Universidad Nacional Autónoma de México
Facultad de Ciencias

Aprendizaje de Máquina

Profesor: Christian Gabriel Miranda Ruíz

Editor: Miguel Ángel Parra Ramírez

Agosto 2022
Semestre 2023-1

Índice

1. Teoría del aprendizaje

Breve descripción: Teoría de Aprendizaje Computacional

Teoría del Aprendizaje Computacional (Computational Learning Theory o *CoLT*), es un campo de estudio enfocado en el uso de métodos matemáticos aplicados a sistemas de aprendizaje. Específicamente, busca usar las herramientas teóricas de ciencias de la computación para cuantificar problemas de aprendizaje. Esto incluye la caracterización de la dificultad de aprendizaje de tareas en específico.

Esta se puede ver también como una extensión de *Teoría de aprendizaje estadístico* o SLT (por sus siglas en inglés) que usa métodos formales para cuantificar problemas de aprendizaje.

Teoría de aprendizaje Computacional (CoLT): Estudio formal de tareas de aprendizaje.
Teoría de aprendizaje Estadístico (SLT): Estudio formal de algoritmos de aprendizaje.

La división entre tareas de aprendizaje vs algoritmos de aprendizaje es arbitraria, y en la práctica, existe mucha intersección entre ambos campos.

> "Se puede extender la teoría del aprendizaje estadístico tomando en cuenta la complejidad computacional del aprendiz. Este camo es llamado Teoría del aprendizaje Computacional (CoLT)".
>
> página 210, Machine Learning: A Probabilistic Perspective, 2012.

Típicamente el foco de CoLT son tareas de aprendizaje supervisado, análisis formales de problemas de algoritmos reales es un reto grande. Es común reducir la complejidad del análisis enfocándose en tareas de clasificación binaria y en simples sistemas basados en reglas binarias. Las aplicaciones prácticas en teoremas puede ser limitado y puede llegara ser un reto interpretar problemas reales y algoritmos.

> " La principal pregunta sin responder es esta: ¿Cómo estamos seguros de que nuestro algoritmo de aprendizaje ha producido una hipótesis que pronostiquen el valor correcto para valores no antes vistos?"
>
> página 713, Artificial Intelligence: A Modern Approach, 3ed, 2009.

Las preguntas exploradas en CoLT pueden incluir:

- ¿Cómo sabemos si un modelo tiene una buena aproximación para la función objetivo?
- ¿Qué hipótesis de espacio se deben usar?
- ¿Cómo sabemos si se tienen una buena solución global o local?
- ¿Cómo evitamos el sobreajuste?
- ¿Cuántos ejemplos de datos son necesitados?

Puede ser muy útil conocer CoLT para aprender bien aprendizaje de máquina. Existen varios subcampos de estudio en CoLT, los más discutidos son:

- **Aprendizaje PAC**: Es la teoría de los problemas de aprendizaje del máquina.
- **Dimensión VC**: Es la teoría de los algoritmos de aprendizaje de máquina.

Breve descripción: Aprendizaje PAC (Teoría de problemas de aprendizaje)

Aprendizaje "probable y aproximadamente correcto"(PAC, por sus siglas en inglés), se refiere al marco teórico de referencia desarrollado por Leslie Valiant. Este enfoque busca cuantificar la dificultad de una tarea y puede ser considerada el primer sub-campo de la teoría de aprendizaje computacional.

Consideremos que en aprendizaje supervisado tratamos de aproximar una función subyacente desconocida de insumos a salidas. No se conoce cómo es la función que mapea, pero se sospecha que existe, y se tienen ejemplos de datos producidos por la función.

Lo que más se preocupa en el aprendizaje PAC es cuánto esfuerzo computacional es requerido para validar una hipótesis (ajustar un modelo) que se parezca lo más posible a la función objetivo.

¿Qué es una hipótesis en Aprendizaje de Máquina?
Sabemos que en aprendizaje de máquina, específicamente en aprendizaje supervisado, se puede describir como el deseo de usar datos disponibles para aprender sobre una función que mejor mapeé los insumos hacia las salidas. Este problema se llama aproximación de función, donde se aproxima a una función objetivo desconocida (que se asume que existe) y se refiere a la que mejor mapee insumos vs salidas de todas las posibles observaciones del problema en cuestión.

Un ejemplo de un modelo que aproxima la función objetivo y hace el mapeo de insumos hacia salidas es llamada una **hipótesis** en aprendizaje de máquina.

Escoger apropiadamente el algoritmo (por ejemplo: redes neuronales) y la configuración de este (topología de red e hiperparámetros) define el conjunto posible de hipótesis que el modelo pueda representar.

El término aprendizaje para un algoritmo de aprendizaje de máquina involucra navegar en todo el espacio de hipótesis hasta encontrar la mejor o la suficiente mejor hipótesis que mejor aproxime la función objetivo.

Una notación común es usar la h - minúscula, que representa una hipótesis dada y la $\mathscr{H}$ - mayúscula representa el espacio de hipótesis que se está buscando.

- **h** (hipótesis): Una sola hipótesis, una instancia o un candidato modelo para ser evaluado.
- $\mathscr{H}$ (conjunto de hipótesis): Un espacio de posibles hipótesis de modelos donde se puede encontrar el mejor modelo, bajo las restricciones del problema, la selección del modelo y la selección de la configuración de este.

La idea de una mala hipótesis recae en las predicciones que hace sobre datos nuevos basado en su error generalizado. Una hipótesis que predice correctamente en su mayoría, tiene un error generalizado pequeño, es probablemente una buena aproximación para la función objetivo.

El lenguaje probabilístico le da al teorema su nombre "Probable aproximadamente correcto". Esto es, una hipótesis busca "aproximar" una función objetivo que es "probablemente" buena si tiene un error generalizado bajo. Precisamente, un algoritmo de aprendizaje PAC hace referencia a un algoritmo que su resultado es PAC.

Usando métodos formales, un error generalizado mínimo puede ser especificado para una tarea de aprendizaje supervisado. El teorema puede ser utilizado para estimar el número esperado de muestras del problema en el dominio que se requerirían para determinar cuando una hipótesis es PAC o no. Esto es que, provee una manera de estimar el número de muestras requeridas para encontrar una hipótesis PAC.

> "La meta del marco de referencia PAC es entender qué tan grande tiene que ser un conjunto de datos para dar una buena generalización. También provee límites para el costo computacional del aprendizaje."
>
> Página 344, Pattern Recognition and Machine Learning, 2006.

Adicionalmente, un espacio de hipótesis (algoritmo de aprendizaje de máquina) es eficiente bajo el marco de referencia PAC si un algoritmo puede encontrar una hipótesis PAC (modelo ajustado) en tiempo polinomial.

> "Un espacio de hipótesis se dice que es eficientemente PAC-aprendizable si existe un algoritmo en tiempo polinomial que pueda identificar una función PAC".

Breve descripción: Dimension VC (Teoría de algoritmos de aprendizaje)
La teoría de Vapnik - Chervonenkis, o la teoría VC, se refiere al marco de referencia desarrollado por Vladimir Vapnik y Alexey Chervonenkis. Busca cuantificar la capacidad de un algoritmo de aprendizaje y puede ser considerada la primera subclase de la teoría de aprendizaje estadístico. Esta comprende varios elementos donde destaca la dimensión VC.

La dimensión VC cuantifica la complejidad de un espacio de hipótesis, es decir, los modelos que se puedan ajustar dada una representación y algoritmo de aprendizaje.

Una manera de considerar la complejidad de un espacio de hipótesis (espacio de modelos que puede ser ajustados) es basado en el número de distintas hipótesis que contiene y cómo el espacio puede ser navegado. La dimensión VC es una aproximación inteligente que mide el número de ejemplos desde el problema objetivo y puede ser discriminado por el espacio de hipótesis.

> " La dimensión VC mide la complejidad de espacio de hipótesis por el número de distintas instancias X que pueden ser completamente discriminadas usando $\mathscr{H}$"
>
> página 214, Machine Learning, 1997

Estima la capacidad de un algoritmo de clasificación de aprendizaje de máquina para un conjunto de datos específico (número y ejemplos de dimensiones).

Formalmente, la dimensión VC es el número más grande de ejemplos del conjunto de entrenamiento que el espacio de hipótesis que el algoritmo pueda "desmenuzar".

La dimensión VC de un algoritmo de aprendizaje de máquina es el número más grande de puntos de datos en un conjunto de datos que una configuración específica del algoritmo (hiperparámetros) o cualquier ajuste de modelo específico que pueda ser "desmenuzado.".

La dimensión VC es usada como parte del marco de referencia del aprendizaje PAC.

1.1. Modelos de aprendizaje

Para estudiar matemáticamente al aprendizaje de máquina, necesitamos definir formalmente el problema de aprendizaje. Esta precisa definición es llamada **modelo de aprendizaje**. Un modelo de aprendizaje debe de ser suficientemente enriquecido para capturar los aspectos más importantes de los problemas reales de aprendizaje, pero suficientemente simple para estudiar el problema matemáticamente. Como en cualquier modelo matemático, simplificar las suposiciones es inevitable.

Un modelo de aprendizaje debe responder varias preguntas:

- ¿Qué se está aprendiendo?
- ¿Cómo se están generando los datos?, ¿De dónde vienen?
- ¿Cómo se le están presentando los datos al que está aprendiendo o al "aprendido"? Por ejemplo, el aprendido ve todos los datos de una vez o un ejemplo por vez.
- ¿Cuál es el objetivo del aprendizaje en este modelo? En otras palabras, ¿Cómo se define qué significa para un algoritmo de aprendizaje ser exitoso?

Antes de empezar con el primer modelo de aprendizaje, se necesitan algunas definiciones básicas. Un ***ejemplo*** (a veces llamado ***instancia***), es el objeto que está siendo clasificado. En OCR las imágenes son ejemplos.

Usualmente, un ejemplo es descrito por un conjunto de ***atributos***, también conocidos como ***características***, ***variables*** o ***dimensiones***. Por ejemplo, un paciente puede ser descrito por sus atributos tales como género, edad, peso, presión sanguínea, temperatura corporal, etc.

La ***etiqueta*** o ***clase*** es la categoría que se está tratando de predecir. En OCR, las etiquetas son las posibles letras o dígitos que están siendo representados. Durante el entrenamiento, el algoritmo de aprendizaje es reemplazado con *ejemplos de etiquetas*, mientras en la prueba, sólo ejemplos *sin etiquetas* se proveen.

Para simplificar las cosas, asumiremos que sólo dos etiquetas son posibles, es decir que pueden ser o 0 o 1. Supondremos existe una mapeo de ejemplos a etiquetas, a este mapeo se le llama ***concepto***. Entonces un concepto, es una función de la forma $c: X \to \{0,1\}$ donde X es el espacio de todos los posibles ejemplos llamado el ***domino*** o el ***espacio de instancias***.

Una colección de conceptos es llamada una ***clase de conceptos***. Frecuentemente se asumirá que los ejemplos han sido etiquetados por un concepto desconocido de una clase de conceptos conocidos.

1.1.1. El modelo consistente

El primer modelo de aprendizaje, llamado el modelo consistente, es en realidad no realista, pero es intuitivo y es un muy buen lugar para empezar. Muchas ideas que surgirán serán de mucha ayuda después. El modelo captura la idea intuitiva de que la regla de predicción es derivada de un conjunto de ejemplos que debe de ser consistente con las etiquetas observadas.

Un concepto de clase $\mathscr{C}$ se dice que es ***aprendible*** con el modelo consistente si existe un algoritmo A el cual, dado un conjunto de ejemplos etiquetados $(x_1, y_1), \ldots, (x_m, y_m)$, donde $x_i \in X$ y $y_i \in \{0,1\}$, encuentra un concepto $c \in \mathscr{C}$ que es consistente con los ejemplos (tal que $c(x_i) = y_i$ para todo i). Además, existe especial interés en encontrar algoritmos eficientes en este modelo.

Se mostrarán algunos ejemplos de algoritmos que puedan encontrar un concepto consistente de una "clase de conceptos" específico que sea consistente con el conjunto de datos dado.

Conjunciones Monótonas

Problema: Supóngase que el dominio es $X = \{0,1\}^m$. En este caso, cada ejemplo es un bit, un vector $x = (x_1, \ldots, x_n)$ donde $x_i \in \{0,1\}$. La clase de conceptos es el conjunto de conjunciones monótonas. Este es el conjunto de funciones definidas como el Y de algunas variables, ninguna de ellas son negadas o es negación. Por ejemplo $c(x) = x_2 \wedge x_3 \wedge x_7$ es una conjunción monótona válida.

Algoritmo: El siguiente algoritmo se utilizará para resolver este problema. Considérese todos los ejemplos positivos y encuéntrese los índices que contengan 1 para cada ejemplo de entrenamiento. Se propondrá el concepto que consista en la conjunción Y de estos índices. Si el concepto retorna 0 para cada ejemplo negativo, entonces se regresa el concepto. Si no, entonces se dice que no existe un concepto consistente.

Ejemplo:
Usemos las siguientes líneas como ejemplos de entrenamiento:

01101 +
11011 +
11001 +
00101 -
11000 -

Dado que los índices x_2 y x_5 contienen 1 para cada ejemplo positivo, se propone $x_2 \wedge x_5$ como concepto. Este concepto rechaza cada ejemplo negativo de las líneas de entrenamiento por lo que es consistente con el conjunto de datos otorgado.
Nota: Si algún concepto consistente existe para este conjunto de datos en esta clase de conceptos, este algoritmo lo encuentra. Cualquier concepto válido tiene que regresar 1 para todos los ejemplos positivos y 0 para todos los ejemplos negativos. Dado que este algoritmo propone que el concepto y los Y's para cada índice que es positivo para todos los ejemplos de entrenamiento, cualquier concepto válido deberá ser un subconjunto del subconjunto de algoritmos escogidos. Por lo tanto si un concepto valido regresa 0 sobre todos los ejemplos negativos, entonces lo hace el concepto propuesto del algoritmo.

Reducciones de Conjunciones Monótonas
Para los próximos ejemplos se considerará que las clases de concepto se reducirán a la clase de conjunciones monótonas. Se continuará asumiendo que $X = \{0,1\}^n$

Disyunciones monótonas
En este ejemplo, la clase de conceptos es el conjunto de disyunciones: los conceptos definidos como el O de las variables no negadas. Usando la ley de Morgan que dice que $\overline{x_i \wedge x_j} = \bar{x_i} \vee \bar{x_j}$, equivalentemente $x_i \vee x_j = \neg(\neg x_i \wedge \neg x_j)$, se puede reducir este ejemplo para encontrar una conjunción monótona por girar la etiqueta y reducir en pedazos cada ejemplo de entrenamiento. Por ejemplo, $x_2 \vee x_5$ es equivalente a $\overline{\bar{x_2} \wedge \bar{x_5}}$ o a $x_2 \vee x_5 = \neg(\neg x_2 \wedge \neg x_5)$. Si $x_2 \vee x_5$ es un concepto consistente de la clase de disyunciones, podemos encontrar la conjunción $\overline{\bar{x_2} \wedge \bar{x_5}}$ que es consistente con el conjunto de datos formado por girar la etiqueta y reducir en pedazos cada ejemplo de entrenamiento del conjunto de datos.

Conjunciones no monótonas
Es el concepto de clases el conjunto de todas las conjunciones. Esto se reduce a encontrar una conjunción monótona formando un nuevo conjunto de datos. Se formará $\mathbf{z}_i$ negando cada pedazo del ejemplo original $\mathbf{x}_i$; entonces se concatena $\mathbf{x}_i$ y $\mathbf{z}_i$ para formar el ejemplo correspondiente en el nuevo conjunto de datos. Por ejemplo, $x_1 \wedge (\neg x_3) \wedge x_7$, esto se reduce a encontrar una conjunción monótona formando un nuevo conjunto de datos, es decir, se crea una variable z_i negando cada pedazo de la variable original x_i, entonces se concatena x_i's y z_i's para formar el nuevo ejemplo correspondiente en el conjunto de datos, usando 1101 mapearía a 11010010 en el nuevo conjunto de datos, $x_1 \wedge (\neg x_3) \wedge x_7$ se convierte en $c(\mathbf{x}) = x_1 \wedge z_3 \wedge x_7$.

k-CNF
Es la clase de conceptos el conjunto de k-CNFs, las cuales son las claúsulas Y y el O, cada una contiene a lo más k términos. Por ejemplo, $(x_1 \wedge x_2) \vee (x_1 \wedge x_2 \wedge x_3)$ es una 3-CNF que consiste en dos cláusulas, o $(x_3 \vee x_4) \wedge ((\neg x_1) \vee x_5 \vee x_8)$ también es una 3-CNF. Se asumirá que k es un

número pequeño. Esto se reduce a encontrar una conjunción monótona creando una variable dummy para cada conjunción posible.
Nota: Aunque el altoritmo sea $O(n^k)$, donde n es el número de pedazos en cada ejemplo, la suposición será que dado que k es pequeña lo hace que esté bien.

2-términos DNF
Es la clase de conceptos 2-términos DNFs. Son las funciones definidas por el O de dos cláusulas, en donde cada cláusula Ys junta varios números de variables. A esto se le llama un problema NP-dificil, el cuál es interesante porque usa álgebra booleana donde se puede escribir una terna-2 DNF como una 2-CNF. Aunque se resuelva el k-CNF en un tiempo polinomial, encontrar un 2-término DNF es NP- difícil.

DNF
Es la clase de conceptos de todos los DNFs, las cuáles son las cláusulas O e Y. Para resolver este se tiene que primero formar una conjunción para cada ejemplo positivo, donde la conjunción es verdadera sólo para el ejemplo correspondiente, y todas las conjunciones juntas O.
Pero esto crea soluciones que son largas y nada elegantes, y parece muy fácil para contar como "aprendizaje".

Concepto de Clases basada en conjuntos
En los ejemplos previos, se consideran clases de conceptos que consisten en funciones. En los ejemplos siguientes consideraremos clases de conceptos que consisten en conjuntos.

Rectángulos alineados al eje
Se asume que el dominio es $X = \mathbb{R}^2$, y una clase de conceptos consistente de todos los rectángulos alineados al eje. Se requiere encontrar un rectángulo tal que todos los ejemplos positivos cigan dentro de este y todos los ejemplos negativos caigan fuera. Para encontrar un ejemplo consistente, se puede escoger el ejemplo positivo mejor del lado izquierdo, el mejor del lado derecho, el mejor del lado superior, el mejor del lado inferior y dibujar el rectángulo más pequeño posible de esos puntos. Si algún ejemplo negativo cae dentro del rectángulo, se establece que no existe un concepto consistente.

Funciones lineales de referencia
En este ejemplo, se asume que el domino es $X = \mathbb{R}^n$, y una clase de conceptos de todas las funciones lineales de referencia. Un concepto consistente es $\mathbf{w} \cdot \mathbf{x} = b$ tiene que tener la propiedad de que $\mathbf{w} \cdot \mathbf{x_i} > b$ para todos los ejemplos positivos x_i, y $\mathbf{w} \cdot \mathbf{x} < b$ para todos los ejemplos negativos. Se puede formular esto como una programación lineal, tal que se pueda resolver este problema dentro de un solver de programación lineal.

Análisis del Modelo Consistente
En las secciones previas se observa que este modelo de aprendizaje no fomenta explícitamente un algoritmo que generalice los ejemplos que ve, y las soluciones no son calificadas como aprendizaje. También, los algoritmos no permiten ruido en los datos. Esto motiva a que el siguiente modelo de aprendizaje, donde se basa en definiciones de probabilidad.

Repaso de Probabilidad
Un *evento* es resultado probabilístico, tal que girar una moneda y el resultado sea "águila". Una

variable aleatoria es una variable que toma los valores posibles, como una variable que representa los resultados de un dado. Una *distribución* es una función de probabilidad $\mathbf{P}[\mathbf{X} = \mathbf{x}]$ tal que para todo x discretos, $\mathbf{P}[\mathbf{X} = \mathbf{x}] \geq \mathbf{0}$, $\sum_x \mathbf{P}[\mathbf{X} = \mathbf{x}] = \mathbf{1}$.
El *valor esperado o esperanza* de una variable aleatoria X está definido por: $\sum_x \mathbf{P}[\mathbf{X} = \mathbf{x}]\mathbf{x}$, esta es una función lineal.
La probabilidad condicional se puede definir como

$$\mathbf{P}[\mathbf{a}|\mathbf{b}] = \frac{\mathbf{P}[\mathbf{a} \vee \mathbf{b}]}{\mathbf{P}[\mathbf{b}]}$$

Se dice que a y b son independientes si al saber información de uno no se sabe más del otro: $P[a|b] = a$ (o, de manera equivalente, $\mathbf{P}[\mathbf{a}]\mathbf{P}[\mathbf{b}]$). Se dice que dos valriables aleatorias X y Y son independientes si $\forall x, y$, los eventos $X = x$ y $Y = y$ son independientes, si es verdad, entonces $E[XY] = E[X]E[Y]$.

1.2. El Modelo de problemas de aprendizaje aproximadamente Correcto (PAC)

Como se ha visto, el modelo de consistencia deja que desear. Por lo que se busca en otro modelo de aprendizaje.
Primero, se define una regla que resulte de un algoritmo de aprendizaje como la hipótesis h y se hacen tres suposiciones:

1. Se supone que para cada ejemplo se escoge de manera independiente de una desconocida, fija y arbitraria distribución $\mathscr{D}$.

2. Se asume que los ejemplos seleccionados para las pruebas se seleccionen de la misma manera.

3. Se asume que los ejemplos son etiquetados de acuerdo a un concepto objetivo $c : X \to [0,1]$ que es desconocido y pertenece a la clase de $\mathscr{C}$ considerada

Para evaluar una hipótesis h, se define el error de h con respecto a una distribución $\mathscr{D}$ como $e(h) = \mathbf{P}_{\mathbf{x} \in \mathbf{D}}[\mathbf{h}(\mathbf{x}) \neq \mathbf{c}(\mathbf{x})]$ que es la probabilidad de que h clasifique mal una ejemplo aleatorio de prueba de $\mathbf{x}$.
Se requiere encontrar una hipótesis con un error pequeño sobre el conjunto de prueba, esto es, se quiere una hipótesis "aproximadamente correcta". Dado que los ejemplos de entrenamiento están seleccionados de manera aleatoria, es posible tener mala suerte y que estos ejemplos estén muy sesgados; para tomar en cuenta esto buscamos algo que sea "'probablemente aproximadamente correcto", que significa que con una alta probabilidad sobre la selección del conjunto de entrenamiento, se requiere que la hipótesis sea aproximadamente correcta. Entonces, se busca el modelo probablemente aproximadamente correcto.

1.2.1. El modelo general

Definición del modelo PAC

Elementos básicos del modelo PAC:

- Sea X el **espacio de instancias o insumos**

- Sea $c \subset C$ un **concepto** sobre X, este elemento se puede pensar como una el conjunto de todas las instancias que positivametne ejemplifiquen una regla.
    - Equivalentemente, podemos definir un concepto como una mapeo booleando $c: X \to \{0,1\}$, con $c(x) = 1$, donde x es una ejemplo positivo, y $c(x) = 0$ donde x es un ejemplo negativo.
- Sea $\mathscr{C}$ la **clase de conceptos** en X
- Se conoce como **concepto objetivo** a cualquier $c \in \mathscr{C}$, donde a través del algoritmo de aprendizaje conoce la clase objetivo $\mathscr{C}$
- Sea $\mathscr{D}$ una función de distribución probabilidad sobre X
- $\mathscr{D}$ es la **distribución objetivo** si h es un concepto en X, entonces la distribución $\mathscr{D}$ provee una medida de **error** natural entre h y el concepto objetivo c:

$$e(h) = \mathbf{P}_{x \in \mathscr{D}}[c(x) \neq h(x)]$$

- Llamaremos a $EX(c, \mathscr{D})$ un procedimiento u oráculo que se evalúa en unidades y devuelve un ejemplos etiquetados como $\langle x, c(x) \rangle$, donde x se toma aleatoriamente e independientemente conforme $\mathscr{D}$.
- El algoritmo de aprendizaje satisface tres propiedades:
    - El número de llamadas a $EX(c, \mathscr{D})$ es pequeño, en el sentido de que está acotado por una función polinomial fija.
    - El monto computación evaluado es pequeño
    - El algoritmo devolverá una **hipótesis concepto** h tal que $e(h)$ es pequeño

Definición 1: (El modelo PAC, definición preliminar) Ses $\mathscr{C}$ una clase de conceptos en x. Decimos que $\mathscr{C}$ es **PAC-aprendible** por un espacio de hipótesis $\mathscr{H}$ si existe un algoritmo $\mathscr{A}$ tal que :

- para cada concepto $c \in \mathscr{C}$
- para toda $\varepsilon, \delta > 0$, y
- para cualquier distribución objetivo $\mathscr{D}$ en X

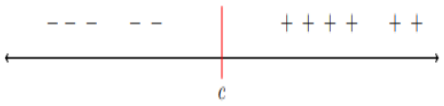
$\mathscr{A}$ requiere una sucesión $S = \langle (x_1, c(x_1)), \ldots, (x_m, c(x_m)) \rangle$ de m=poly$(\frac{1}{\varepsilon}, \frac{1}{\delta}, \ldots)$ ejemplos donde cada x_i proviene de manera independiente de $\mathscr{D}$ produce una hipótesis $h \in H$ donde $\mathbf{P}[\mathbf{e}(\mathbf{h}) \leq \varepsilon] \geq \mathbf{1} - \delta$ se cumple.
En otras palabras, si $\mathscr{A}$ puede calcular $EX(c, \mathscr{D})$ y dados ε y δ, entonces con probabilidad de al menos $1 - \delta$, $\mathscr{A}$ genera una *hipótesis concepto* $h \in \mathscr{C}$ que satisface $e(h) \leq \varepsilon$.
El tamaño de la muestra m se hace más grande cuando ε y δ se hacen pequeños, por lo que un algoritmo para tener mejor exactitud (pequeño ε) o mayor confianza (pequeña δ), comúnmente requerirá muchos datos.
Si $\mathscr{A}$ va in tiempo polinomial en $1/\varepsilon$ y $1/\delta$, decimos que C es **eficientemente PAC aprendible**.
Decimos que ε es el parámetro de error y δ el parámetro de confianza.

La hipótesis $h \in \mathscr{C}$ del algoritmo de aprendizaje PAC es entonces "aproximadamente corecto" con probablidad alta, de ahí el nombre de algoritmo Probable y Aproximadamente Correcto (PAC).

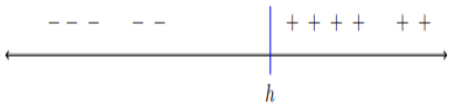
1.2.2. Ejemplos PAC

Aprendiendo de lineas-medias positivas

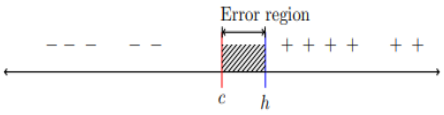
Este es un ejemplo de clases de conceptos PAC-aprendibles. Específicamente, se considera la clase de concepto de lineas-medias positivas. Una linea media positiva es un rayo que se extiende derecho (hacia infinito) a partir de un punto en los reales. Todos los valores a la derecha de este punto están etiquetados como positivos. Los valores a la izquierda están etiquetados como negativos. Por simplicidad, se denota el concepto y el punto de referencia con c. Para este ejemplo, el dominio es $X = \mathbb{R}$, y el espacio de hipótesis y la clase de conceptos son las lineas medias positivas, es decir, $\mathscr{H} = \mathscr{C} = \{\text{lineas-medias positivas}\}$.



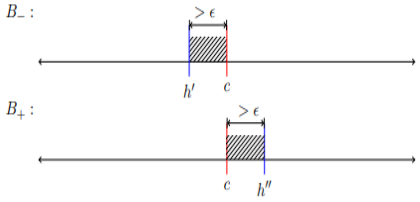
Una aproximación para escoger una hipótesis sería tomar la media aritmética del ejemplo del valor negativo más grande observado y el ejemplo del valor positivo observado. Cualquier valor entre los mencionados valores extremos sería una hipótesis consistente. Este es un ejemplo:



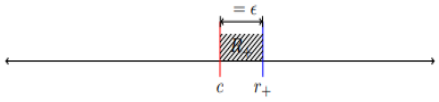
Solo los puntos fuera del intervalo $[c,h]$, la "región de error", será etiquetada apropiadamente por h. Dado que $\forall x \in [c,h]$, $c(x) \neq h(x)$ se cumple, $e(h)$ será la masa de probabilidad en $[c,h]$. En la imagen se observa que el concepto objetivo c y una hipótesis h están sobre la recta numérica. La región $[c,h]$ es el error.



Refiérase a h con el error más grande que ε como "ε-malo". El aprendizaje-PAC puede también ser refraseado al requerir h que sea ε-bueno con probabilidad de al menos $1-\delta$. Si h está muy lejos de hacia la izquierda o de la derecha de c, entonces el error será muy alto y mayor que ε. Se ilustrarán casos malos, donde se nota de inmediato que son escenarios simétricos. Se deberá probar que la mayoría de los casos, esto no pasa. Esto es que $\mathbf{P}[\mathbf{e}(\mathbf{h}) > \varepsilon] \leq \delta$. Aquí se muestra que el concepto objetivo c y dos ejemplos malos B_- y B_+ con hipótesis h' y h'', respectivamente.



Ahora considérese la probabilidad de B_+. Imaginése que el punto se encontró barriéndo desde el punto c hasta la región de probabilidad es ε. Se puede llamar este punto r_+, y la región $[c, r_+]$ como R_+. Como se muestra a continuación:



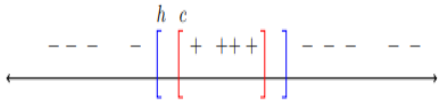
Claramente, si un ejemplo de entrenamiento cayera dentro de R_+, entonces $h < r_+$, el error sería menor que ε, y B_+ no ocurriría. Entonces, la probabilidad de B_+ sería a lo más la probabilidad de que ningún punto caiga en $[c, r_+]$. Un punto x_1 cae en R_+ con probabilidad ε, entonces $\mathbf{P}[\mathbf{x_1} \notin \mathbf{R}_+] = \mathbf{1} - \varepsilon$. Dado que todos los puntos son independientes e idénticamente distribuidos (i.i.d), se tiene que:

$$\mathbf{P}[\mathbf{x_1} \notin \mathbf{R}_+ \wedge \mathbf{x_2} \notin \mathbf{R}_+ \wedge \ldots \mathbf{x_m} \notin \mathbf{R}_+] = \mathbf{P}[\mathbf{x_1} \notin \mathbf{R}_+] \cdot \mathbf{P}[\mathbf{x_2} \notin \mathbf{R}_+] \cdots \mathbf{P}[\mathbf{x_m} \notin \mathbf{R}_+] = (\mathbf{1} - \varepsilon)^{\mathbf{m}}$$

Tal cual, la probabilidad de B_+ es a a lo más la probabilidad de todos esos, o bien $\mathbf{P}[\mathbf{B}_+] \leq (\mathbf{1} - \varepsilon)^{\mathbf{m}}$. Recálquese que hay dos eventos simétricos, B_+ Y B_-. Por la cota de unión se sabe que $\mathbf{P}[\mathbf{B}_+ \vee \mathbf{B}_-] \leq \mathbf{P}[\mathbf{B}_+] + \mathbf{P}[\mathbf{B}_-]$ y entonces $\mathbf{P}[\mathbf{B}_+ \vee \mathbf{B}_-] \leq \mathbf{2}(\mathbf{1} - \varepsilon)^{\mathbf{m}}$. Y $\forall x, 1 + x \leq e^x$, se sabe que $\mathbf{P}[\mathbf{B}_+ \vee \mathbf{B}_-] \leq \mathbf{2e}^{-\varepsilon \mathbf{m}}$, que se requiere que sea δ a lo mucho. Para satisfacer este requerimiento, tiene que pasar que $m \geq \frac{1}{\varepsilon} \ln \frac{2}{\delta}$. Retomar que $\mathbf{P}[\mathbf{e}(\mathbf{h}) > \varepsilon] = \mathbf{P}[\mathbf{B}_+ \vee \mathbf{B}_-]$, y que $\mathbf{P}[\mathbf{e}(\mathbf{h}) > \varepsilon] \leq \delta]$ si $m \geq \frac{1}{\varepsilon} \ln \frac{2}{\delta}$. Por lo que se ha demostrado que $\mathscr{C}$ es PAC-aprendible por $\mathscr{H}$. □

Aprendiendo intervalos

Considérese ahora que la clase de conceptos $\mathscr{C} = \{\text{Intervalos en } \mathbb{R}\}$, y que otra vez, $\mathscr{C} = \mathscr{H}$. Como se muestra en la figura debajo, un concepto objetivo $c \in \mathscr{C}$ es un intervalo $[c_L, c_R] \in \mathbb{R}$. Todos los valores que caen dentro del intervalo son etiquetadas como positivas. Todos los otros valores son etiquetados negativos. El algoritmo $\mathscr{A}$ debe encontrar una hipótesis $h \in \mathscr{H}$, y esto se puede lograr de una manera similar como en el ejemplo previo. La demostración de que $\mathscr{A}$ es PAC-aprendible por $\mathscr{H}$ es también similar.



De la misma manera barremos un intervalo con una probabilidad de $\frac{\varepsilon}{2}$ de cada lado de c_L, y nuevamente para c_R. Los dos casos son simétricos. Claramente, el caso de c_L es el mismo que el caso de las lineas medias positivas previas, y se puede decir lo mismo para c_R. Lo demás se deja de ejercicio para el lector.

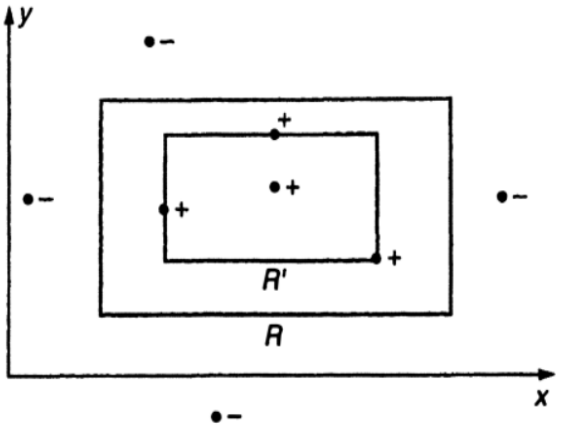
El juego de aprendizaje del rectángulo

Número de jugadores: 1
Modo de juego: El jugador recibe información sobre **R** dado lo siguiente:
1. De un punto p que es seleccionado de una distribución de probabilidad fija $\mathscr{D}$
2. Se le avisa al jugador si el punto está contenido en **R** o no
Objetivo: Es aprender de un rectángulo desconocido **R** (objetivo) en $\mathbb{R}^2$, donde sus lados son

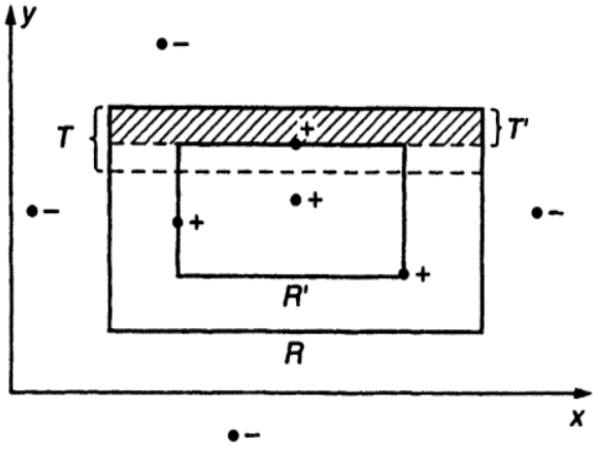
paralelos a los ejes verticales y horizontales, utilizando los menores ejemplos posibles y los mínimos cálculos posibles para encontrar un rectángulo hipótesis $\mathbf{R}'$ que sea una aproximación a $\mathbf{R}$.
Técnicamente se mide el error de $\mathbf{R}'$ como la probabilidad de que un punto aleatorio de $\mathscr{D}$ caiga en la región $\mathbf{R} \triangle \mathbf{R}' = (\mathbf{R} - \mathbf{R}') \cup (\mathbf{R}' - \mathbf{R})$.

Estrategia

1. Solicitar una cantidad *m suficientemente grande* de ejemplos aleatorios
2. Escoger la hipótesis del rectángulo (alineado a los ejes horizontal y vertical) $\mathbf{R}'$ que ajuste lo mejor posible a los ejemplos que pertenecen a $\mathbf{R}$. Obviamente, si no hay ejemplos en $\mathbf{R}$, $\mathbf{R}' =$, un ejemplo de hipótesis es el siguiente:



Notemos que para cualquier rectángulo $\mathbf{R}$ y distribución $\mathbf{D}$, y para valores pequeños $\varepsilon > 0$ y $\delta \leq \frac{1}{2}$, para un valor conveniente m podemos asegurar que con la probabilidad de al menos $1 - \delta$, que el rectángulo más ajustado $\mathbf{R}'$ tiene un error a lo más de ε con respecto a $\mathbf{R}$ y $\mathbf{D}$.
Sabemos que $\mathbf{R}'\mathbf{R}$ $\mathbf{R} \triangle \mathbf{R}' = \mathbf{R} - \mathbf{R}'$, justo esta diferencia se puede expresar como unión de cuatro pedazos rectangulares. Si observamos la probabilidad de que un punto no caiga en alguno de esos pedazos rectangulares bajo $\mathbf{D}$, y garantizáramos que es por mucho $\varepsilon/4$, entonces el error de $\mathbf{R}'$ es a lo mucho ε.
Analicemos el peso de un pedazo rectangular T'. Definiendo T como el pedazo rectangular superior interior de $\mathbf{R}$ que encuadra exactamente el peso $\varepsilon/4$ bajo $\mathbf{D}$.



Ahora bien si el ancho de T' es mayor que T entonces el peso de T' es myor que $\varepsilon/4$, cosa que no pasa en la imagen.
Dada la definición de T, la probabilidad de que el pedazo de la distribución $\mathbf{D}$ no caiga en la región T es exactamente $1-\varepsilon/4$. Por lo tanto, la probabilidad de que m eventos independientes de $\mathbf{D}$ no caigan en la región T, es exactamente $(1-\varepsilon/4)^m$. Usando la misma lógica para las otras 3 regiones rectangulares de $\mathbf{R}-\mathbf{R}'$, por la probabilidad de las uniones, la probabilidad de que uno de los cuatro pedazos tenga un peso mayor que $\varepsilon/4$ es a lo mucho $4(1-\varepsilon/4)^m$.
Ahora bien, escogiendo m para que satisfaga que $4(1-\varepsilon/4)^m \leq \delta$, entonces con probabilidad $1-\delta$ sobre los m ejemplos aleatorios, el peso de error de la región $\mathbf{R}-\mathbf{R}'$ será acotado por ε como se supuso anteriormente. Si usamos la desigualdad $(1-x) \leq e^{-x}$ observamos que para cualquier m que satisfaga que $4(1-\varepsilon/4)^m \leq 4e^{-\varepsilon m/4} \leq \delta$ también cumple la condición anterior. Despejando m tenemos que $m \geq (4/\varepsilon)\log(4/\delta)$.
Finalmente, hemos visto que el algoritmo de ajuste más cerrado toma una muestra de al menos $(4/\varepsilon)\log(4/\delta)$ ejemplos para crear la hipótesis del rectángulo $\mathbf{R}'$, podemos asegurar que con probabilidad de al menos $1-\delta$, $\mathbf{R}'$ clasificará mal el nuevo punto acorde con la nueva distribución, con probabilidad de a lo más de ε.

Ciertos puntos a destacar:

1. El análisis previo aplica para cualquier probabilidad de distribución fija
2. La cota del tamaño de muestra se comporta conforme a lo esperado, es decir, para mayor precisión se disminuye la ε o mejor confianza se disminuye la δ, para eso el algoritmo necesita más ejemplos para cubrir esas demandas.
3. El algoritmo es eficiente, el tamaño de la muestra requerida es una función que crece lentamente de $1/\varepsilon$ y $1/\delta$ (lineal y logarítmico, respectivamente), una vez que la muestra es dada, el cálculo de la hipótesis de ajuste puede resultar muy rápido.

Dado el ejemplo del rectángulo hay elementos esenciales para el modelo PAC.

1. EL objetivo del juego de aprendizaje es aprender un conjunto objetivo desconocido no arbitrario.
2. El aprendizaje ocurre con base en probabilidades.
3. La hipótesis del "aprendido" es evaluada relativa a la configuración de probabilidad en el que el entrenamiento ocurre y se permiten hipótesis que son sólo aproximaciones hacia el objetivo
4. El interés se basa en una solución eficiente

1.2.3. Generalización para capacidad de aprendizaje PAC para $|\mathscr{H}|$ finito

Hasta el momento, se han considerando algunas demostraciones de la capacidad de aprendizaje PAC o "PAC-aprendibilidad" para casos muy específicos. Esta es una manera muy ineficiente de demostrar las cosas. El interés en conocer cuando la consistencia es suficiente para el aprendizaje en un modelo general (PAC-aprendizaje). A continuación se mostrará el teorema de PAC-aprendibilidad para espacios de hipótesis de cardinalidad finita.

Teorema 1 Supóngase que el algoritmo $\mathscr{A}$ encuentra una hipótesis $h_{\mathscr{A}} \in \mathscr{H}$ consistente con m ejemplos donde $m \geq \frac{1}{\varepsilon}(\ln|\mathscr{H}| + \ln\frac{1}{\delta})$. Entonces $\mathbf{P}[\mathbf{e}(\mathbf{h}_{\mathscr{H}}) > \varepsilon] \leq \delta$.

Nótese que este teorema no involucra directamente una clase de concepto $\mathscr{C}$, pero aplica siempre que se gestione encontrar una hipótesis consistente. Se puede refrasear como sigue: con probabilidad de al menos $1-\delta$, si $h_{\mathscr{A}} \in \mathscr{H}$ es consistente, entonces $e(h_{\mathscr{A}}) \leq (\ln|\mathscr{H}| + \ln\frac{1}{\delta})/m$.
Este término nuevo, $\ln|\mathscr{H}|$ es interesante, y está relacionado a la noción de complejidad o que no tan simple es la hipótesis. Específicamente, es una medida de complejidad de $\mathscr{H}$. Cuando la base para este logaritmo es 2, el término es el número de pedazos que se necesitan para escribir nombres para cada hipótesis $h \in \mathscr{H}$. También se puede pensar en $\ln|\mathscr{H}|$ como medida de la longitud de hipótesis en $\mathscr{H}$. Cabe destacar, que esto se refiere solamente a la complejidad de $\mathscr{H}$, no de las hipótesis individuales.

Para explorar más esto, el Dr. Shchapire le preguntó a sus estudiantes que consideraran el problema de determinar las etiquetas binarias (0 o 1) para el conjunto de enteros $\{1,2,\ldots,30\}$. Después de que todos escribieran las etiquetas para el conjunto, el Dr. Schapire, dividió los ejemplos en dos conjuntos: (1) $\{1,2,\ldots,10\}$ para el conjunto de entrenamiento, y $\{11,12,\ldots,20\}$ para el conjunto de prueba. La hipótesis del estudiante con el menor error de entrenamiento era de 20 % pero un 40 % en el conjunto de prueba. Sin embargo, la clase sin saberlo, las etiquetas apropiadas habían sido generadas por lanzar una moneda. Por lo cual, incluso las hipótesis con bajo error de entrenamiento, se espera que mal clasifiquen la mitad de los ejemplos de prueba. Cuando el espacio de hipótesis crece, también lo hace la probabilidad de encontrar hipótesis que se comporten mejor en los datos de entrenamiento por suerte. Sin embargo, dichas hipótesis siguen teniendo un error esperado del 50 % en el conjunto de pruebas. Esto significa que espacios grandes de hipótesis requieren más datos de entrenamiento para evitar estas malas hipótesis que sólo tienen error bajo de entrenamiento por suerte.

Conjunciones monótonas
Considérese $\mathscr{C} = \{\text{conjunciones monótonas de longitud } n\}$. De nuevo se hace $\mathscr{C} = \mathscr{H}$. Dado que cada h incluye o excluye una variable dada, entonces existen 2^n hipótesis y $|\mathscr{H}| = 2^n$. Se ha visto anteriormente, que existe un algoritmo $\mathscr{A}$ que encuentra hipótesis consistentes $h \in \mathscr{H}$, Por el teorema 1, $m \geq \frac{1}{\varepsilon}(n\ln 2 + \ln\frac{1}{\delta})$ implica que $\mathbf{P}[\mathbf{e}(\mathbf{h}_{\mathscr{A}}) > \varepsilon] \leq \delta$. Esto va en tiempo polinomial con respecto a n,$1/\varepsilon$ y $1/\delta$. Por lo que se ha demostrado que las conjunciones monótonas son PAC-aprendibles. □

Breve integración de conceptos:
(1) Sea h que denota la hipótesis de el espacio $\mathscr{H}$ de tamaño $|\mathscr{H}|$. $h_{\mathscr{A}}$ denota la hipótesis encontrada por el algoritmo $\mathscr{A}$.
(2) c denota el concepto de la clase de conceptos $\mathscr{C}$.
(3) Sea m que denota el tamaño del conjunto de entrenamiento $\mathscr{S}$.
(4) Sea x_i que denonta la i-ésima muestra del conjunto de entrenamiento. y $c(x_i)$ denota la correspondiente etiqueta para la muestra. Se asume que los ejemplos provienen de una distribución objetivo $x_i \sim D$.

El error de generalización de una hipótesis sobre una distribución objetivo es:

$$e(h) = \mathbf{P}[\mathbf{h}(\mathbf{x}) \neq \mathbf{c}(\mathbf{x})]$$

Una hipótesis h es consistente si etiqueta correctamente todos los ejemplos de un conjunto de entrenamiento acorde al concepto objetivo c: si $h(x_i) = c(x_i)\ \forall i \in \{1,\ldots,m\}$.

Occam's Razor

Occam's Razor provee una técnica general para mostrar que el algoritmo de aprendizaje generaliza con error bajo a nuevos datos con alta probabilidad y es probablemente aproximadamente correcto (PAC).

Teorema: Occam's Razor. Si el algoritmo $\mathscr{A}$ encuentra una hipótesis $h_{\mathscr{A}} \in \mathscr{H}$ consistente con m ejemplos donde $m > \frac{1}{\varepsilon}(\ln|\mathscr{H}| + \ln\frac{1}{\delta})$. Entonces:

$$\mathbf{P}[\mathbf{e}(\mathbf{h}_{\mathscr{A}}) > \varepsilon] \leq \delta$$

En otras palabras, la probabilidad de que el error de generalización de $h_{\mathscr{A}}$ es ε-mala y está acotada por δ.

Teorema: Equivalente. Con probabilidad de al menos $1-\delta$ si $h_{\mathscr{A}} \in \mathscr{H}$ es consistente entonces el error generalizado está acotado

$$e(h_{\mathscr{A}}) \leq \frac{\ln|\mathscr{H}| + \ln\frac{1}{\delta}}{m}$$

Se sabe que $\ln|\mathscr{H}|$ es proporcional al número de pedazos que se necesitan para describir la hipótesis en $\mathscr{H}$, esto implica que:

El aprendizaje es posible si se puede encontrar una hipótesis consistente que sea simple con los suficientes datos provistos

Capacidad de aprendizaje en el modelo consistente implica PAC-aprendibilidad

Proposición:*Si se puede encontrar una hipótesis consistente, entonces el aprendizaje es posible*
Supongamos que el algoritmo encuentra una hipótesis que sea ε-mala con una tasa de error del 1%. Si hay cientos de ejemplos de entrenamiento, la hipótesis muy seguramente tendrá un error y no será consistente. Se hará la siguiente observación generalizando para todas las hipótesis ε-malas en el espacio de hipótesis.

Observación:*Con probabilidad alta, cualquier hipótesis ε-mala será eliminada del conjunto de entrenamiento y las que queden serán ε-buenas*

PAC-aprendibilidad implica capacidad de aprendizaje en el modelo consistente

Proposición:*Si se puede hacer aprendizaje PAC apropiadamente, entonces se puede tener aprendizaje en el modelo consistente.*

1.3. La dimensión Vapnik-Chervnonenkis

Para cualquier clase de conceptos $\mathscr{C}$ sobre X (donde estos pueden ser infinitos), y cualquier $S \subseteq X$,

$$\Pi_{\mathscr{C}}(S) = \{c \cap S \colon c \in \mathscr{C}\}.$$

Equivalentemente, si $S = \{x_1, \ldots, x_m\}$ entonces veremos a $\Pi_{\mathscr{C}}$ como el conjunto de vectores $\Pi_{\mathscr{C}} \subseteq \{0,1\}^m$ definidos como

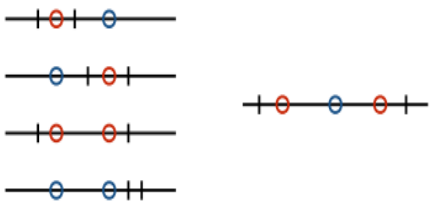
$$\Pi_{\mathscr{C}}(S) = \{(c(x_1), \ldots, c(x_m)) \colon c \in \mathscr{C}\}.$$

Entonces, $\Pi_{\mathscr{C}}(S)$ es el conjunto de todos los **comportamientos** o **dicotomías** sobre S que son inducidos o **creados** por $\mathscr{C}$. Usaremos las descripciones de $\Pi_{\mathscr{C}}(S)$ como una colección de subconjuntos de S y un conjunto de vectores intercambiables.
Si $\Pi_{\mathscr{C}}(S) = \{0,1\}^m$ (donde $m = |S|$), entonces decimos que S es desmenuzado por $\mathscr{C}$. Entonces, S es desmenuzado por $\mathscr{C}$ si $\mathscr{C}$ hace posibles todas las posibles dicotomías de S.
Ahora bien, *La **dimensión Vapnik-Chervonenkis (VC)** de $\mathscr{C}$, denotada como $VCD(\mathscr{C})$, es la cardinalidad d del conjunto más largo S desmenuzado por $\mathscr{C}$. Si arbitrariamente conjuntos grandes pueden ser desmenuzados por $\mathscr{C}$, entonces $VCD(\mathscr{C}) = \infty$*

Ejemplos de Dimensión VC

Consideraremos algunas clases de conceptos geométricos y calcularemos la dimensión VC. Para demostrar que la definición de la dimensión VC es al menos de clase d, se debe solamente encontrar un conjunto que desmenuze el conjunto de tamaño d, se debe encontrar que no hay un conjunto del tamaño $d+1$ que pueda ser desmenuzado, justo por esta razón demostrar cotas superiorer es mas complicado que las cotas inferiores.
Recálquese que la dimensión VC es un valor que estima la capacidad de un clasificador binario. Si se pueden encontrar un conjunto de n puntos, tal que puedan ser desmenuzados por el clasificador (i.e. clasificar todas las posibles 2^n etiquetados correctamente) y no se pueden encontrar ningún conjunto de $n+1$ puntos que puedan ser desmenuzados (i.e. para cualquier conjunto de $n+1$ puntos existe al menos un etiquetado ordenado tal que el clasificador no pueda separar todos los puntos correctamente), entonces la VCD es n.

Ejemplo 1. Intervalos en la linea real. Para este concepto de clase, cualquier conjunto de dos puntos puede ser demenuzado, como se muestra en la parte izquierda de la imagen, por lo que la *VCD* es al menos dos, pero si tomamos 3 puntos, estos no se pueden desmenuzar, si tomamos tres puntos como se muestra en la figura, no se puede demenuzar en el intervalo, por lo que VCD para esta clase es dos.



Ejemplo 2: Intervalos en $\mathbb{R}^2$
Sea

$h(x) = \begin{cases} 1 & \text{si } a \leq x \leq b \\ 0 & \text{en otro caso} \end{cases}$ con parámetros $(a,b) \in \mathbb{R}^2$

Considérese primero dos puntos (x_1, x_2) tal que $x_1 < x2$. Entonces existen $2^2 = 4$ posibles clasificaciones o dicotomías.

1. $x_1 : 1$, $x_2 : 1$
2. $x_1 : 0$, $x_2 : 0$
3. $x_1 : 1$, $x_2 : 0$
4. $x_1 : 0$, $x_2 : 1$

Todos los etiquetados se logran a través del clasificador h fijando los parámetros $a < b \in \mathbb{R}$ tal que

1. $a < x_1 < x_2 < b$
2. $x_1 < x_2 < a < b$
3. $a < x_1 < b < x_2$
4. $x_1 < a < x_2 < b$

respectivamente.

Ahora, consideremos tres puntos arbitrarios x_1, x_2, x_3 y asumamos, sin pérdida de generalidad que x_1, x_2, x_3, entonces no se puede tener el etiquetado $(1,0,1)$. Como en el caso 3 de arriba, las etiquetas $x_1 : 1$ y $x_2 : 0$ implican que $a < x_1 < b < x_2$, lo que implica que $x_3 > b$ y entonces la etiqueta de x_3 tiene que ser 0. Entonces el clasificador no puede desmenuzar ningún conjunto de tres puntos y la dimensión VC es 2.

Ejemplo 3: Funciones lineales de referencia en $\mathbb{R}^n$

Sea $h(x) = \begin{cases} 1 & \text{si } \mathbf{w} \cdot \mathbf{x} \leq b \\ 0 & \text{en otro caso} \end{cases}$ con parámetros $(w) \in \mathbb{R}^n$ y $b \in \mathbb{R}$ Su VCD=$n+1$

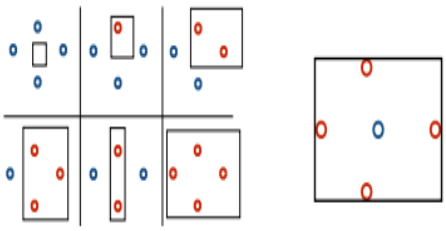
Ejemplo 4: Funciones lineales de referencia que pasan por el origen de R^n

Con respecto al ejemplo anterior $b = 0$, en este caso VCD=n.

Nótese que en ambos caso anteriores, VCD es igual al número de parámetros, pero siempre es el caso. Por ejemplo, para la clase de funciones que mapea un número real x a $\text{sign}(\sin(ax))$ con sólo un parámetro a su dimensión VCD es infinita.

Ejemplo 5: Rectángulos alineados al eje

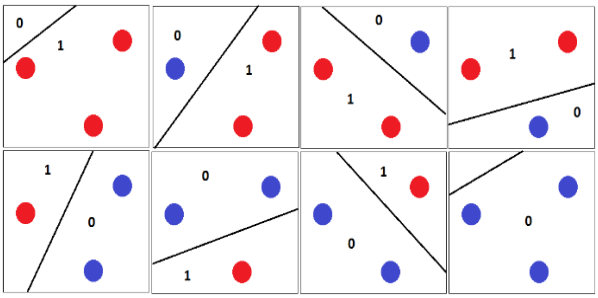
Para el caso cuando $\mathscr{H} = \{\text{rectángulos alineados al eje}\}$, $VCD = 4$, en la imagen se puede ver (del lado izquierdo) que un conjunto de 4 puntos puede ser desmenuzado por rectángulos alineados al eje. Del lado derecho, para cualquier ordenamiento de 5 puntos, se puede escoger puntos en la parte más alta superior, la parte más baja inferior, la parte más pegada a la derecha, y la más pegada a la izquierda, y asignar "+" a ellos, y a los puntos que falten "-". Ningún rectángulo que contenga los puntos "+" tiene que contener "-", lo que significa que este conjunto no puede ser desmenuzado.



Ejemplo 6: Hiperrectángulos en R^n
Bajo la lógica anterior, VCD=$2n$.

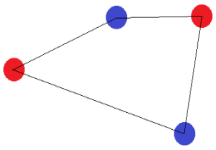
Ejemplo 7: Hiperplanos

Considérese hiperplanos, es decir, líneas en dos dimensiones (2D).
Es más fácil encontrar un conjunto de tres puntos que puedan ser clasificados correctamente no importa cómo sean etiquetados, veamos las 2^3 combinaciones.

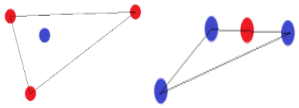


Se observa que para todas las posibles etiquetas se puede encontrar un hiperplano que las separa perfectamente.
Sin embargo, no se puede encontrar un conjunto de cuatro puntos tal que se pueda clasificarlos en 2^4 posibles etiquetas correctamente. Veamos un argumento visual.

Supóngase que los 4 puntos forman una figura de 4 lados, entonces es imposible encontrar un hiperplano que pueda separar los puntos correctamente si se etiquetan las esquinas opuestas con la misma etiqueta:



Si no forman una figura de 4 lados, entonces hay dos casos acotados: Los puntos externos forman un triángulo o todos forman una linea recta. En el caso del triángulo, es fácil ver el etiquetado, donde el punto interno es etiquetado diferente de los otros no se puede lograr.



En el caso de la línea del segmento, la misma idea aplica, si los puntos finales son etiquetados diferentes de uno de los otros puntos, no pueden ser separados por un hiperplano.
Dado que se cubren todas las posibles formaciones para cuatro puntos en 2D, se concluye que no existen 4 puntos que no se puedan desmenuzar. Entonces la VCD es 3.

Para una clase de hipótesis $\mathscr{H}$ con $\mathrm{VCD}(\mathscr{H}) = d$ implica que $m = O(\frac{1}{\varepsilon}(\ln(1/\delta) + d\ln(1/\varepsilon))$ es suficiente para aprendizaje PAC. Esto provee una cota superior en la complejidad de la muestra que es lineal en la dimensión VC. Se verá una cota inferior en la complejidad de la muestra, para saber qué tantos datos son necesario para el aprendizaje en un modelo particular. Se utilizará la dimensión VC para esto.

Una cota inferior en la complejidad de la muestra

Teorema 2: Para algún algoritmo $\mathscr{A}$, existe un $c \in \mathscr{C}$ y una distribución $\mathscr{D}$ tal que si $\mathscr{A}$ obtiene una muestra del tamaño $\leq d/2$, entonces

$$P_S[e(h_{\mathscr{A}}) > 1/8] \geq 1/8$$

Esto es equivalente a decir que si se requiere $\varepsilon < 1/8$ y $\delta \leq 1/8$ entonces se necesita $m > d/2$. La demostración va en el sentido de escoger c de manera uniforme tal que todas las posibilidades de manera desmenuzada sean igualmente probables.

1.4. Generalizando el modelo PAC

A menudo no es posible encontrar hipótesis que sean consistentes con el conjunto de entrenamiento. Puede ser que el concepto verdadero no está en la clase de hipótesis, o que es computacionalmente difícil encontrar una relación consistente, o que no existen relaciones funcionales entre las instancias y sus etiquetas. En el último caso la relación puede ser fundamentalmente probabilística, como en el caso del clima. Esto motiva a ajustar el modelo de referencia de PAC.

Marco viejo	Marco nuevo
Observar x, $c(x)$ con $x \in X$ y $c \in \mathscr{C}$	Observar x, y donde $(x,y) \sim \mathscr{D}$
$\mathscr{D}$ sobre X	$\mathscr{D}$ sobre $X \times \{0,1\}$
$e(h) = \mathbf{P}(\mathbf{h}(\mathbf{x}) \neq \mathbf{c}(\mathbf{x}))$	$e(h) = \mathbf{P}_{(\mathbf{x},\mathbf{y})}(\mathbf{h}(\mathbf{x}) \neq \mathbf{y})$

En el nuevo marco, se tiene $(x,y) \sim \mathscr{D}$, donde $\mathscr{D}$ es una distribución sobre $X \times \{0,1,\}$. Para contrastar los marcos, obsérvese que:

$$\mathbf{P}(\mathbf{x},\mathbf{y}) = \mathbf{P}(\mathbf{x}) \cdot \mathbf{P}(\mathbf{y}|\mathbf{x})$$

Donde la distribución de probabilidad es con respecto a $\mathscr{D}$. Se piensa que (x,y) se genera como un par. Por un lado, x se genera primero y luego la etiqueta y generada probabilísticamente, que depende de x. Antes se tenía $\mathbf{P}(\mathbf{y} = \mathbf{1}|\mathbf{x}) = \mathbf{0}$ o 1. En el nuevo marco se tiene un valor en $[0,1]$. Para manejar esta nueva distribución de datos hay que generalizar la noción de error.

Hipótesis óptima de Bayes

Para entender mejor el error bajo el nuevo marco, considérese la pregunta: si no se restringe a h, ¿qué tan pequeño puede ser la generalización del error?
Como un ejemplo muy sencillo, considérese el lanzamiento de una moneda que sea águila con probabilidad p y sol con probabilidad $1-p$. Si se requiere adivinar el resultado, la predicción óptima sería

$$\begin{cases} \text{águila} & \text{si } p > 1/2 \\ \text{sol} & \text{si } p < 1/2 \end{cases}$$

Si $p = 1/2$, escoger águila o sol sería lo óptimo. Esto sugiere que en general, cuando se asignan clasificaciones a través de una hipótesis, la hipótesis óptima h_{opt} sería

$$h_{\text{opt}}(x) = \begin{cases} 1 & \text{si } \mathbf{P}(\mathbf{y} = \mathbf{1}|\mathbf{x}) > \mathbf{1/2} \\ 0 & \text{si } \mathbf{P}(\mathbf{y} = \mathbf{1}|\mathbf{x}) < \mathbf{1/2} \end{cases}$$

Esta es conocida como el "Clasificador óptimo de Bayes" o "Regla de decisión óptima de Bayes". El "Error de Bayes" es el mínimo error que se puede obtener:

$$e(h_{\text{opt}}) = \min_{\forall h} e(h)$$

Aprendizaje PAC

El error de Bayes es útil cuando se trata de entender el nuevo modelo. Sin embargo, usualmente la gente no apunta a encontrar el error de Bayes, solamente se dedica a encontrar la mejor hipótesis dada una clase de hipótesis $\mathscr{H}$, $\min_{h\in} e(h)$.
Considérese una muestra $S = \langle (x_1, y_1), \ldots, (x_m, y_m) \rangle$ donde $(x_i, y_i) \sim \mathscr{D}$. Se puede definir el error empírico de la hipótesis $h \in \mathscr{H}$ como:

$$\hat{e}(h) = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}\{h(x_i) \neq y_i\}$$

y la hipótesis óptima sobre el cojunto como

$$\hat{h} = \arg\min_{h\in\mathscr{H}} \hat{e}(h)$$

Supóngase que se puede probar que el error empírico de h es aproximadamente igual a su verdadero error, entonces

$$\forall h \in \mathscr{H}: \quad |\hat{e}(h) - e(h)| \leq \varepsilon$$

Será suficiente probar que $\hat{h}$ tiene error bajo de generalización:

$$\begin{array}{rcll} e(\hat{h}) & \leq & \hat{e}(\hat{h}) + \varepsilon & \text{por la suposición} \\ & \leq & \hat{e}(h) + \varepsilon & \forall h, \text{ dado que } \hat{h} = \arg\min_{h\in\mathscr{H}} \hat{e}(h) \\ & \leq & e(h) + 2\varepsilon & \text{por la suposición} \end{array}$$

Dado que se cumple para todo $h \in \mathscr{H}$, se cumple para uno con el error mínimo de generalización. Por lo tanto, $\hat{h}$ tendrá un error de generalización que está dentro de 2ε de la mejor hipótesis en $\mathscr{H}$.