar H, y sea k el número de puntos que queremos separar, se supone que hay 2^k formas de separar los puntos. Si por alguna razón $mH(k) < 2^k$, entonces k es un punto de ruptura. Dicho de otro modo, si H no es capaz de separar ningún conjunto de k puntos de las 2^k formas posibles de hacerlo, entonces k es un punto de ruptura. Por supuesto, el mínimo punto de ruptura es el primer k que, en orden ascendente ($k = 1, 2, 3, ...$), le ocurre esto.

Dado un número de puntos N, podemos suponer que elegir intervalos es asignar signos a los puntos, por lo que el número de dicotomías es el número de combinaciones posibles de 1 punto positivo y los demas negativos (y su contrario), dos puntos negativos y todos positivos (y su contrario)... y además todos positivos o todos negativos, lo que se calcula con la expresión $\binom{N+1}{2}$. La forma más restrictiva de asignar intervalos y puntos es, por ejemplo, si $N = 5$: $+ - + - +$, es decir, la alternancia. Entonces, podemos diferenciar dos casos: Si $N \leq M$, la separación siempre se puede hacer, ya que los intervalos sobrantes quedan a izquierda o derecha de los puntos. Si $N > M$, es cuando tenemos el problema. Si tenemos N puntos y M intervalos tal que N es mayor que M, entonces, el más pequeño punto de ruptura siempre es $k = \frac{N}{2}$ (truncado) si N es impar y $k = \frac{N}{2} - 1$ si N es par. Supongamos el caso anterior, $N = 5$, entonces, la configuración más restrictiva es $+ - + - +$ donde $N > M$. Si $M = truncado(\frac{N}{2}) = 2$, esta configuración no se puede alcanzar nunca ya que necesitaríamos, al menos 3 intervalos. Así que vemos que el punto de ruptura, en este caso, siempre depende de N: $k = \frac{N}{2} - 1$. Si $N > M$ y $M = \frac{N}{2}$, entonces siempre se puede hacer incluso la partición más restrictiva.

En resumen, el punto más pequeño de ruptura depende de N. Si N es par, $k = \frac{N}{2} - 1$, si N es impar, $k = truncado(\frac{N}{2})$.

8. Suponga un conjunto de k^* puntos $x_1, x_2, ..., x_{k^*}$ sobre los cuales la clase H implementa $< 2^{k^*}$ dicotomías. ¿Cuales de las siguientes afirmaciones son correctas?

a) k^* es un punto de ruptura

NO. No podemos saberlo, pues si hay otro conjunto de k^* puntos para los que H implementa las 2^{k^*} dicotomías, entonces k^* no es un punto de ruptura.

b) k^* no es un punto de ruptura

NO. No podemos saberlo, pues si para todo conjunto de k^* puntos H siempre implementa menos de 2^{k^*} dicotomías, entonces k^* es un punto de ruptura.

c) todos los puntos de ruptura son estrictamente mayores que k^*

NO. No podemos saberlo ya que no podemos saber si k^* es un punto de ruptura.

d) todos los puntos de ruptura son menores o iguales a k^*

NO. Al igual que en c), no podemos saberlo, pues para ello deberíamos poder asegurar que k^* es un punto de ruptura.

e) no conocemos nada acerca del punto de ruptura

SÍ. Exacto, sólo con un conjunto no podemos asegurar nada acerca de si k^* es un punto de ruptura o no.

9. Para todo conjunto de k^* puntos, H implementa $< 2^{k^*}$ dicotomías. ¿Cuales de las siguientes afirmaciones son correctas?

a) k^* es un punto de ruptura

SÍ. Podemos asegurarlo, pues sabemos que H implementa menos de 2^{k^*} dicotomías para todos y cada uno de los conjuntos de k^* puntos.

b) k^* no es un punto de ruptura

NO. Lo descartamos automáticamente. Sabemos con total seguridad que k^* es un punto de ruptura.

c) todos los $k \geq k^*$ son puntos de ruptura

SÍ. Si H no llega a implementar 2^{k^*} dicotomías, entonces tampoco llegará a implementar $2^{k^*+1}, 2^{k^*+2}, ..., 2^{k^*+n}$

d) todos los $k < k^*$ son puntos de ruptura

NO. No podemos asegurarlo. si k^* es el menor punto de ruptura (el primero), entonces $k^* - 1$ no lo es y H podrá implementar 2^{k^*-1} dicotomías.

e) no conocemos nada acerca del punto de ruptura

NO. Conocemos que k^* es un punto de ruptura y que, al menos todos los $k^* + n$ también serán puntos de ruptura.

10. Si queremos mostrar que k^* es un punto de ruptura cuales de las siguientes afirmaciones nos servirían para ello:

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

NO. Si H puede separar al menos un conjunto de k^* puntos, estaríamos demostrando todo lo contrario, que k^* NO es un punto de ruptura.

b) Mostrar que H puede separar cualquier conjunto de k^* puntos.

NO. Si H puede separar cualquier conjunto de k^* puntos estaríamos demostrando que k^* NO es un punto de ruptura.

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

NO. Como hemos visto en el ejercicio 8, si sabemos que H no es capaz de separar un conjunto de k^* puntos, no quiere decir que no pudiera existir otro conjunto de k^* puntos que H si pudiera separar, en ese caso estaríamos demostrando que k^* NO es un punto de ruptura.

d) Mostrar que H no puede separar ningún conjunto de k^* puntos

SÍ. Como hemos visto en el ejercicio 9, si sabemos que H no es capaz de separar ningún conjunto de k^* puntos, podemos asegurar con toda certeza que k^* es un punto de ruptura.

e) Mostrar que $mH(k) = 2^{k^*}$

NO. Aquí estamos diciendo que la función de crecimiento de H para k puntos es igual a 2^{k^*}, por lo general $k \neq k^*$, por este argumento no podemos asegurar que k^* sea un punto de ruptura. Pero aunque $k = k^*$, lo que estaríamos demostrando en este caso es que k^* seguro, no es un punto de ruptura, pues el número de dicotomías que sería capaz de implementar H sería 2^{k^*}.

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

Si la dimension de Varnip & Chervonenkis es 10, entonces sabemos que $mH(2N) = (2N)^{d_{VC}} + 1$, entonces $mH(2N) = (2N)^{10} + 1$. Si resolvemos la ecuación,

$$\sqrt{\frac{8}{N} \log \frac{4((2N)^{10} + 1)}{0.05}} = 0.05$$

, obtenemos que $N = 452957$, es decir, el tamaño de la muestra debería ser 452957 como mínimo.

(Para resolver la ecuación he utilizado la herramienta online Wolfram y he representado las funciones de izquierda y derecha del signo =, posteriormente he estimado el valor de N en el punto donde se cruzan ambas funciones. La variable N no se puede despejar ya que, aparece en distintos niveles de la ecuación)

Según la desigualdad de Hoeffding tenemos que:

$$P[|E_{out}(h) - E_{in}(h)| \leq \epsilon] \geq 1 - 4mH(2N)e^{-\frac{N\epsilon^2}{8}}$$

Entonces, podemos decir que:

$$P[|E_{out}(h) - E_{in}(h)| \leq 0.05] \geq 1 - 4((2 * 452957)^{10} + 1)e^{-\frac{452957 * 0.05^2}{8}}$$

Fijémonos que, como el término $e^{-\frac{452957 * 0.05^2}{8}}$ es muy cercano a 0, la probabilidad tenderá a $1 - 0 = 0$, por lo que, con total seguridad, con un tamaño de muestra mayor o igual que 452957, la diferencia entre el error fuera de la muestra y el error dentro de la muestra será como mucho 0.05.

12. Consideremos un escenario de aprendizaje simple. Supongamos que la dimensión de entrada es uno. Supongamos que la variable de entrada x está uniformemente distribuida en el intervalo $[-1, 1]$ y el conjunto de datos consiste en 2 puntos $\{x_1, x_2\}$ y que la función objetivo es $f(x) = x^2$. Por tanto es conjunto de datos completo es $D = \{(x_1, x_1^2), (x_2, x_2^2)\}$. El algoritmo de aprendizaje devuelve la línea que ajusta estos dos puntos como g (i.e. H consiste en funciones de la forma $h(x) = ax + b$).

a) Dar una expresión analítica para la función promedio $\bar{g}(x)$.

Vamos a obtener primero la expresión de la recta que pasa por los puntos de D. Partimos de la teoría: dados los dos puntos $(x_1, x_1^2), (x_2, x_2^2)$, podemos calcular la pendiente de la recta y el punto de corte con el eje Y para obtener una ecuación de la recta de la forma $y = mx + b$:

- Pendiente:
$$m = \frac{\Delta x^2}{\Delta x} = \frac{x_2^2 - x_1^2}{x_2 - x_1}$$
- Punto de corte con Y:
$$b - x_1^2 = m(x - x_1) \Rightarrow b = \frac{x_2^2 - x_1^2}{x_2 - x_1}(x - x_1) + x_1^2 \Rightarrow$$
$$\Rightarrow b = \frac{x_2^2 - x_1^2}{x_2 - x_1}x - \frac{x_2^2 - x_1^2}{x_2 - x_1}x_1 + x_1^2 \Rightarrow$$
(El término $\frac{x_2^2 - x_1^2}{x_2 - x_1}x = 0$ porque x en el punto de corte es 0.)
$$\Rightarrow b = -\frac{x_2^2 - x_1^2}{x_2 - x_1}x_1 + x_1^2 \Rightarrow$$
Si aplicamos "suma por diferencia igual a diferencia de cuadrados", nos queda:
$$\Rightarrow b = -\frac{(x_2 + x_1)(x_2 - x_1)}{(x_2 - x_1)}x_1 + x_1^2 \Rightarrow b = -(x_2 + x_1)x_1 + x_1^2 \Rightarrow$$
$$\Rightarrow b = -x_1x_2 - x_1^2 + x_1^2 \Rightarrow b = -x_1x_2$$

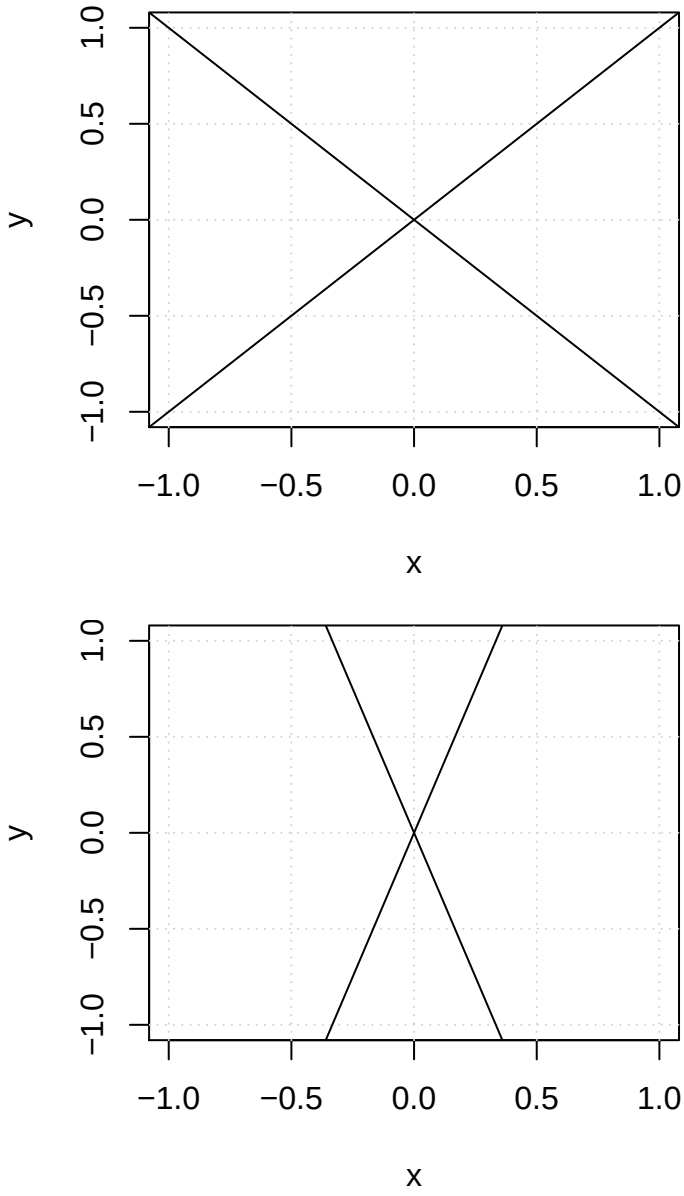
Entonces, tenemos que la función que aproxima a $f(x)$ es:

$$g^{(S)}(x) = \frac{x_2^2 - x_1^2}{x_2 - x_1}x - x_1x_2$$

Vamos a calcular la función promedio $\bar{g}(x)$, cuya expresión es aproximadamente igual a $\frac{1}{K}g^1(x) + ... + g^K(x)$, lo cual no es más que la media de todos los valores de g para todos los puntos. Concretamente, lo haremos para el rango de x [-1,1].

La estrategia a seguir para calcular la funcion $\bar{g}(x)$ será buscar el promedio de la pendiente y el promedio del punto de corte, así tendremos una función promedio $\bar{g}(x) = \bar{m}x + \bar{b}$

- Promedio de la pendiente: encontrar el promedio de la pendiente es de sentido común. Si eligiéramos dos puntos cualesquiera en el intervalo $X = [-1, 1]$, siempre encontraríamos la recta contraría, que la contrarrestaría en media. Si, por extensión, pudiéramos elegir los infinitos puntos para dicho intervalo, siempre encontraríamos la recta de pendiente contraria que la contrarrestaría, por lo que la media de la pendiente siempre es 0. Ejemplos de lo que digo son:



- Promedio del punto de corte: encontrar el promedio de la pendiente es un poco más complicado, pero procederemos igual que en el punto anterior. Supongamos que podemos elegir las infinitas rectas que pasan por el intervalo $X = [-1, 1]$. ¿Cuáles serían los puntos de corte con el eje Y máximo y mínimo?. Para ello, consideraremos cuatro casos extremos posibles. Los dos primeros son las rectas que pasan por los puntos (-1,1),(1,1) y (1,1),(-1,1) (son la misma). Estos dos casos van a decirnos cuál es el máximo punto de corte con el eje Y. Los dos casos que faltan son los que nos darán el mínimo valor de punto de corte con el eje Y, para ello recurrimos a la recta tangente a los puntos $x = -1$ y $x = 1$.

Con los dos primeros casos, tenemos que el máximo punto de corte es $Y = 1 = 0x + b \Rightarrow b = 1$. Con los dos casos de la derivada, sabemos que $f'(x) = 2x$, entonces, La recta tangente a un punto de la curva será $y = mx + b$ tal que cumple los siguientes requisitos: su pendiente (m) es la derivada de la función en el punto y, por supuesto, debe pasar por el punto $(a, f(a))$, entonces, la recta que buscamos es $y = f'(a)x + b$, de donde se deduce que $b = f(a) - f'(a)x$. En nuestro caso concreto, el punto de corte con el eje Y de la recta tangente a un punto es, $b = x^2 - 2x^2$.

Apliquemos esto a los dos puntos extremos de nuestro intervalo $X = [-1, 1]$, tenemos que:

- Para -1: $b = (-1)^2 - 2(-1)^2 = -1$
- Para 1: $b = (1)^2 - 2(1)^2 = -1$

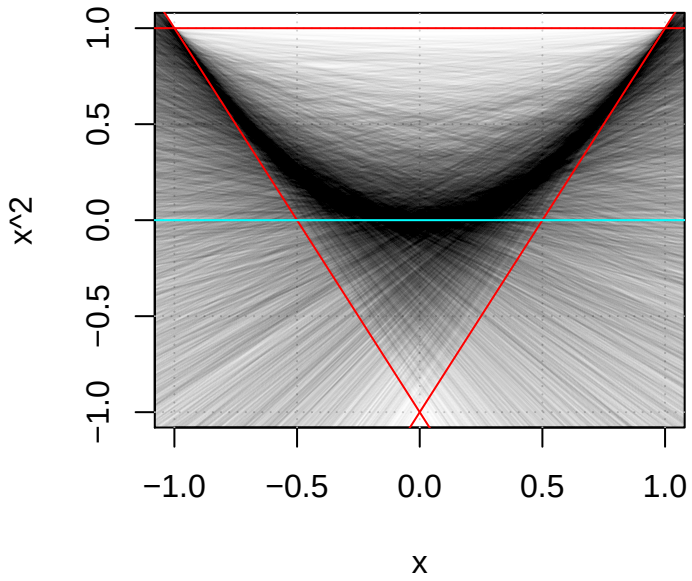
Teóricamente, sabemos que el máximo punto de corte con Y es 1 y que el mínimo punto de corte con Y es -1, por lo que el punto de corte promedio sería 0.

Ya tenemos, por tanto, la pendiente promedio y el punto de corte promedio, por lo que, $\bar{g}(x) = \bar{m}x + \bar{b} = 0x + 0 = 0$.

Apoyemos éste análisis con un razonamiento empírico. Vamos a elegir 10000 representantes de las infinitas rectas aleatorias que pasen por puntos del intervalo. En rojo, dibujaremos los casos extremos y, en cyan dibujaremos la función promedio $\bar{g}(x)$.

```
simula_recta<-function(intervalo){
  x=runif(2,-1,1)
  y=x^2
  #plot(x,y) Para asegurarse de que la recrta pasa por los puntos generados
  a=(y[2]-y[1])/(x[2]-x[1])#Calculo la pendiente
  b=(y[1])-((a)*(x[1]))#Aplico y=ax+b a un punto cualquiera
  c(x,y,a,b)
}

curve(x^2,-1,1,ylim=c(-1,1))
grid()
nRectas=10000
for(i in 1:nRectas){
  r<-simula_recta()
  abline(r[6],r[5],col=rgb(0,0,0,0.01))
}
abline(1,0,col="red")
abline(-1,-2,col="red")
abline(-1,2,col="red")
abline(0,0,col="cyan")
```



La expresión analítica, por tanto, de la función promedio es: $\bar{g}(x) = 0$.

b) Calcular analiticamente los valores de E_{out} , *bias*, y *var*.

Comencemos por la varianza. La teoría nos dice que: $var(x) = \mathbb{E}_S[(g^{(S)}(x) - \bar{g}(x))^2]$, entonces sustituyendo lo que tenemos nos sale que:

$$var(x) = \mathbb{E}_S[(\frac{x_2^2 - x_1^2}{x_2 - x_1}x - x_1x_2 - 0)^2] = \mathbb{E}_S[(\frac{x_2^2 - x_1^2}{x_2 - x_1}x - x_1x_2)^2]$$

Dado que la esperanza es con respecto a S, ($\mathbb{E}_S$), lo que queda dentro es una constante y se puede sacar, por lo que nos queda que:

$$var(x) = (\frac{x_2^2 - x_1^2}{x_2 - x_1}x - x_1x_2)^2$$

Continuemos calculando el sesgo ($bias(x)$). La teoría nos dice que $bias(x) = [(\bar{g}(x) - f(x))^2]$: Sustituyendo lo que tenemos, nos queda que:

$$bias(x) = [(0 - f(x))^2] = f(x)^2 = (x^2)^2 = x^4$$

Por último, calcularemos E_{out}. La teoría nos dice que $E_{out}(x) = \mathbb{E}_S[E_{out}^S(x)]$. $E_{out}^S(x) = (g^{(S)}(x) - f(x))^2$, entonces, en definitiva, lo que tenemos es:

$$E_{out}(x) = \mathbb{E}_S[(g^{(S)}(x) - f(x))^2] = \mathbb{E}_S[(g^{(S)}(x) - f(x))^2] = (\frac{x_2^2 - x_1^2}{x_2 - x_1}x - x_1x_2 - x^2)^2 = ((x_2 + x_1)x - x_1x_2 - x^2)^2$$