

Diapositivas-Parcial-1-Traducida...



Anónimo



Aprendizaje Automatico



3º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de
Telecomunicación
Universidad de Granada



VITALDENT
Queremos verte sonreír

PRESUME DE SONRISA
Escanea este código y estrena tu ortodoncia invisible

AL TERMINAR TU TRATAMIENTO
BLANQUEAMIENTO DENTAL GRATIS

*Blanqueamiento bajo prescripción médica. Promoción no acumulable a otros descuentos y/o promociones. CSD076

VITALDENT

Queremos verte sonreír

PRESUME DE SONRISA

Escanea este código y estrena tu ortodoncia invisible



AL TERMINAR TU TRATAMIENTO
BLANQUEAMIENTO* DENTAL GRATIS

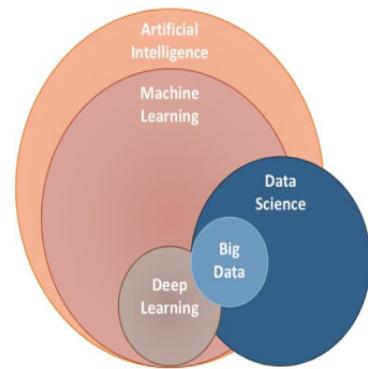
*Blanqueamiento bajo prescripción médica. Promoción no acumulable a otros descuentos y/o promociones. CS10715

Estudiar sin publi es posible.

Compra Wuolah Coins y que nada te distraiga durante el estudio.



Aprendizaje Automático



Learning from Data (Machine Learning)



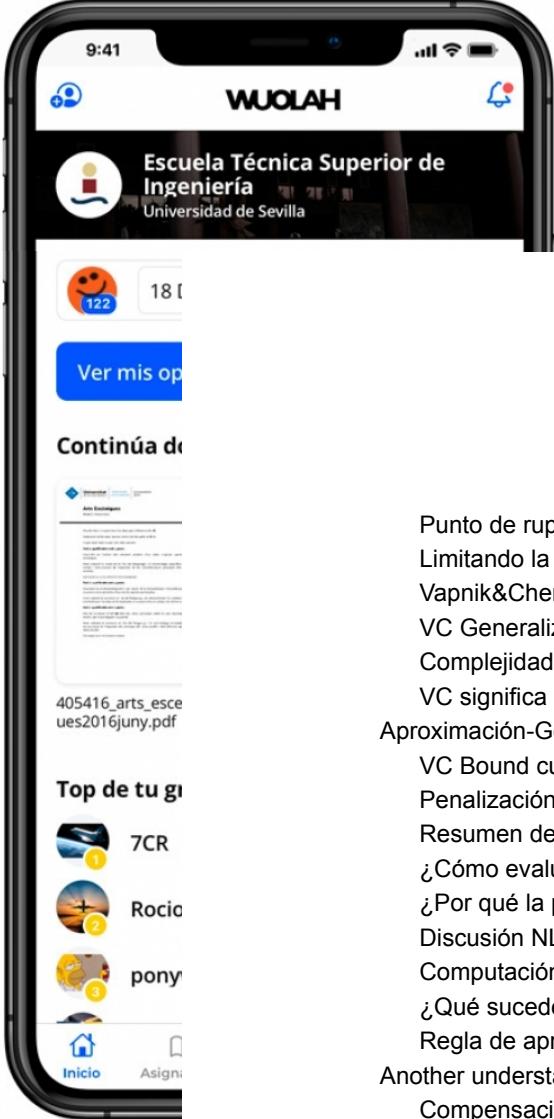
WUOLAH

Índice

Sesión 1	6
Enfoques de aprendizaje	6
Learning vs Design: What Learning is not!	6
Machine Learning definitions	7
Three main ML Paradigms	7
Aprendizaje supervisado	7
Let's check	8
Aprendizaje no supervisado	9
Formalizando el enfoque	9
Elementos principales de una tarea de aprendizaje	10
Aprendizaje en acción	10
Diagrama de la configuración de aprendizaje inicial	10
Sesión 2	11
Modelos lineales: configuración general	11
Regresión	11
Elementos del problema del crédito	11
Regresión lineal	11
¿Cuándo es óptima la pérdida cuadrática?	12
Ilustración de Regresión Lineal	12
Expresión matricial para Ein	12
Minimización Ein: ecuaciones normales	12
Math details	13
Some vector notation	13
Algoritmo de Regresión Lineal	13
Propiedades de Regresión Lineal	13
Alternativa: Descenso del Gradiente	14
¿Cómo fijar η ? (learning rate, tasa de aprendizaje)	14
¿Qué óptimo logramos?	14
Estimación de parámetros: Descenso de Gradiente	15
Stochastic Gradient Descent: SGD (Descenso de Gradiente Estocástico)	15
Un Algoritmo de Regresión Iterativo	15
Método de Newton: Una cura para la oscilación	16
En la práctica	16
Características categóricas y valores perdidos	16
Clasificación	17
Representación de entrada	17
Ilustración de características	17
Un caso muy simple	17
Perceptrón: dos clases	17
Perceptrón o Separador Lineal	18
Analizamos un detalle más	18
Algoritmo de aprendizaje del Perceptrón (PLA)	18

Algoritmo de Clasificación Lineal	18
Cómo trabaja el Perceptrón Cómo trabaja PLA	19
Modelo de Clasificación Lineal	19
Clasificación Lineal: $\text{Ein}(g) \approx 0?$	19
The 'Pocket' Algorithm	20
Límite de Clasificación (Classification Boundary) - PLA vs Pocket	20
Regresión Lineal para clasificación	21
Límite de Regresión Lineal	21
Etiquetas ruidosas: Una configuración general	21
Configuración de aprendizaje actual	21
Muestras ruidosas	21
Distribución objetivo	22
Configuración de aprendizaje actualizado	22
ESTIMACIÓN DE PROBABILIDAD	22
Regresión Logística (LGR)	22
Notación de Regresión Logística	23
Medida de error (pérdida)	23
Criterio de aprendizaje: ML	23
Clasificación de multietiquetas	23
La probabilidad de la muestra (K clases)	24
Clasificación de multietiquetas: SoftMax	24
Regla de Bayes	24
ERM: Regla de Aprendizaje	25
SGD + ERM: Una nueva regla de inducción	25
COSTE DE ERROR	26
Medida de error especificada por el usuario	26
Configuración de aprendizaje con medición de error y objetivo ruidoso	26
AGREGAR FUNCIONES A h: TRANSFORMACIONES NO LINEALES	27
Los predictores lineales son limitados	27
Regresión: predictores no lineales	27
Modelo lineal significa linealidad en w	27
Transformaciones no lineales (NLT)	28
Regresión Polinomial	28
Transforma los datos de forma no lineal	28
Transformaciones no lineales	29
Qué se transforma en qué	29
Ejemplo NTL	29
Funciones de error / pérdida generalizadas	29
Cálculo y Generalización	30
Different learners (A, h)	30
MÉTRICAS DE EVALUACIÓN	30
Alias y otras medidas	30
Pares de medidas y medidas compuestas	30
Evaluación de la salida (Caso Binario)	31

Curva de ROC (Caso Binario)	31
Medidas de desempeño basadas en matriz de confusión	31
Sesión 3	31
Intentemos aprovechar la aleatoriedad ...	32
Pero, ¿es suficiente la probabilidad?	32
Media de la población a partir de la media de la muestra	33
Desigualdad de Hoeffding	33
Desigualdad de Hoeffding: hechos notables	34
Configuración de aprendizaje	34
La función de error de aprendizaje	34
Relacionar la papelera con el aprendizaje	35
Relacionar la papelera con el aprendizaje: los datos	35
Relacionar el contenedor con el aprendizaje: los datos	36
Modelo de contenedor y aprendizaje	36
Hoeffding la desigualdad en el aprendizaje	36
Hoeffding dice que $Ein(h) \approx Eout(h)$	37
Comprensión de los resultados de la PAC	37
Aprendizaje real: modelo de aprendizaje finito	37
La desigualdad de Hoeffding para hipótesis múltiples	38
¿Ahora qué?	38
Interpretando el límite	38
Un caso particular: la hipótesis de la realizabilidad	39
Definición formal de aprendizaje PAC realizable	39
Muestra de complejidad en PAC-Learnability	39
Definición formal de aprendizaje PAC agnóstico	39
Complejidad de la muestra Agnostic-PAC	40
Viabilidad del aprendizaje versus complejidad	40
Viabilidad de aprendizaje: $E_{out} \approx 0$	40
Viabilidad del aprendizaje (H finito): Resumen	41
Sesión 4	41
Teoría de la generalización ERM: La dimensión Vapnik-Chervonenkis	42
El aprendizaje es factible para H finito	42
H- infinito: el truco de la discretización	42
¿Cuál es el problema de la desigualdad uniforme?	42
VC Generalización Bound	43
Midiendo la diversidad de H	43
La función de crecimiento	43
Función de crecimiento: ejemplo 1	44
Función de crecimiento: ejemplo 2	44
Función de crecimiento: ejemplo 3	44
Función de crecimiento: ejemplo 4	45
Función de crecimiento: ejemplo 5	45
Función de crecimiento y generalización	45



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.

Available on the
App Store

GET IT ON
Google Play

Punto de ruptura	46
Limitando la función de crecimiento	46
Vapnik&Chervonenkis: dimensión VC:	46
VC Generalization Bound	47
Complejidad de la muestra	47
VC significa penalización por complejidad del modelo	47
Aproximación-Generalización Tradeoff: una nueva perspectiva para elegir g	48
VC Bound cuantifica la aproximación frente a la generalización	48
Penalización por complejidad del modelo	48
Resumen del VC Bound	48
¿Cómo evaluar nuestro ajuste?	49
¿Por qué la prueba E debería ser un buen estimador de E out?	49
Discusión NLT	49
Computación y generalización	50
¿Qué sucede cuando dVC = ∞?	50
Regla de aprendizaje no uniforme: SRM	50
Another understanding for E out	51
Compensación de sesgo-varianza	51
Un simple problema de aprendizaje	51
Repite el experimento varias veces ...	51
¿Qué está pasando en promedio?	52
E en el punto de prueba x para los datos D	52
Compensación de sesgo-varianza	52
Compensación entre sesgo y varianza: comentarios	53
De vuelta a H 0 y H 1; y nuestro ganador es...	53
Haga coincidir el poder de aprendizaje con los datos. . .No a f	54
Curva de aprendizaje	54
Curvas de aprendizaje para regresión lineal	54
Curva de aprendizaje para regresión lineal	55

Sesión 1

Enfoques de aprendizaje

Machine Learning (computer science):

- El enfoque principal es la predicción precisa de problemas a gran escala (la generalización es importante)
- La eficiencia del algoritmo es un problema
- Muy dependiente de los avances en técnicas de optimización y regularización.
- Contras: el sobreajuste es siempre una posibilidad

Statistical Learning (statistical goals):

- El enfoque principal es la inferencia (explicación de los datos) utilizando distribuciones de probabilidad
- Buenos resultados solo bajo la hipótesis asumida
- Muy poca atención a problemas de muy gran escala

Data Mining (statistical & computer science):

- El enfoque principal es extraer dependencias entre variables en grandes bases de datos. Esa es una gran inferencia
- Comparte muchas herramientas con ML
- Los algoritmos y hardware con alto nivel de escalabilidad son importantes

Bayesian Learning (probabilistic):

- Un enfoque probabilístico completo basado en distribuciones a priori como conocimiento previo
- El sobreajuste no es un problema en general
- Mucho más complejo matemática y computacionalmente
- Muy poca atención a problemas de algoritmos y computacionales

Learning vs Design: What Learning is not!

- Algunos enfoques solo usan datos para corregir algunos parámetros de un problema bien especificado (**diseño**).
- Ejemplo: supongamos que queremos construir un modelo para reconocer las monedas por su tamaño y masa: $\{(tamaño_i, masa_i), i = 1, \dots, N\}$
- **Diseño:** recopilamos información sobre el tamaño y la masa de cada tipo de moneda y la cantidad de monedas en uso. Construimos un modelo físico para masa y tamaño, teniendo en cuenta las variaciones por el uso y los errores medidos. Finalmente, construimos una distribución de probabilidad en (tamaño, masa) que usamos para clasificar.
- **Aprendizaje:** recopilamos datos etiquetados de cada tipo de moneda. El algoritmo de aprendizaje busca una hipótesis que clasifique bien los datos. Para clasificar una nueva moneda usamos la hipótesis aprendida.
- La información disponible sobre el problema es la clave para adoptar uno u otro enfoque.

Machine Learning definitions

- Arthur Samuel: "el campo de estudio que brinda a las computadoras la capacidad de aprender sin estar programadas explícitamente". Ésta es una definición informal más antigua.
- Tom Mitchell proporciona una definición más moderna: "Se dice que un programa de computadora aprende de la experiencia E con respecto a alguna clase de tareas T y medida de desempeño P, si su desempeño en las tareas de T, medido por P, mejora con la experiencia E."
 - Ejemplo: jugar a las damas
 - E = la experiencia de jugar muchos juegos de damas.
 - T = la tarea de jugar a las damas.
 - P = la probabilidad de que el programa gane el próximo juego.

En general, cualquier problema de aprendizaje automático se puede asignar a una de dos clases amplias:

- Aprendizaje supervisado: aprenda una función a partir de etiquetas y datos de muestra.
- Aprendizaje no supervisado: aprenda una función a partir de datos de muestra.

Three main ML Paradigms

- Aprendizaje supervisado: datos de muestra + profesor (etiqueta) (**aprendizaje estático**).
- Aprendizaje reforzado: datos de muestra + recompensas (no etiqueta) (**aprendizaje dinámico**).
- Aprendizaje no supervisado: solo datos (**representación**).

Aprendizaje supervisado

- **Regresión**: la salida es un número real (variable continua)
 - Predecir la altura de una persona a partir de una muestra de datos:
 - Características: peso; (peso, longitud de los pies); (peso, largo de pies, ancho de hombros), etc.
 - Predecir la temperatura para el día siguiente a partir de un registro anterior de temperaturas
- **Clasificación**: la salida es una etiqueta de clase (variable discreta / categórica)
 - Predecir el tiempo para mañana: (soleado, nublado, ventoso)
 - Predecir si una imagen contiene una cara: (Sí, No), (0,1), (1, -1), etc.
 - Predecir si un correo electrónico es SPAM o no: (Sí, No), (0,1), (1, -1), etc.
- **Clasificación probabilística**: la salida es un vector de probabilidad sobre las etiquetas.
 - Predecir valores reales
 - Resuelve los mismos problemas que la clasificación.

Let's check

A) Supongamos que alimentamos un algoritmo de aprendizaje con una gran cantidad de datos meteorológicos históricos y hacemos que aprenda a predecir el tiempo. En este contexto, ¿qué es T?

1. La probabilidad de que prediga correctamente el clima de una fecha futura.
2. La tarea de predicción del tiempo.
3. Ninguno de esos.
4. El proceso del algoritmo que examina una gran cantidad de datos meteorológicos históricos.

B) Suponga que está trabajando en la predicción del tiempo y utiliza un algoritmo de aprendizaje para predecir la temperatura de mañana (en grados Centígrados / Fahrenheit). ¿Trataría esto como una clasificación o un problema de regresión?

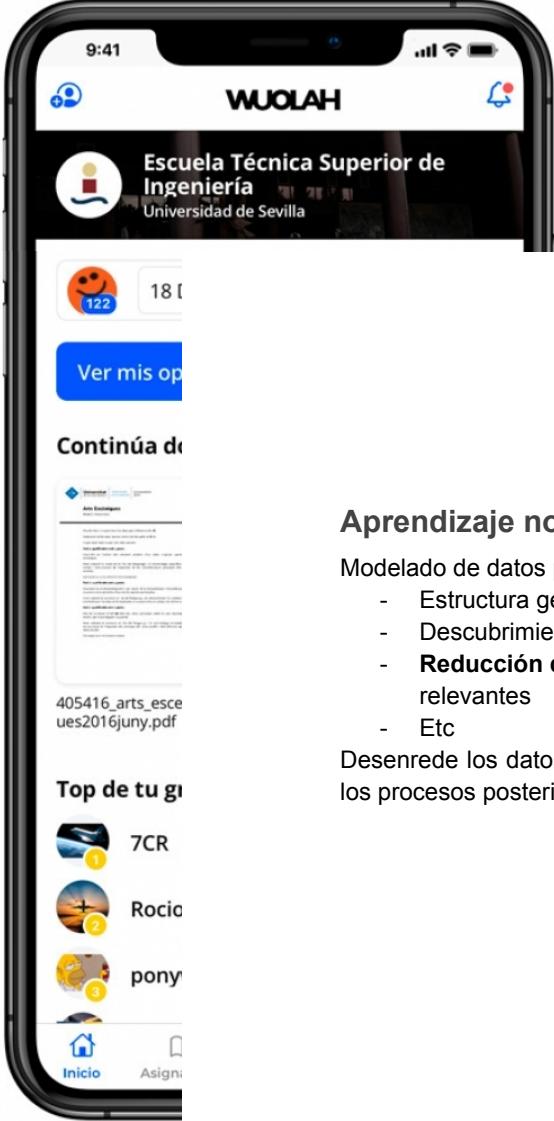
C) Suponga que está trabajando en la predicción del mercado de valores. Le gustaría predecir si una determinada empresa se declarará en quiebra dentro de los próximos 7 días (mediante formación sobre datos de empresas similares que anteriormente habían estado en riesgo de quiebra). ¿Trataría esto como una clasificación o un problema de regresión?

D) Algunos de los problemas siguientes se abordan mejor con un algoritmo de aprendizaje supervisado y otros con un algoritmo de aprendizaje no supervisado. ¿A cuál de los siguientes aplicaría el aprendizaje supervisado? (Seleccione todo lo que corresponda). En cada caso, suponga que hay un conjunto de datos apropiado disponible para su algoritmo para aprender.

- 1) Tome una colección de 1000 ensayos escritos sobre la economía de los EE. UU. Y encuentre una manera de agrupar automáticamente estos ensayos en una pequeña cantidad de grupos de ensayos que son de alguna manera "similares" o "relacionados".
- 2) Dados los datos históricos de las edades y alturas de los niños, prediga la estatura de los niños en función de su edad.
- 3) Examine una gran colección de correos electrónicos que se sabe que son correos electrónicos no deseados, para descubrir si hay subtipos de correo no deseado.
- 4) Dados 50 artículos escritos por autores masculinos y 50 artículos escritos por autoras mujeres, aprenda a predecir el género del autor de un nuevo manuscrito (cuando el se desconoce la identidad de este autor).

E) ¿Cuál de las siguientes es una definición razonable de aprendizaje automático?

1. El aprendizaje automático es el campo de estudio que brinda a las computadoras la capacidad de aprender sin estar programadas explícitamente.
2. El aprendizaje automático aprende de los datos etiquetados.
3. El aprendizaje automático es la ciencia de programar computadoras.
4. El aprendizaje automático es el campo que permite a los robots actuar de manera inteligente.



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.

Available on the
App Store

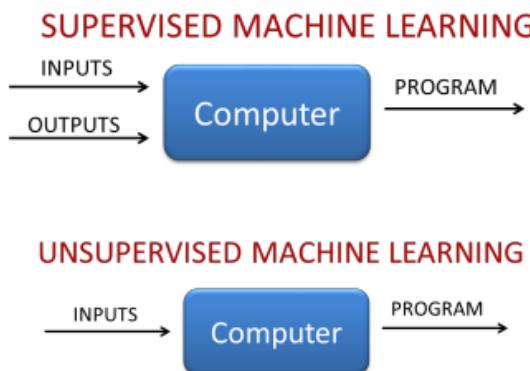
GET IT ON
Google Play

Aprendizaje no supervisado

Modelado de datos para descubrir qué hay dentro:

- Estructura geométrica: **agrupamiento (clustering)**
- Descubrimiento de dependencias: **patrones (patterns)**
- **Reducción de dimensionalidad (Dimensionality reduction)**: características relevantes
- Etc

Desenrede los datos muestreados de forma más independiente características que facilitan los procesos posteriores.



"Se dice que un programa de computadora aprende de la experiencia E con respecto a alguna clase de tareas T y medida de desempeño P, si su desempeño en las tareas de T, medido por P, mejora con la experiencia E."

Formalizando el enfoque

1. ¿Cuál es la información disponible?

- ¿De dónde proceden los datos? : **P (D)**
- ¿Qué funciones utilizar? **X**
- ¿Alguna condición de muestreo?: **independiente distribuida de forma idéntica (i.i.d.)**
independent identically distributed

2. ¿Cuál es la tarea de predicción? **f: X → Y (etiquetas)**

3. ¿Cómo configurar un modelo?

- ¿Qué representación utiliza ?:
 - ¿Qué clase de función vamos a utilizar? **H**
 - ¿Cómo caracterizar cada elemento **h ∈ H**? **parámetros**

4. ¿Cómo buscar dentro de **H**?

- ¿Qué criterios optimizar para garantizar el aprendizaje? **ERM, SRM, MDL**
- La función para optimizar: (función de pérdida)
- ¿Qué algoritmo usar para encontrar la mejor función de **H**? **A**

Elementos principales de una tarea de aprendizaje

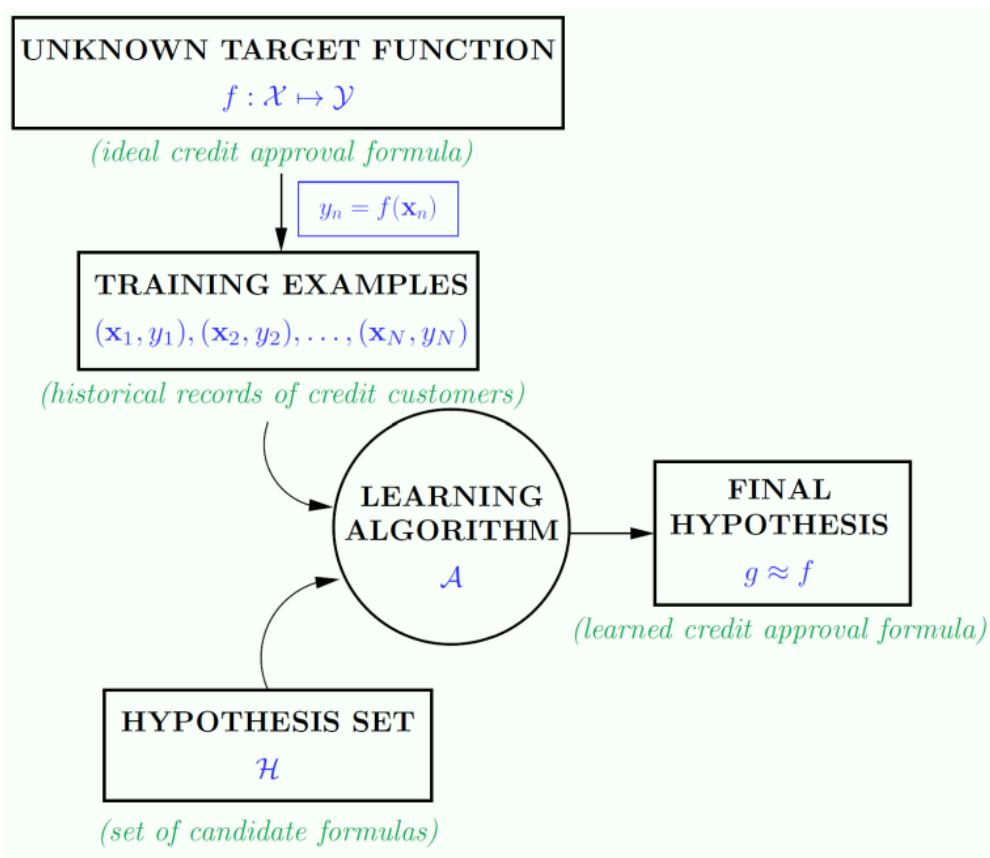
- **Entrada:** vector de características
- **Salida:** clase o etiqueta
- **Función de destino:** desconocida
- **Muestra de datos:** i.i.d \mathbf{x}_i
- **Muestra de entrenamiento:** datos etiquetados

Conjunto de datos significa muestra de entrenamiento A

Aprendizaje en acción

- Supongamos que X , Y y D vienen dados por la tarea de aprendizaje.
 - Comience con un conjunto de hipótesis candidatas H que cree que probablemente representen f
$$H = \{h_1, h_2, \dots, h_n\}$$
 se llama conjunto de hipótesis o modelo.
 - Seleccione una hipótesis g de H . La forma en que hacemos esto se llama **algoritmo de aprendizaje**.
 - Utilice g para nuevas muestras. Esperamos $g \approx f$.
- El objetivo f es fijo pero desconocido.**
- Elegimos H y el algoritmo de aprendizaje A

Diagrama de la configuración de aprendizaje inicial



Sesión 2

Modelos lineales: configuración general

- H es la clase de todos los hiperplanos (funciones lineales de predictores/características).
- Esto es, $h(\mathbf{x}) = w_0 + w_1x_1 + \dots + w_dx_d = w_0 + \sum_{i=1}^d x_i w_i$
 - Equivalentemente usando $\mathbf{x}^T = (1, x_1, x_2, \dots, x_d)$, $\mathbf{w}^T = (w_0, w_1, w_2, \dots, w_d)$
 - Podemos escribir: $h(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$
- Se pueden abordar tres problemas diferentes utilizando esta clase de funciones:
 - Regresión
 - Clasificación
 - Estimación de probabilidades
- Una regla general, cuando se enfrentan a problemas de aprendizaje: **generalmente es una estrategia ganadora probar primero un modelo lineal**

Regresión

Elementos del problema del crédito

El conjunto de datos: $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$

y_n es la línea de crédito para el cliente x_n (vector de características)

Hemos asumido $h(\mathbf{x}) = w_0 + w_1x_1 + \dots + w_dx_d = \mathbf{w}^T \mathbf{x}$ -> salida de la regresión lineal (**predicción del error**)

Para cada w dado obtenemos los valores $h_w(x_1), \dots, h_w(x_n)$

Para saber qué tan bueno es h_w con respecto a la verdadera f , la regresión usa un error cuadrático $(h_w - f)^2$

Entonces el error de muestra es:

$$E_{in}(h_w) = \frac{1}{N} \sum_{n=1}^N (h_w(x_n) - y_n)^2$$

¿Cómo continuar con esto?

Regresión lineal

- Ahora $X = \{1\} \times \mathbb{R}^d$, $y = \mathbb{R}$ y $h: X \rightarrow y$
 $E_{out}(h) = E(x, y) \sim p[(h(x) - y)^2]$
- Queremos encontrar \hat{h} tal que, $\hat{h} = \min_{h \in H} E_{out}(h)$
- **Ahora H toma la forma: $h(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$**
- Para encontrar \hat{h} **minimizamos el Riesgo Empírico (ERM)**

$$E_{in}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^N (\mathbf{w}^T \mathbf{x}_n - y_n)^2$$

$$\mathbf{w}_{lin} = \min_{\mathbf{w} \in \mathbb{R}^{d+1}} E_{in}(\mathbf{w})$$

Pregunta relevante: ¿Por qué el mínimo de E_{in} garantiza un mínimo de E_{out} ?

¿Cuándo es óptima la pérdida cuadrática?

- El Modelo de Mínimos Cuadrados Ordinarios (MCO) (**Ordinary Least Square Model - OLS**) sólo asume un error en la variable dependiente

- Esta hipótesis no es válida en todos los casos
- Pero es una buena aproximación en muchos casos

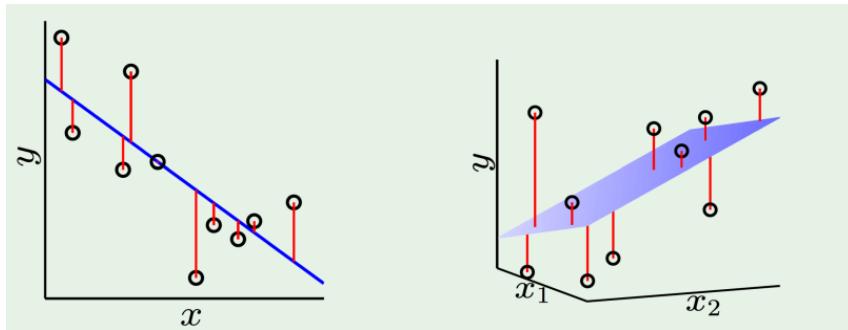
- Modelo: $y_i = f(x_i, \beta) + \text{ruido}$, f lineal en β

- **Teorema de Gauss-Markov:** Bajo el supuesto de ruido incorrecto, media cero y varianza constante, la técnica OLS alcanza el estimador insesgado de varianza mínima para β

Ecuaciones:

- $y_i = \mathbf{x}_i^T \beta + \epsilon_i$,
- $E(\epsilon_i) = 0$, $E(\epsilon_i, \epsilon_j) = 0$, $\text{Var}(\epsilon_i^2) = \sigma^2 < \infty$
- $\hat{\beta}$ - OLS da el estimador de varianza mínima tal que $E(\hat{\beta}_i) = \beta_i$ (imparcial)

Ilustración de Regresión Lineal



Expresión matricial para E_{in}

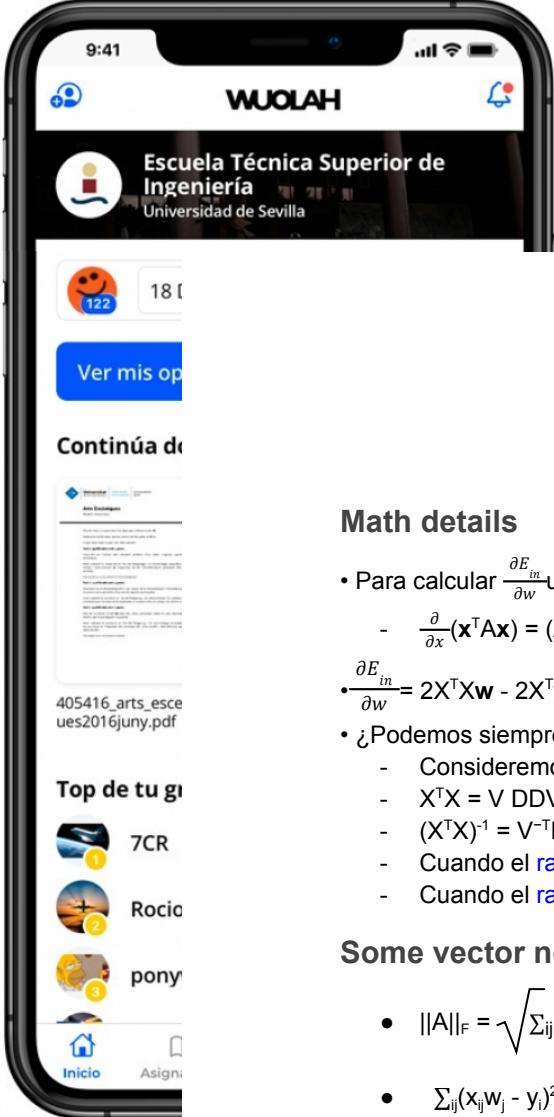
$$E_{in}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^N (\mathbf{w}^T \mathbf{x}_n - y_n)^2 = \frac{1}{N} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2$$

Donde:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

Minimización E_{in} : ecuaciones normales

- $E_{in}(\mathbf{w}) = \frac{1}{N} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2$
- $\nabla E_{in}(\mathbf{w}) = \frac{2}{N} \mathbf{X}^T (\mathbf{X}\mathbf{w} - \mathbf{y}) = \mathbf{0}$
↳ $\nabla E_{in}(\mathbf{w})$ derivada de $E_{in}(\mathbf{w})$
- $\mathbf{X}^T \mathbf{X}\mathbf{w} = \mathbf{X}^T \mathbf{y}$
- $\mathbf{w} = \mathbf{X}^{-1} \mathbf{y}$ donde $\mathbf{X}^{-1} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$ es la **pseudo-inversa** de \mathbf{X}



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.

Available on the App Store

GET IT ON Google Play

Math details

- Para calcular $\frac{\partial E_{in}}{\partial w}$ utilizamos dos propiedades bien conocidas:
 - $\frac{\partial}{\partial x}(\mathbf{x}^T A \mathbf{x}) = (A^T + A)\mathbf{x}$, $\frac{\partial}{\partial x}(\mathbf{x}^T A) = A$
 - $\frac{\partial E_{in}}{\partial w} = 2X^T X w - 2X^T y = \mathbf{0} \rightarrow w_{lin} = (X^T X)^{-1} X^T y$
- ¿Podemos siempre calcular $(X^T X)^{-1}$?
 - Consideraremos la descomposición de valores singulares (SVD): $X = UDV^T$
 - $X^T X = V D D V^T$
 - $(X^T X)^{-1} = V^{-T} D^{-1} V^{-1} = V D^{-1} V^T$ este producto siempre se puede calcular
 - Cuando el rango $(X^T X) = d$, solo existe una solución.
 - Cuando el rango $(X^T X) < d$, existen infinitas soluciones.

Some vector notation

- $\|A\|_F = \sqrt{\sum_{ij} a_{ij}}$
- $\sum_{ij} (x_{ij}w_j - y_i)^2 = \sum_i (x_i^T w - y_i)^2 = \|Xw - y\|^2$
- $\|Xw - y\|^2 = (Xw - y)^T (Xw - y) = (Xw)^T Xw - y^T Xw - (Xw)^T y + y^T y = w^T X^T X w - 2y^T Xw + y^T y$

Algoritmo de Regresión Lineal

- Construye la matriz X y el vector y del conjunto de datos $(x_1, y_1), \dots, (x_N, y_N)$ como sigue:

$$X = \underbrace{\begin{bmatrix} -x_1^T - \\ -x_2^T - \\ \vdots \\ -x_N^T - \end{bmatrix}}_{\text{input data matrix}}, \quad y = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}}_{\text{target vector}}$$

- Calcula la pseudo-inversa $X^T = (X^T X)^{-1} X^T$

- Devuelve $w = X^T y$

Propiedades de Regresión Lineal

- Hat-Matrix (H):** Sea $X(N \times (d+1))$ las muestras

$$\hat{y} = Xw_{lin} = X(X^T X)^{-1} X^T y = Hy$$
- El w_{lin} es un intento de mapear la entrada X con la salida y , pero

$$H = X(X^T X)^{-1} X^T$$
 mapea de y a \hat{y}
- Matriz de protección H :
 - Las propiedades de H son relevantes en el análisis de E_{out} y E_{in}
 - La matriz de sombrero (hat-matrix) es **idempotente** $\rightarrow H^2 = H$
 - La traza (H) = $d+1$ (vector de entrada dimensión+1)

- **Error de generalización:** Para la regresión lineal se pueden encontrar fórmulas exactas para E_{out} y E_{in} :
$$E_{\text{out}}(\mathbf{w}_{\text{lin}}) = E_{\text{in}}(\mathbf{w}_{\text{lin}}) + O(\frac{d}{N})$$

Alternativa: Descenso del Gradiente

- Dado \mathbf{w}_0 queremos encontrar \mathbf{V} como $E_{in}(\mathbf{w}_0 + \eta \mathbf{V}) < E_{in}(\mathbf{w}_0)$
- ¿Como decrementa $E_{in}(\mathbf{w}_0)$? Aplicar la expansión de Taylor al primer orden, con $\|\mathbf{V}\| = 1$

$$\begin{aligned}\Delta E_{in} &= E_{in}(\mathbf{w}(0)) + \eta \mathbf{V} - E_{in}(\mathbf{w}(0)) = \\ &= \eta \nabla E_{in}(\mathbf{w}(0))^T \mathbf{V} + O(\eta^2) \\ &\geq -\eta \|\nabla E_{in}(\mathbf{w}(0))\|\end{aligned}$$

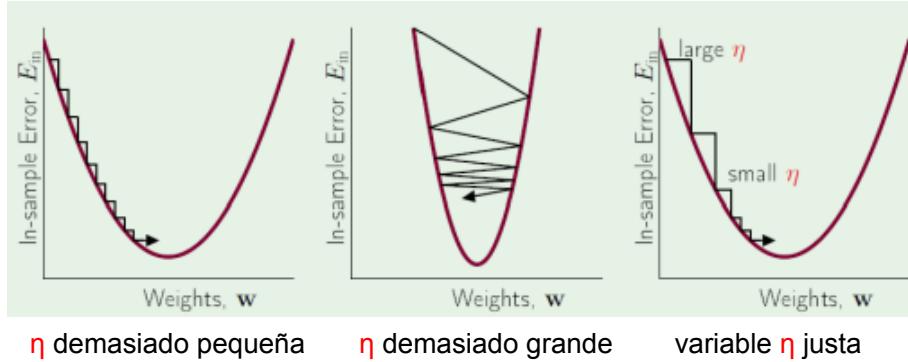
La igualdad se cumple si y sólo si $\mathbf{V} = -\frac{\nabla E_{in}(\mathbf{w}(0))}{\|\nabla E_{in}(\mathbf{w}(0))\|}$ (gradiente negativo)

- Gradient Descent (GD):

- El Descenso de Gradiente es una técnica de optimización iterativa general que alcanza un óptimo local siguiendo la dirección del vector de gradiente en cada punto.

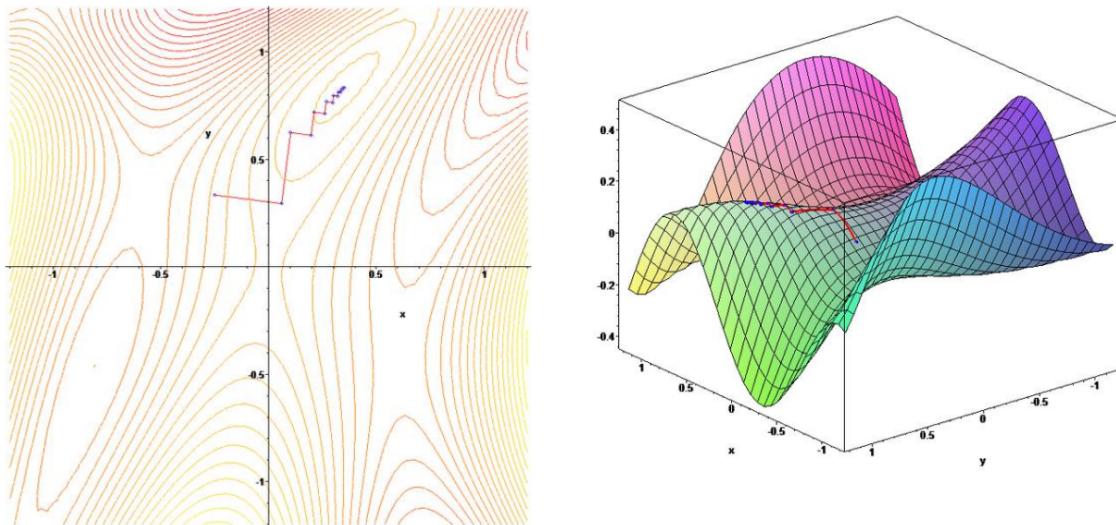
¿Cómo fijar η ? (learning rate, tasa de aprendizaje)

Cómo η afecta al algoritmo:



- η debería incrementar con la pendiente

¿Qué óptimo logramos?



Estimación de parámetros: Descenso de Gradiente

Inicia en algún valor inicial de \mathbf{w}_0 y repetidamente realiza la actualización,

$$\mathbf{w}_j := \mathbf{w}_j - \eta \frac{\partial E_{in}(\mathbf{w})}{\partial w_j} \quad (\text{ECUACIÓN GENERAL})$$

(Esta actualización se realiza simultáneamente para todos los valor de $j = 0, \dots, N$). Aquí, η es llamada tasa de aprendizaje (learning rate).

$$\frac{\partial E_{in}(\mathbf{w})}{\partial w_j} = \frac{\partial}{\partial w_j} \frac{1}{N} \sum_{n=1}^N (\mathbf{w}^\top \mathbf{x}_n - y_n)^2 = \frac{2}{N} \sum_{n=1}^N x_{nj} (\mathbf{w}^\top \mathbf{x}_n - y_n) = \frac{2}{N} \sum_{n=1}^N x_{nj} (\mathbf{h}(\mathbf{x}_n) - y_n)$$

Cada punto (\mathbf{x}_n, y_n) contribuye a la actualización en una cantidad proporcional a su error de predicción.

En este caso todos los puntos son usados para calcular el gradiente: **BATCH GRADIENT DESCENT** (Descenso de Gradiente por Lotes)

Stochastic Gradient Descent: SGD (Descenso de Gradiente Estocástico)

- Una alternativa es usar una **estimación estocástica** seleccionando una parte de la muestra para calcular el gradiente, $M < N$ (**SGD**)

$$\frac{\partial E_{in}(\mathbf{w})}{\partial w_j} = \frac{2}{N} \sum_{n=1}^N x_{nj} (\mathbf{h}(\mathbf{x}_n) - y_n)$$

- Mayor viabilidad en la estimación del gradiente (menos ejemplos en la media)
- Muy rápido de calcular
- En funciones no convexas, **evidencia empírica de tener un buen mínimo local**

- Aunque se puede usar un único artículo en cada iteración, la regla aceptada es un **minilote** de artículos (tamaño: 32-128).

Un Algoritmo de Regresión Iterativo

Descenso de Gradiente por Lotes (Batch Gradient Descend)

Dado un conjunto de datos $(\mathbf{x}_n, y_n), n = 1, 2, \dots, N$

1. Fijar $\mathbf{w}=0$, $\eta=\eta_0$
2. Iterar
 - for $j=0, \dots, K$:
$$\mathbf{w}_j := \mathbf{w}_j - \eta \sum_{n=1}^N x_{nj} (\mathbf{h}(\mathbf{x}_n) - y_n)$$
3. Hasta $E_{in}(\mathbf{w}) < \text{epsilon}$

Descenso de Gradiente Estocástico (Stochastic Gradient Descend)

1. Fijar $\mathbf{w}=0$, $\eta=\eta_0$
2. Fijar una secuencia de minilotes
3. Iterar en minilotes
 - for $j=0, \dots, K$:

$$\mathbf{w}_j := \mathbf{w}_j - \eta \sum_{n \in \text{Minilote}} x_{nj} (\mathbf{h}(\mathbf{x}_n) - y_n)$$

4. Hasta $E_{in}(\mathbf{w}) < \text{epsilon}$

Estudiar sin publi es posible.

Compra Wuolah Coins y que nada te distraiga durante el estudio.



Método de Newton: Una cura para la oscilación

- Una nueva regla de actualización para \mathbf{w} basada en las **derivadas de segundo orden** (Hessian)

$$f(x + \Delta x, y + \Delta y) \approx f(x, y) + \frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial y} \Delta y + \frac{1}{2} \left(\frac{\partial^2 f}{\partial x^2} \Delta x^2 + \frac{\partial^2 f}{\partial y^2} \Delta y^2 + \frac{\partial^2 f}{\partial x \partial y} \Delta x \Delta y + \frac{\partial^2 f}{\partial y \partial x} \Delta x \Delta y \right)$$

- Acerquémonos a $E_{in}(\mathbf{w})$ hasta el segundo orden.

$$g(\Delta \mathbf{w}) = E_{in}(\mathbf{w}_0 + \Delta \mathbf{w}) \approx E_{in}(\mathbf{w}_0) + \Delta \mathbf{w}^\top \nabla E_{in}(\mathbf{w}_0) + \frac{1}{2} \Delta \mathbf{w}^\top \nabla^2 E_{in}(\mathbf{w}_0) \Delta \mathbf{w}$$

$$\frac{\partial g(\Delta \mathbf{w})}{\partial \Delta \mathbf{w}} = 0 \rightarrow \nabla E_{in}(\mathbf{w}_0) + \nabla^2 E_{in}(\mathbf{w}_0) \Delta \mathbf{w} = 0$$

$$\hookrightarrow \mathbf{H} = \nabla^2 E_{in}(\mathbf{w}_0)$$

$$\Delta \mathbf{w} = -\mathbf{H}^{-1} \nabla E_{in}(\mathbf{w}_0)$$

- Ahora la matriz \mathbf{H} define el avance en la dirección del gradiente.
- La mayor complejidad es invertir \mathbf{H} .

En la práctica

- Analizando los datos:
 - La regresión 1D permite el análisis de residuos para detectar muestras fuera de la hipótesis OLS
- Analizando los residuales con gráficos (pequeñas muestras):
 - Errores correlacionados
 - Errores con varianza no constantes
 - Dependencia entre predictores
 - Valores atípicos: (relevante en pequeñas muestras)
- En dimensiones superiores o muestras de datos muy grandes, el escenario es más complejo y la hipótesis original pierde relevancia.
- Predictores categóricos
- Valores perdidos

Características categóricas y valores perdidos

- **Codificación de características categóricas:**
 - Para modelos lineales, una característica categórica se codifica como un vector one-hot: [0,0,0, ..., 1, ..., 0,0] cuya longitud es el número de valores categóricos
 - Algunos modelos no lineales pueden gestionar características categóricas: Árboles
- **Valores perdidos:**
 - Los modelos lineales no pueden gestionar los valores perdidos.
 - Valores sustitutos:
 - Cada valor se puede interpolar a partir de su histograma de características.
 - Cada valor se puede predecir usando un modelo específico
 - Algunos modelos no lineales pueden gestionar esta situación: árboles



WUOLAH

Clasificación

Representación de entrada

Datos 'crudos' \mathbf{x} : $(x_0, x_1, \dots, x_{256})$

Modelo lineal: $(w_0, w_1, \dots, w_{256})$

Características: Extraer información útil

Intensidad y simetría $\mathbf{x} = (x_0, x_1, x_2)$

Modelo lineal: (w_0, w_1, w_2)

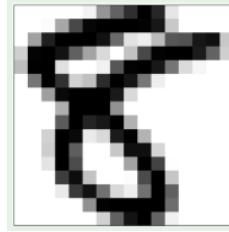
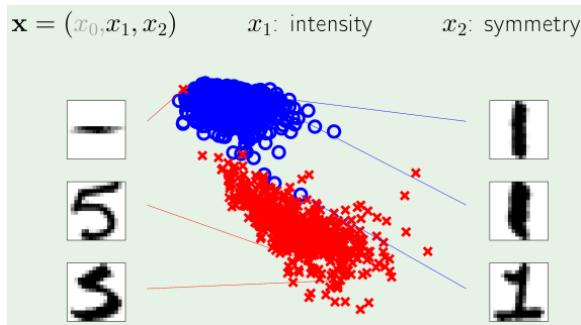
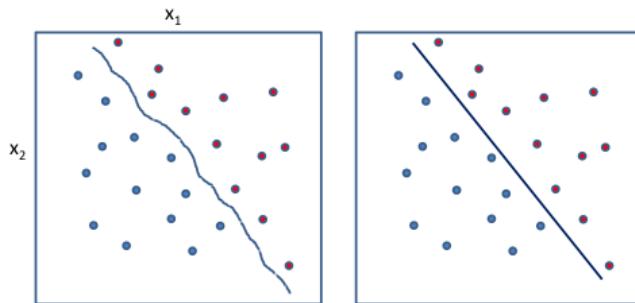


Ilustración de características



Un caso muy simple



- ¿Cómo encontrar una función de separación?
- Supongamos una clase de funciones H muy simple (líneas en 2D, hiperplanos en general)
- Supongamos un algoritmo A muy simple para buscar la mejor función

Perceptrón: dos clases

Regla de asignación (problema del crédito):

Si $\sum_{i=1}^d x_i w_i \geq b$ (Umbral) entonces **Aprueba** crédito

Si $\sum_{i=1}^d x_i w_i < b$ (Umbral) entonces **Rechaza** crédito

Entonces, $h(\mathbf{x}) = \text{sign}((\sum_{i=1}^d x_i w_i) + b)$, $h(\mathbf{x}) \in \{-1, 1\}$

- Usando equivalentemente $\mathbf{x}^\top = (1, x_1, x_2, \dots, x_d)$, $\mathbf{w}^\top = (b, w_1, w_2, \dots, w_d)$
- Podemos escribir: $H \{h \mid h(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x}), \text{ para } \mathbf{w} \text{ fijada}\}$

Perceptrón o Separador Lineal

Esta nueva clase H es llamada Perceptrón o Separador Lineal

$$\text{Criterio de aprendizaje (ERM): } (\min_{w \in \mathbb{R}^{d+1}} \frac{1}{N} \sum_{n=1}^N [\text{sign}(w^T x_n) \neq y_n]) = E_{in}$$

¿Cómo minimizarlo?

Ahora aparece una nueva función de Error/Pérdida $[\text{sign}(w^T x_n) \neq y_n]$

ERM es una función discreta no derivable ... diferente de la regresión

Analizamos un detalle más

La función error es:

$$\text{error}(w^T x_n, y_n) = [\text{sign}(w^T x_n) \neq y_n] = \begin{cases} -y_n w^T x_n & \text{si } \text{sign}(w^T x_n) \neq y_n \\ 0 & \text{si } \text{sign}(w^T x_n) = y_n \end{cases}$$

Entonces se puede deducir un nuevo y derivable error de:

$$\text{error}(w^T x_n, y_n) = \begin{cases} -y_n w^T x_n & \text{si } \text{sign}(w^T x_n) \neq y_n \\ 0 & \text{si } \text{sign}(w^T x_n) = y_n \end{cases}$$

Equivalentemente, $\text{error}(w^T x_n, y_n) = \max(0, -y_n w^T x_n)$

Algoritmo de aprendizaje del Perceptrón (PLA)

SGD con tamaño de lote = 1 y grado de aprendizaje $\lambda=1$

Regla de adaptación:

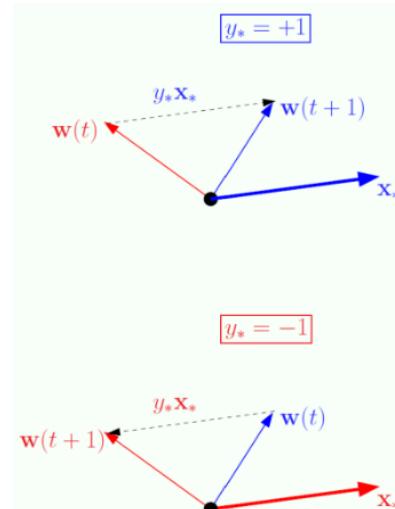
$$\begin{aligned} w_{\text{actualizado}} &= w_{\text{actual}} + y^* x & \text{si } \text{sign}(w^T x_n) \neq y_n \\ w_{\text{actualizado}} &= w_{\text{actual}} & \text{si } \text{sign}(w^T x_n) = y_n \end{aligned}$$

- El algoritmo de aprendizaje itera sobre los datos de muestra pero sin memoria

- Algoritmo simple que busca lo mejor, $h \in H$, a través de evaluaciones iterativas de los datos de entrenamiento (D):

- Para conjuntos de datos linealmente separables:

- El algoritmo PLA siempre puede encontrar un vector \hat{w} tal que $h(x_i) = y_i$ para todas las muestras de entrenamiento (D) en tiempo finito.
- ¡PLA aprende una regla óptima usando sólo datos!!



Algoritmo de Clasificación Lineal

- Perceptrón Lineal:

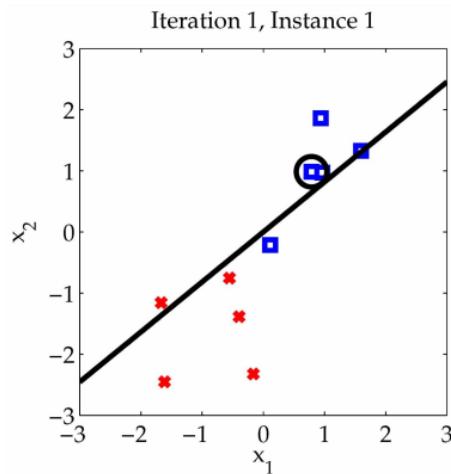
- Dado un conjunto de datos $(x_n, y_n), n = 1, 2, \dots, N$
- Paso 1: Fijar $w_{ini} = 0$
- Paso 2: Iterar en la muestra D mejorando la solución
- Repite

```

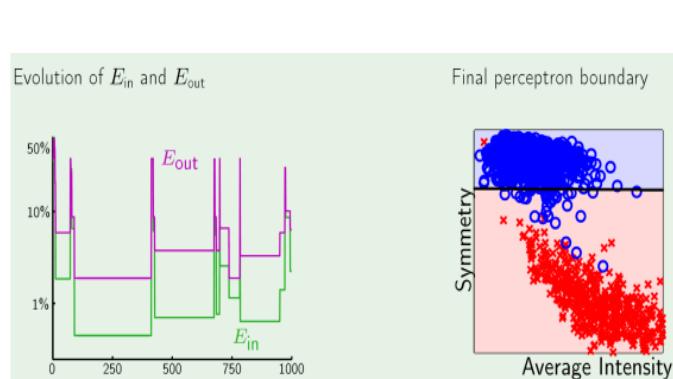
    For each  $x_i \in D$  do
        if:  $\text{sign}(w^T x_i) \neq y_i$  then
            update  $w$ :  $w_{new} = w_{old} + y_i x_i$ 
        else continue
    End for
  
```

- Hasta que no haya cambios en un pase completo de D

Cómo trabaja el Perceptrón



Cómo trabaja PLA



Modelo de Clasificación Lineal

- Supongamos ahora $X=\{1\} \times \mathbb{R}^d$, $y = \{-1,1\}$ y $f:X \rightarrow y$
- El modelo Lineal para clasificación binaria usa la hipótesis de clase H ,
$$h(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x})$$

donde $\mathbf{w} \in \mathbb{R}^{d+1}$, siendo d la dimensionalidad del espacio de entrada y $x_0=1$

- Dos principales cuestiones a responder:
 - ¿Podemos asegurarnos de que $E_{out}(g)-E_{in}(g) \approx 0$?
 - ¿Podemos hacer $E_{in}(g) \approx 0$?
- Según la teoría de aprendizaje (ERM)

$$E_{out}(g) \leq E_{in}(g) + O(\sqrt{d \frac{\log(N)}{N}})$$

Como podemos ver, el error de generalización es comparable a la de la regresión lineal $O(\frac{d}{N})$

Entonces para una N lo suficientemente grande E_{ou} y E_{in} estarán muy cerca uno del otro

Clasificación Lineal: $E_{in}(g) \approx 0$?

Consideramos dos escenarios:

- Datos linealmente separables: lo que significa que existen \mathbf{w}^* con $E_{in}(\mathbf{w}^*) = 0$
 - Datos no separables
- Escenario 1:
 - Resultado PLA:** Sea $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)$ una muestra separable de ejemplos. Sea $B = \min \{ \|\mathbf{w}\| : \forall i \in [m], y_i \mathbf{w}^T \mathbf{x}_i \geq 1, \mathbf{w} \in \mathbb{R}^d \}$ y sea $R = \max_i \|\mathbf{x}_i\|$. Luego, el PLA se detiene después de como máximo $(RB)^2$ iteraciones, y cuando se detiene se sostiene que $\forall i \in [m], y_i \mathbf{w}^T(t) \mathbf{x}_i \geq 0$.
 - La tasa de convergencia depende del parámetro B . Pero este valor puede ser exponencialmente grande en d .
 - En lo que respecta a PLA, la separabilidad lineal es una propiedad de los datos D , no del objetivo f .



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.

Available on the
App Store

GET IT ON
Google Play



122

18

Ver mis op

Continúa d



405416_arts_esceues2016juny.pdf

Top de tu gu



7CR



Rocio



pony



Inicio

Asign

- Una D lineal separable podría haberse generado a partir de un objetivo lineal separable o (por casualidad) de un objetivo no separable

- Escenario 2: Dos escenarios diferentes

- 2.1 Errores en una solución lineal separable

- 2.2 No existe una solución lineal separable

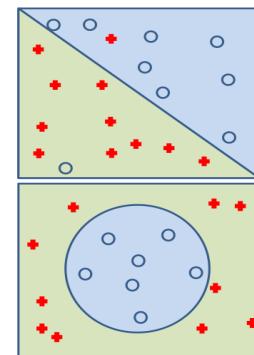
Para encontrar una solución en el 2.1 tenemos que resolver

$$(\min_{w \in R^{d+1}} \frac{1}{N} \sum_{n=1}^N [\text{sign}(w^T x_n) \neq y_n]) = E_{in}$$

Pero se sabe que este problema es un problema combinatorio NP-difícil en el caso general.

Aproximaremos el problema con el Algoritmo de Bolsillo (**Pocket Algorithm**)

2.2 se aborda utilizando transformaciones no lineales (más adelante)



- El algoritmo de bolsillo: esencialmente el algoritmo de bolsillo mantiene "en su bolsillo" el mejor vector solución encontrado en la iteración t en PLA

- ALGORITMO DE BOLSILLO:

1. Establecer el vector de peso de bolsillo \hat{w} en $w(0)$ de PLA

2. For $t = 1, \dots, T$ do

3. Ejecutar PLA para una actualización para obtener $w(t+1)$

4. Evaluar $E_{in}(w(t+1))$

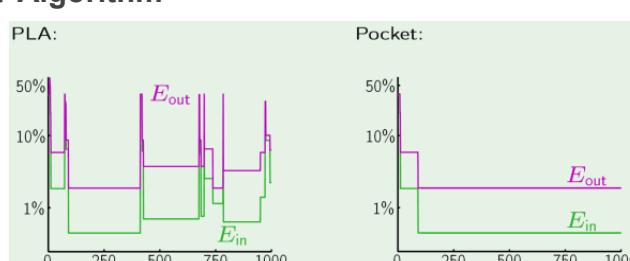
5. Si $w(t+1)$ es mejor que \hat{w} en términos de $E_{in}(w(t+1))$, establezca $\hat{w} = w(t+1)$

6. Return \hat{w}

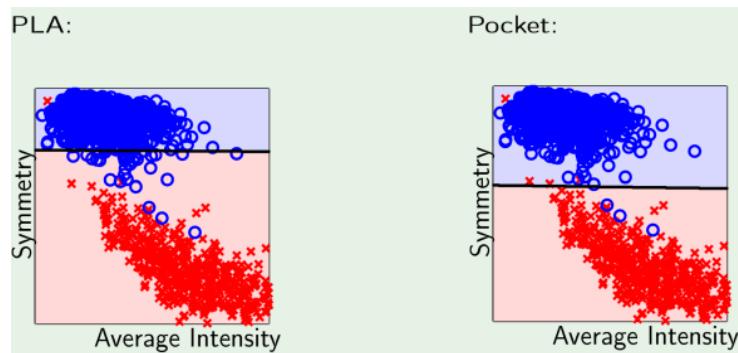
- El algoritmo de bolsillo tiene una clara penalización de eficiencia en el punto 4.

- Pero se garantiza que obtendrá una buena solución después de un gran número fijo de actualizaciones.

The 'Pocket' Algorithm



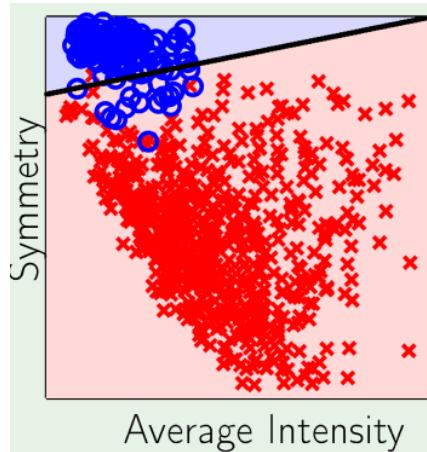
Límite de Clasificación (Classification Boundary) - PLA vs Pocket



Regresión Lineal para clasificación

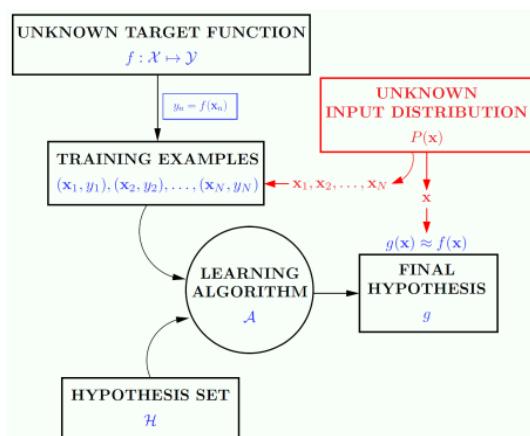
- La regresión lineal aprende una función real $y = f(\mathbf{x}) \in \mathbb{R}$
- ¡Las funciones con valores binarios también tienen valor real! $\pm 1 \in \mathbb{R}$
- Utilice la regresión lineal para obtener \mathbf{w} donde $\mathbf{w}^T \mathbf{x}_n \approx y_n = \pm 1$
- En este caso, es probable que el $\text{sign}(\mathbf{w}^T \mathbf{x}_n)$ esté de acuerdo con $y_n = \pm 1$
- ¡Buenos pesos iniciales para la clasificación!

Límite de Regresión Lineal



Etiquetas ruidosas: Una configuración general

Configuración de aprendizaje actual



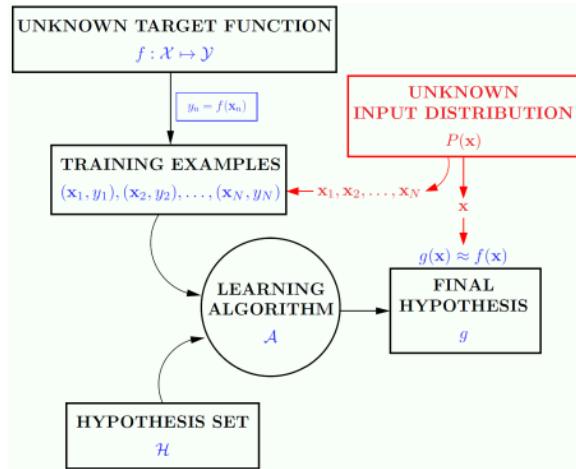
Muestras ruidosas

- Un escenario realista debe considerar ETIQUETAS ruidosas, esto es: $y_n \neq f(\mathbf{x}_n)$
 - En general tenemos **objetivos ruidosos** por falta de información sobre las variables que definen la función desconocida
 - En otras palabras, **la función objetivo no siempre es una función determinista.**
- Caso de aprobación de crédito:
 - Dos clientes con datos idénticos pueden tener una respuesta diferente
- Mangos sabrosos:
 - Los artículos con las mismas características se pueden etiquetar como de diferente clase.

Distribución objetivo

- En lugar de $y = f(x)$ (determinista) ahora usamos: $P(y|x)$
- Cada muestra (x,y) se genera a partir de la distribución conjunta: $P(x)P(y|x)$
- El **objetivo ruidoso** es un objetivo determinista $f(x) = E(y|x)$ más ruido $y - f(x)$
- El **objetivo determinista** es un caso especial de objetivo ruidoso
 $P(y|x)$ es cero excepto en $y=f(x)$

Configuración de aprendizaje actualizado



ESTIMACIÓN DE PROBABILIDAD

Regresión Logística (LGR)

- Ahora la función de etiquetado desconocida es: $f: X \rightarrow [0,1]$ tal que $f(x) = P(y=1|x)$
- Pero los valores que nos dan son deterministas (± 1)-etiquetas de las clases
- Como las etiquetas- (± 1) están altamente correlacionadas con la función f , elegimos como $h(x) = \sigma(w^T x)$ el w -vector que maximiza la probabilidad del simple $s = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$,

dado por: $L(s) = \prod_{i=1}^N P(y_i|\mathbf{x}_i)$

es decir, tratamos de encontrar una función h dada la misma (± 1)-etiquetas en los puntos muestrales \mathbf{x}_i

- Regresión lineal: $h(x) = w^T x$,
- Regresión logística: $h(x) = \sigma(w^T x) \in [0,1]$ o también $[-1,1]$
- La función σ se llama **función logística (sigmoide)**: \rightarrow

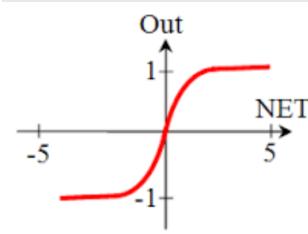
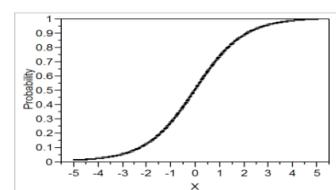
$$\sigma(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x + 1}$$

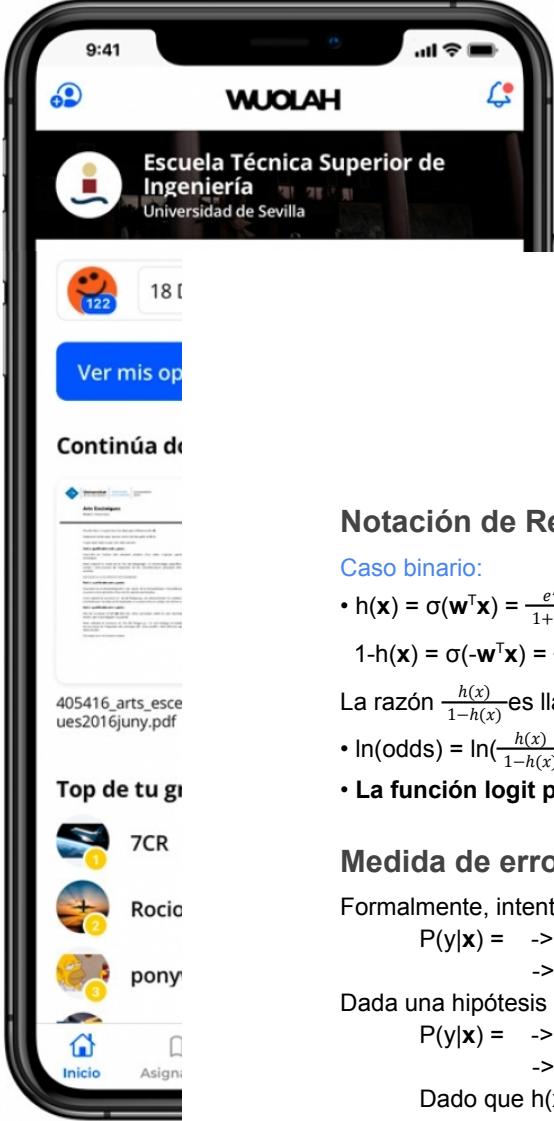
- Otra función:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

- Aparecen algunas ventajas:

- La salida LGR se puede considerar como una probabilidad de clasificación.
- Nos permite cierta flexibilidad para asignar muestras a las etiquetas.





Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.

Available on the
App Store

GET IT ON
Google Play

Notación de Regresión Logística

Caso binario:

$$\bullet h(\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x}) = \frac{e^{(\mathbf{w}^T \mathbf{x})}}{1+e^{(\mathbf{w}^T \mathbf{x})}} \quad \text{Probabilidad C1}$$

$$1-h(\mathbf{x}) = \sigma(-\mathbf{w}^T \mathbf{x}) = \frac{1}{1+e^{(\mathbf{w}^T \mathbf{x})}} \quad \text{Probabilidad C2}$$

La razón $\frac{h(x)}{1-h(x)}$ es llamado "probabilidades" ("odds")

$$\bullet \ln(\text{odds}) = \ln\left(\frac{h(x)}{1-h(x)}\right) = \ln(e^{\mathbf{w}^T \mathbf{x}}) = \mathbf{w}^T \mathbf{x} \quad (\text{función logit})$$

• La función logit puede ser cualquier modelo de regresión

Medida de error (pérdida)

Formalmente, intentemos aprender la función de destino $f(\mathbf{x}) = \mathbb{P}[y = +1|\mathbf{x}]$

$$\begin{aligned} P(y|\mathbf{x}) &= \rightarrow f(\mathbf{x}) && \text{para } y = +1 \\ &= \rightarrow 1-f(\mathbf{x}) && \text{para } y = -1 \end{aligned}$$

Dada una hipótesis h , ¿qué tan cerca está de f en términos del objetivo ruidoso?

$$\begin{aligned} P(y|\mathbf{x}) &= \rightarrow h(\mathbf{x}) && \text{para } y = +1 \\ &= \rightarrow 1-h(\mathbf{x}) && \text{para } y = -1 \end{aligned}$$

Dado que $h(\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x})$ entonces $P(y|\mathbf{x}) = \sigma(y\mathbf{w}^T \mathbf{x})$

Dado que los puntos muestrales son independientes, la probabilidad de la muestra es

$$L(\mathbf{w}) = \prod_{i=1}^N P(y_i|\mathbf{x}_i) = \prod_{i=1}^N \sigma(y_i \mathbf{w}^T \mathbf{x}_i) = \prod_{j=1}^{N_1} \sigma(\mathbf{w}^T \mathbf{x}_j) \prod_{k=1}^{N_{-1}} \sigma(-\mathbf{w}^T \mathbf{x}_k)$$

Criterio de aprendizaje: ML

Máxima probabilidad (ML - Maximum Likelihood): elija la hipótesis h que maximiza $L(\mathbf{w})$

Minimizar de forma equivalente ERM:

$$E_{in}(\mathbf{w}) = -\frac{1}{N} \ln(L(\mathbf{w})) = \frac{1}{N} \sum_{i=0}^N \ln\left(\frac{1}{P(y_i|\mathbf{x}_i)}\right) = \frac{1}{N} \sum_{i=0}^N (\ln(1+e^{-y_i \mathbf{w}^T \mathbf{x}_i}))$$

$$\hookrightarrow e(h(\mathbf{x}_i), y_i)$$

Podemos ver que $e(h(\mathbf{x}_i), y_i)$ es pequeño cuando $y_i \mathbf{w}^T \mathbf{x}_i \gg 0$

$$\begin{aligned} \nabla_{\mathbf{w}} E_{in}(\mathbf{w}) &= \frac{\partial}{\partial \mathbf{w}} \left(\frac{1}{N} \sum_{i=0}^N \ln(1+e^{-y_i \mathbf{w}^T \mathbf{x}_i}) \right) = \frac{1}{N} \sum_{i=0}^N -y_i \mathbf{x}_i \frac{e^{-y_i \mathbf{w}^T \mathbf{x}_i}}{1+e^{-y_i \mathbf{w}^T \mathbf{x}_i}} = \\ &= \frac{1}{N} \sum_{i=0}^N -y_i \mathbf{x}_i \sigma(-y_i \mathbf{w}^T \mathbf{x}_i) \end{aligned}$$

Clasificación de multietiquetas

- La regresión logística binaria se puede generalizar para hacer frente a k etiquetas
 - En el enfoque binario maximizamos la probabilidad del etiquetado binario dada por la probabilidad de N variables de Bernoulli $\{0, 1\}$

$$L(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) = \prod_{i=1}^N \sigma(\mathbf{w}^T \mathbf{x}_i)^{[y_i == 1]} (1-\sigma(\mathbf{w}^T \mathbf{x}_i))^{[y_i == 0]} = \prod_{i=1}^N \prod_{k=0}^{N-1} \sigma(\mathbf{w}^T \mathbf{x}_i)^{y_i k}$$

- Para las etiquetas K aplicamos la misma metodología, PERO ahora la probabilidad es definido usando N variables multi-Bernoulli (es decir, variables con k salidas)

- Las **etiquetas**, denotadas por y , están representadas por el **vector 1 de K** : $[0, 0, \dots, 1, \dots, 0]$ o $[-1, -1, \dots, 1, \dots, -1]$

La probabilidad de la muestra (K clases)

Sea (x, y) una sola muestra, **asumiendo independencia condicional**

$$P(y|w_1, \dots, w_k, x) = \prod_{k=1}^K P(y_k|w_k, x) = \prod_{k=1}^K \sigma(w_k^T x)^{y_k}$$

Sea $\{X, Y\}$ una muestra de N ítems

$$P(Y|w_1, \dots, w_k) = \prod_{n=1}^N \prod_{k=1}^K P(y_{nk}|w_k, x_n)$$

La probabilidad se define por

$$L(Y|w_1, \dots, w_k) = \prod_{n=1}^N \prod_{k=1}^K (\sigma(w_k^T x_n))^{y_{nk}}$$

Clasificación de multietiquetas: SoftMax

$$\bullet t_{nk} = \sigma(w_k^T x_n), \quad \sum_{k=1}^K t_{nk} = 1$$

$$\bullet E(w_1, \dots, w_k) = -\ln L(Y|w_1, \dots, w_k) = \sum_{n=1}^N \sum_{k=1}^K y_{nk} \ln t_{nk}$$

$$\bullet \nabla_{w_j} E(w_1, \dots, w_k) = \sum_{n=1}^N (t_{nj} - y_{nj}) x_n$$

• **SOFTMAX:** de cada muestra x se calcula un vector de probabilidades utilizando el vector

estimado w_1, \dots, w_k . $P(C_j|x) = \frac{\exp(w_j^T x)}{\sum_{k=1}^K \exp(w_k^T x)}, \quad j = 1, \dots, K$

Regla de decisión: Asignar x a C_j si $P(C_j|x) = \max_k P(C_k|x), \quad k = 1, \dots, K$

Regla de Bayes

• Una pregunta relevante para hacer es: **si conociéramos la verdadera distribución de probabilidad P , ¿podríamos generar una hipótesis óptima (la hipótesis de riesgo mínimo) para cualquier tarea de aprendizaje?**

• **Respuesta:** Dada cualquier distribución de probabilidad P sobre $X^* \{0,1\}$, **la mejor función de predicción** de etiquetas de X a $\{0,1\}$ será

$$f_p(x) = \begin{cases} 1 & \text{si } P[y=1|x] \geq \frac{1}{2} \\ 0 & \text{caso contrario} \end{cases} \quad \text{Regla de Bayes}$$

• No es difícil demostrar que para cualquier distribución de probabilidad NO existe otra función con menor riesgo que f_p , que para cualquier otra función g , $E_{out}(f_p) \leq E_{out}(g)$

• **Desafortunadamente, NO conocemos la distribución de probabilidad P**

• Un **enfoque alternativo para el aprendizaje** de funciones es estimar P a partir de las muestras y clasificar con la regla de Bayes (**MODELOS GENERATIVOS**)

“Resolver un problema nunca resuelve uno más complejo como paso intermedio” (Vapnik 1995)

ERM: Regla de Aprendizaje

- El objetivo: minimizar el error de muestreo para elegir una hipótesis final con error de salida baja E_{out} error.

- **Minimización de riesgos empíricos (ERM): ejemplos**

- Problemas de clasificación: $ERM_H(D) = \operatorname{argmin}_h \left\{ \frac{1}{N} \sum_{i=1}^N I[y_i \neq h(x_i)] \right\}$
- Problemas de regresión: $ERM_H(D) = \operatorname{argmin}_h \left\{ \frac{1}{N} \sum_{i=1}^N (y_i - h(x_i))^2 \right\}$
- Regresión logística: $ERM_H(D) = \operatorname{argmin}_h \left\{ \frac{1}{N} \sum_{i=1}^N \ln(1+e^{-y_i h(x_i)}) \right\}$

- Dos casos:

- Caso simple: f es una función DESCONOCIDA determinista (aquí, $y_i = f(x_i)$)
- Caso real: f es una función DESCONOCIDA estocástica dada por $P(Y|X)$

$$ERM_H(D) \in \operatorname{argmin}_{h \in H} E_{in}(h)$$

- Cuando $f \in H$, se espera que $ERM_H(D) \leq \varepsilon$ para algún valor-N finito

SGD + ERM: Una nueva regla de inducción

- Podemos aplicar SGD para resolver cualquier problema de optimización donde se pueda calcular el gradiente.

- Ejemplo 1: Regresión

- La función de error es $e_n(\mathbf{w}) = (y_n - \mathbf{w}^T \mathbf{x}_n)^2$
- SGD puede resolver el problema iterando sobre un montón de ejemplos.
- Además, la convergencia de regresión lineal se puede acelerar utilizando la segunda derivada y el **método de Newton**.

- Ejemplo 3: perceptrón (PLA)

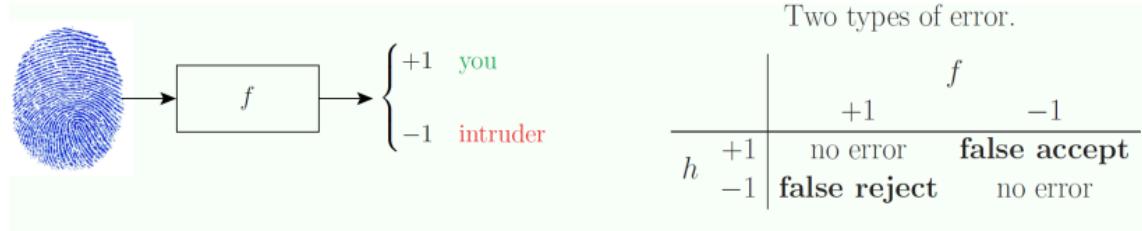
- En PLA minimizamos el error según la regla de actualización: $\mathbf{w}(t+1) = \mathbf{w}(t) + y_i \mathbf{x}_i$
- Esta regla puede verse como una regla SGD con $\eta = 1$ y gradiente dado por $y_i \mathbf{x}_i$
- Se puede verificar que la función $e_n(\mathbf{w}) = \max(0, -y_n \mathbf{w}^T \mathbf{x}_n)$ proporciona el gradiente

- Ejemplo 2: Regresión logística (LGR)

- Ahora la función de error es $e_n(\mathbf{w}) = \ln(1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n})$
- Para \mathbf{w} grande $e_n(\mathbf{w}) \approx \max(0, -y_n \mathbf{w}^T \mathbf{x}_n)$ lo que hace que LGR+SGD sea equivalente a PLA

COSTE DE ERROR

Medida de error especificada por el usuario



En cualquier aplicación hay que pensar en cómo penalizar cada tipo de error

		f	
		$+1$	-1
h	$+1$	0	1
	-1	10	0

		f	
		$+1$	-1
h	$+1$	0	1000
	-1	1	0

Supermarket

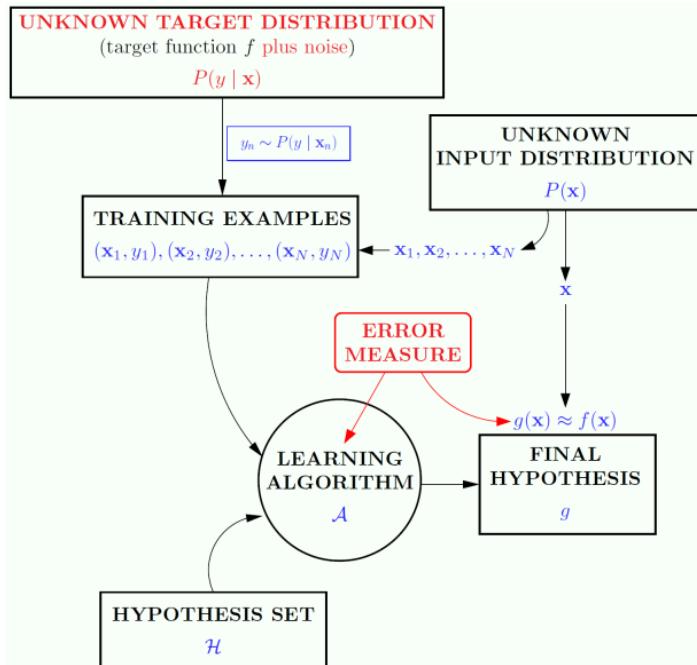
CIA

Take Away: la medida del error la especifica el usuario.

Si no, elija uno que sea:

- Plausible (conceptualmente atractivo)
- Amistoso (prácticamente atractivo)

Configuración de aprendizaje con medición de error y objetivo ruidoso





Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.

Available on the
App Store

GET IT ON
Google Play



122

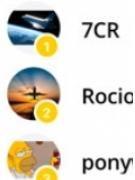
l

[Ver mis op](#)

Continúa d



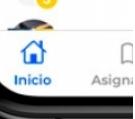
Top de tu gu



7CR

Rocio

pony

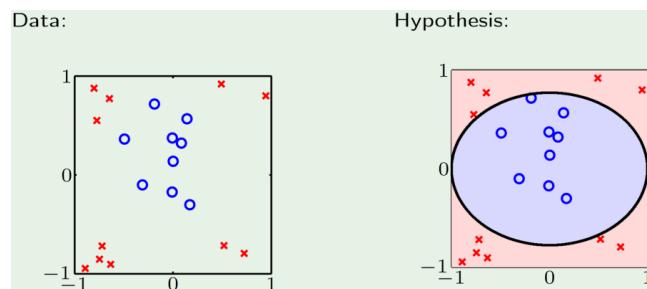


Inicio

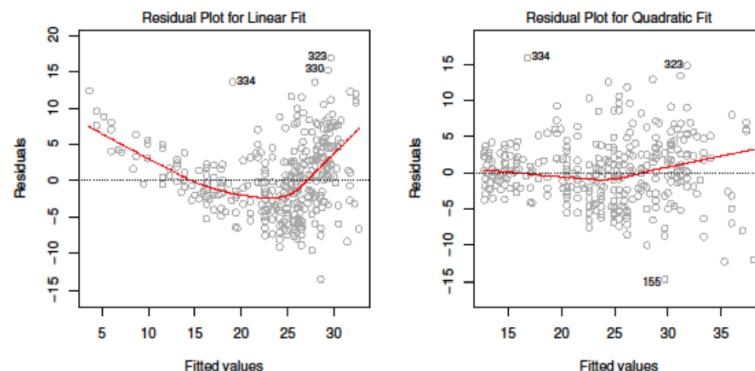
Asign

AGREGAR FUNCIONES A H : TRANSFORMACIONES NO LINEALES

Los predictores lineales son limitados



Regresión: predictores no lineales



- Gráfico de la izquierda: un modelo de entidad lineal muestra un comportamiento no lineal claro en el residuo
- Gráfico de la derecha: un modelo de características cuadráticas mejora el comportamiento residual

Modelo lineal significa linealidad en W

- Ejemplo de crédito: la línea de crédito se ve afectada por los "años de residencia" pero NO de forma lineal

• Características no lineales: $[x_i > 5]$ y $[x_i < 1]$ podrían ser características mejores

¿Cómo podemos hacer esto con modelos lineales?

- La regresión lineal calcula: $\sum_{i=0}^d x_i w_i$

- La clasificación lineal calcula: $\text{sign}(\sum_{i=0}^d x_i w_i)$

Los algoritmos funcionan debido a la **linealidad en los pesos**

Transformaciones no lineales (NLT)

- Un modelo lineal solo tiene que ser lineal en w
- Por lo tanto, podemos usar funciones no lineales de las características x_i y aún permanecer dentro de la clase del modelo lineal.
 - Funciones polinomiales: x^2, x^3, \dots, x^n
 - Funciones trascendentes: $\log(x), \sin(x), x^{1/2}, \dots,$
 - Funciones booleanas: $[x > 1] \& [x < 5], [x < 1] \& [x > 5]$, etc.
 - etc
- El uso de NLT NO cambia la clase de hipótesis H PERO el espacio-X original se transforma en el espacio-Z, incluidas las características no lineales.
- Las transformaciones no lineales deben usarse cuando:
 - El error residual después del ajuste con las características originales muestra un comportamiento no lineal
 - Sabemos que algunas características son funciones no lineales de otras
 - PERO NO SOLO !!

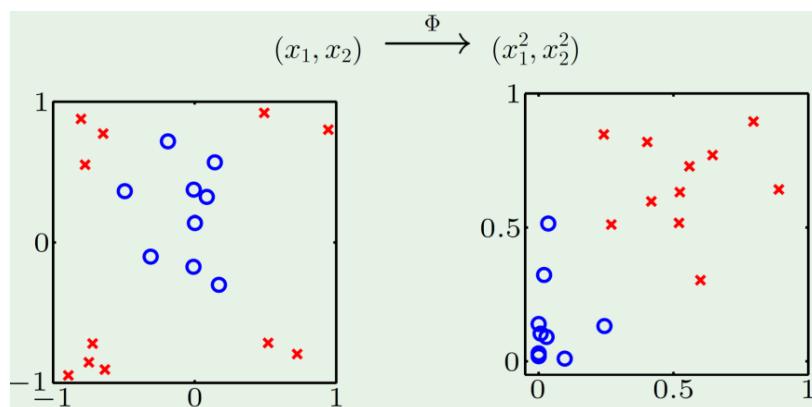
Regresión Polinomial

- En este caso, los predictores son potencias de una sola característica, por ejemplo $y = \beta_0 + \beta_1 X + \beta_2 X^2$ (modelo cuadrático) $y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3$ (modelo cúbico)
- Claramente, todos los predictores son independientes
- La matriz X para datos muestrales de tamaño N se define como

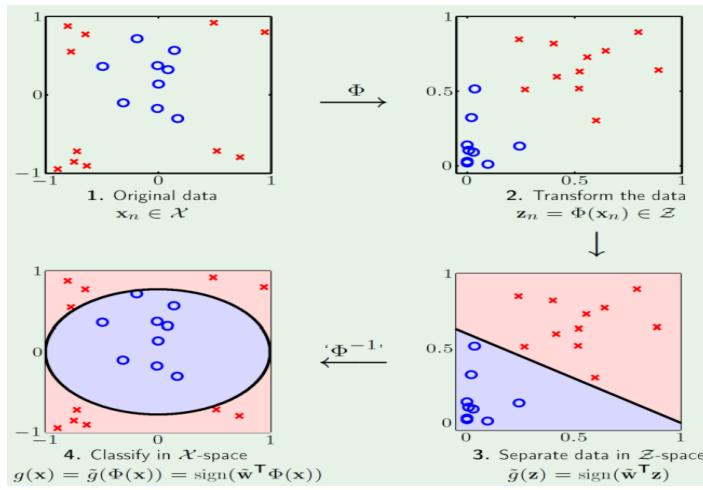
$$X = \begin{bmatrix} 1 & x_1 & \dots & x_1^p \\ 1 & x_2 & \dots & x_2^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & \dots & x_N^p \end{bmatrix}$$

- ¿Cuántos términos elegir? Hablamos de eso más tarde.

Transforma los datos de forma no lineal



Transformaciones no lineales



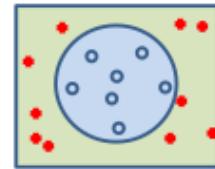
Qué se transforma en qué

$$\begin{array}{lll}
 \mathbf{x} = (x_0, x_1, \dots, x_d) & \xrightarrow{\Phi} & \mathbf{z} = (z_0, z_1, \dots, z_d) \\
 x_1, x_2, \dots, x_N & \xrightarrow{\Phi} & \mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N \\
 y_1, y_2, \dots, y_N & \xrightarrow{\Phi} & y_1, y_2, \dots, y_N
 \end{array}$$

$$\begin{array}{ll}
 \text{Sin pesos en } \mathbf{X} & \hat{\mathbf{w}} = (w_0, w_1, \dots, w_d) \\
 g(\mathbf{x}) & = \text{sign}(\hat{\mathbf{w}}^T \Phi(\mathbf{x}))
 \end{array}$$

Ejemplo NTL

- Consideremos el problema de clasificación dado en la figura.
 - Supongamos que la función no lineal $x_1^2 + x_2^2 = 0.6$ representa la curva de separación entre las dos clases (dentro y fuera del círculo).
 - Entonces sabemos que la hipótesis $h(\mathbf{x}) = \text{sign}((1)x_1^2 + (1)x_2^2 - 0.6)$ resuelve el problema, **PERO** esta hipótesis NO puede ser implementada por el PLA regular
- $(x_1, x_2, 1) \xrightarrow{\Phi} (x_1^2, x_2^2, 1) = (z_1, z_2, 1)$
- Escriba la solución usando z : $\hat{h}(\mathbf{z}) = \text{sign}((1)z_1 + (1)z_2 - 0.6)$ ¡¡Bien !!
 $\hat{h}(\mathbf{z}) = \hat{h}(\Phi(\mathbf{x})) = \hat{h}(\mathbf{x})$
 - En problemas reales, debemos configurar la transformación antes de ver los datos.
Entonces, no sabemos de antemano cuál es la mejor transformación específica.



Funciones de error / pérdida generalizadas

- Denotemos $l(h, z)$ una función de error / pérdida general de una hipótesis h en un ejemplo $z = (\phi(\mathbf{x}), y)$
 $l(h, z)$ representa una de estas funciones: $\{(h(z) - y)^2, 1(h(z) \neq y), \text{etc}\}$
- $l_{0-1}(h, z) \stackrel{\text{def}}{\rightarrow} 0 \text{ si } h(\Phi(\mathbf{x})) = y$
 $\rightarrow 1 \text{ si } h(\Phi(\mathbf{x})) \neq y$ Pérdida 0-1
- $l_{sq}(h, z) \stackrel{\text{def}}{\equiv} (h(\Phi(\mathbf{x})) - y)^2$ Pérdida cuadrática
- La función de riesgo se utiliza para medir el error / pérdida esperado de una hipótesis con respecto a una probabilidad de distribución P : $L_p(h) \stackrel{\text{def}}{=} E_{z \sim p}[l(h, z)] \stackrel{\text{def}}{=} E_{\text{out}}(h)$
 - Esa definición es un promedio de todos los ejemplos posibles (riesgo $(h) =$ pérdida promedio)
 - Esta expresión es aplicable en cualquier entorno de aprendizaje.

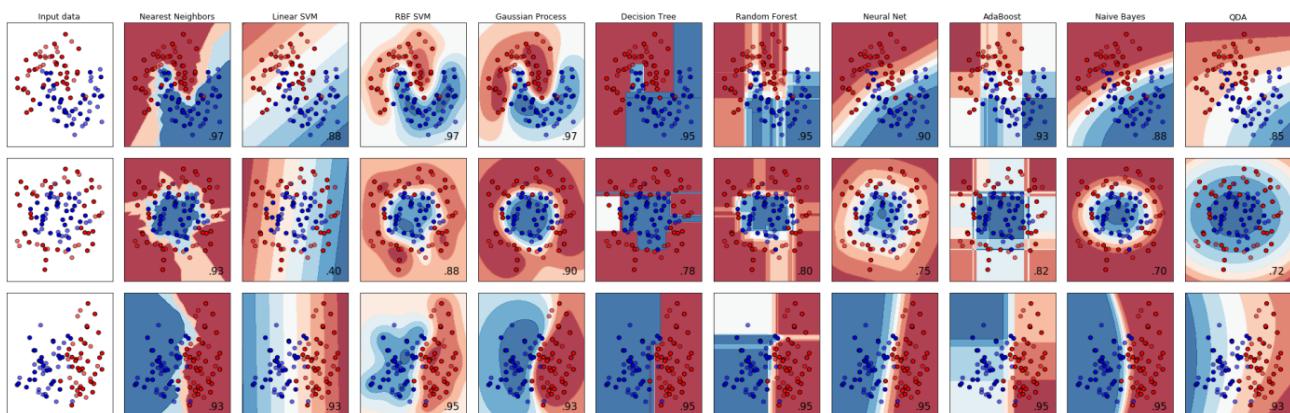
Cálculo y Generalización

- Denotemos por Φ_Q la transformada polinomial de Q-ésimo orden
 - $\Phi_4(x) = (1, x_1, x_2, x_1^2, x_2^2, x_1x_2, x_1^3, x_2^3, x_1x_2^2, x_2x_1^2, x_1^4, x_2^4, x_1^2x_2^2, x_2^1x_1^3, x_1^1x_2^3)$

Una Q más grande proporciona una mayor flexibilidad en términos de la forma del límite de decisión, pero hay un precio que pagar.

- El cálculo es un problema porque la transformación de características Φ_Q asigna x (el vector inicial) a $d = \frac{Q(Q+3)}{2} = O(Q^2)$ dimensiones, lo que aumenta la memoria y el costo computacional.
 - Cuanto mayor sea el valor-Q, mayor (orden cuadrático) será el número de muestras que necesitaremos para obtener el mismo nivel de error de generalización. (recuerde $O(dN^{-1} \log(N))$)
- En general, al elegir la dimensión adecuada para la transformación de características, debemos utilizar una compensación de aproximación-generalización:
 - mayor d mayor probabilidad de ser linealmente separables ($E_{in} \downarrow$) y $E_{out} \uparrow$
 - menor d posiblemente no separable linealmente ($E_{in} \uparrow$) y $E_{out} \downarrow$

Different learners (A, H)



MÉTRICAS DE EVALUACIÓN

Alias y otras medidas

Accuracy : tasa de error

Recall = TRP = Tasa de aciertos = Sensitivity

Fallout = FRP = Tasa de Falsas Alarmas

Precision = Valor Predictivo Positivo (VPP)

Valor predictivo negativo (VPN) = TN / (TN + FN)

Razones de probabilidad (Likelihood Ratios):

LR+ = Sensitivity / (1-specificity)

LS- = (Sensitivity) / specificity

Pares de medidas y medidas compuestas

Precision / Recall

Sensitivity / Specificity

Razones de probabilidad (LR+ y LR-)

Valores Predictivos Positivos / Negativos

Estudiar sin publi es posible.

Compra Wuolah Coins y que nada te distraiga durante el estudio.



F-Measure: $\alpha = 1, 2, 0.5$

$$F_\alpha = [(1+\alpha) (\text{Precision} * \text{Recall})] / [(\alpha * \text{Precision}) + \text{Recall}]$$

G-Mean = $\text{Sqrt}(TPR * TNR)$ or $\text{Sqrt}(TPR * \text{Precisión})$

Versión clase 2

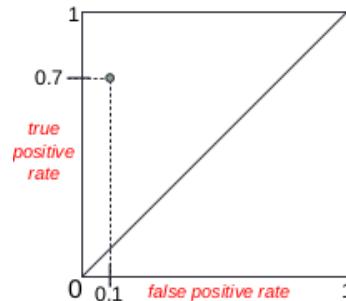
Versión clase única

Evaluación de la salida (Caso Binario)

¿Cómo podemos medir el desempeño de un clasificador determinista?

"recall"
$$\frac{\# \text{ verdaderos positivos } (TP_{\text{true positives}})}{\# \text{ positivos } (TP+FP_{\text{false positives}})}$$

1-”specificity”
$$\frac{\# \text{ falsos positivos } (FP)}{\# \text{ negativos } (TN_{\text{true negatives}}+FP)}$$



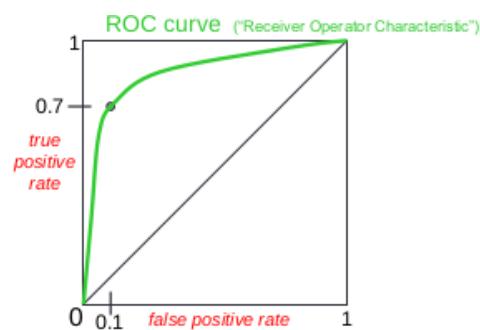
Curva de ROC (Caso Binario)

¿Cómo podemos medir el desempeño de un clasificador determinista?

AUC (Area under the curve): área bajo la curva:

El AUC de un clasificador es equivalente a la probabilidad de que el clasificador clasifique una instancia positiva elegida aleatoriamente por encima de una instancia negativa elegida aleatoriamente.

Se necesita una estimación de la varianza (curva de ROC) para comparar dos clasificadores.



Medidas de desempeño basadas en matriz de confusión

True class → Hypothesized class	Pos	Neg
Yes	TP	FP
No	FN	TN
	P=TP+FN	N=FP+TN

A Confusion Matrix

Enfoque de varias clases:

$$\text{Accuracy} = (TP + TN) / (P+N)$$

Enfoque de clase única:

$$\text{Precision} = TP / (TP+FP)$$

$$\text{Recall} = TP / P$$

$$\text{Fallout} = FP / N$$

$$\text{Sensitivity} = TP / (TP+FN)$$

$$\text{Specificity} = TN / (FP+TN)$$



WUOLAH

Sesión 3

- El **aprendizaje inductivo** es un enfoque desesperado:

¡En un sentido estricto, aprender de la muestra no es posible!

(ver Turquía inductivista (Bertrand Russell) ☺)

¿Hay alguna esperanza de saber algo sobre f fuera del conjunto de datos **sin hacer supuestos** acerca de f ?

Sí, si estamos dispuestos a rendirnos "seguro".

Trate de aprender algo menos exigente que la función **desconocida** adecuada, es decir, alguna propiedad útil sobre la función **desconocida**

Intentemos aprovechar la aleatoriedad ...

- **NUEVA hipótesis:** los elementos dentro de D son **muestras i.i.d de una distribución de probabilidad P**
- **Consecuencias:**
 - D es la salida de una variable aleatoria (vector)
 - No es realista esperar que cada muestra D represente igualmente bien la distribución P
 - La función g depende de D , por lo que su elección también es un proceso aleatorio.
- **¿Dónde está la novedad?**
 - La teoría de la probabilidad muestra que existen dependencias probabilísticas entre una variable aleatoria y una muestra de ella (bajo condiciones).
 - Ejemplo: intervalo de confianza para la media muestral

Pero, ¿es suficiente la probabilidad?

- **PREGUNTA PRINCIPAL:**

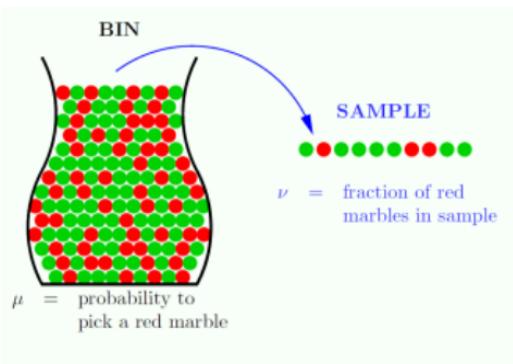
Existe un algoritmo de aprendizaje A y un tamaño de muestra m tal que para cada distribución P , si A recibe m i.i.d. muestras de P , hay una alta probabilidad de que genere un predictor g con un error bajo?

- **Teorema de no almuerzo gratis (NFL) (informal):** “Para cada algoritmo existe un P en el que falla, aunque ese P puede ser aprendido con éxito por otro learner. Además, todos los algoritmos son equivalentes en promedio en todas las posibles funciones objetivo f ”
- Para tener éxito, cada alumno (A, H) debe aplicarse en la clase de distribuciones P que puede aprender.
- Esto destaca la necesidad de **explorar el conocimiento específico del problema** para lograr un desempeño mejor que el aleatorio.
 - Restricción geométrica
 - Clase de función con salida E_{out} cero o muy pequeña
 - Clase finita H
 - Dimensión VC finita
 - etc

¿Podemos inferir algo fuera de los datos usando solo D ?: La respuesta del PAC

Media de la población a partir de la media de la muestra

BIN MODEL



The BIN Model

- Bin with red and green marbles.
 - Pick a sample of N marbles *independently*.
 - μ : probability to pick a red marble.
 ν : fraction of red marbles in the sample.
- Sample → the data set → ν
BIN → outside the data → μ

¿Podemos garantizar algo sobre μ (fuera de los datos) después de observar ν (los datos)?

RESPUESTA: No. Es **posible** que la muestra sea todo canicas verdes y que el contenedor sea mayormente rojo.

Entonces, ¿por qué confiamos en las encuestas (por ejemplo, para predecir el resultado de una elección presidencial)?

RESPUESTA: El caso malo es **posible**, pero **no probable**.

Desigualdad de Hoeffding

Hoeffding/Chernoff demostró que, la mayoría de las veces, **para un μ fijo, ν no puede estar demasiado lejos de μ**

$$\mathbb{P}(D:|\mu-\nu| > \epsilon) \leq 2e^{-2\epsilon^2 N} \quad \text{para todo } \epsilon > 0$$

$$\mathbb{P}(D:|\mu-\nu| \leq \epsilon) \geq 1 - 2e^{-2\epsilon^2 N} \quad \text{para todo } \epsilon > 0$$

Pregunta: ¿Qué nos dice el valor de ν sobre μ : $\mu \approx \nu \Leftrightarrow \nu \approx \mu$

Example: $N = 1,000$; draw a sample and observe ν .

$$99\% \text{ of the time} \quad \mu - 0.05 \leq \nu \leq \mu + 0.05 \quad (\epsilon = 0.05)$$

$$99.999996\% \text{ of the time} \quad \mu - 0.10 \leq \nu \leq \mu + 0.10 \quad (\epsilon = 0.10)$$

What does this mean? If I repeatedly pick a sample of size 1,000, observe ν and claim that

$$\mu \in [\nu - 0.05, \nu + 0.05], \quad (\text{the error bar is } \pm 0.05)$$

I will be right 99% of the time. On any particular sample you may be wrong, but not often.

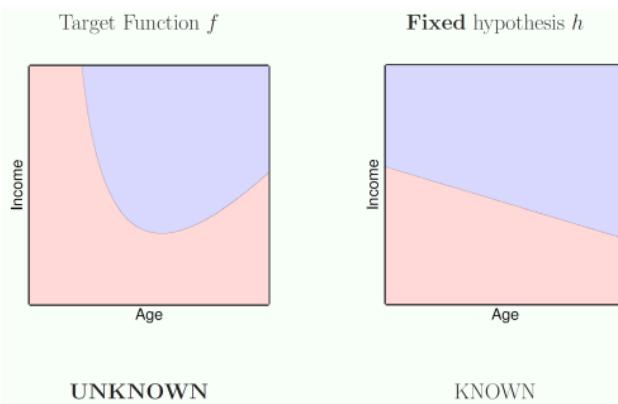
Aprendimos algo. De ν , llegamos fuera de los datos a μ .

Desigualdad de Hoeffding: hechos notables

- El ingrediente clave: las muestras deben ser i.i.d.
 - Si la muestra se construye de alguna manera arbitraria, entonces, de hecho, no podemos decir nada.
 - Incluso con independencia, v puede asumir valores arbitrarios; pero algunos valores son más probables que otros.
 - Esto es lo que nos permite aprender algo - es probable que $v \approx \mu$.
- El límite $2e^{-2\epsilon^2 N}$ no depende de μ o del tamaño del contenedor
 - El contenedor puede ser infinito.
 - Es genial que no dependa de μ porque μ es desconocido; y queremos decir **desconocido**.
- El jugador clave en el límite $2e^{-2\epsilon^2 N}$ es N
 - Si $N \rightarrow \infty$, $\mu \approx v$ con muy muy muy... alta probabilidad, pero no con seguridad.
 - ¿Puede vivir con una probabilidad de error de 10 - 100?

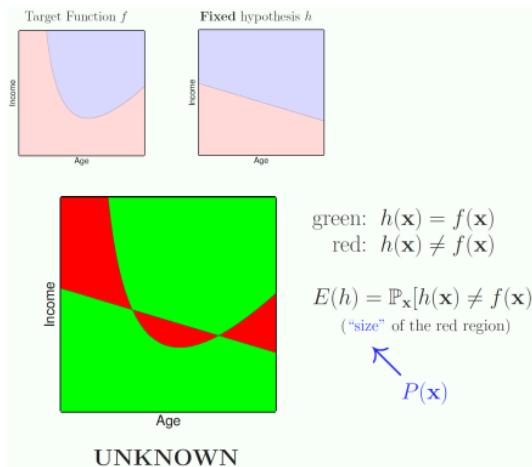
$$\mathbb{P}(D: |u-v| > \epsilon) \leq 2e^{-2\epsilon^2 N}$$

Configuración de aprendizaje



En el aprendizaje, lo desconocido es una función completa f ; en el contenedor era un solo número μ .

La función de error de aprendizaje



La función h define un error de probabilidad desconocido pero fijo $E(h)$



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.

Available on the
App Store

GET IT ON
Google Play



122

I

Ver mis op

Continúa d

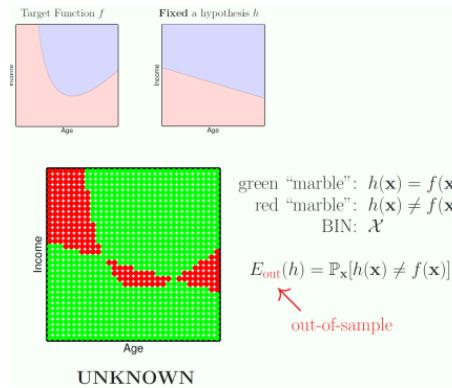
405416_arts_esce
ues2016juniy.pdf

Top de tu gu

7CR
Rocio
pony

Inicio Asigni

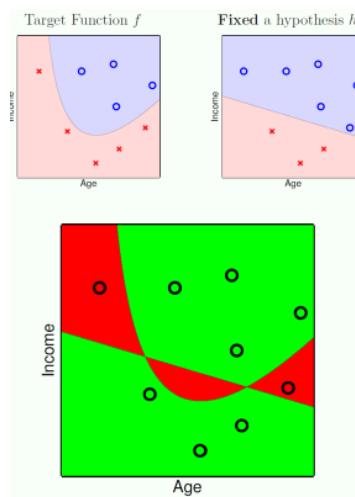
Relacionar la papelera con el aprendizaje



Consideremos todos los posibles puntos muestrales

Ahora, un modelo de contenedor está definido por h y f

Relacionar la papelera con el aprendizaje: los datos

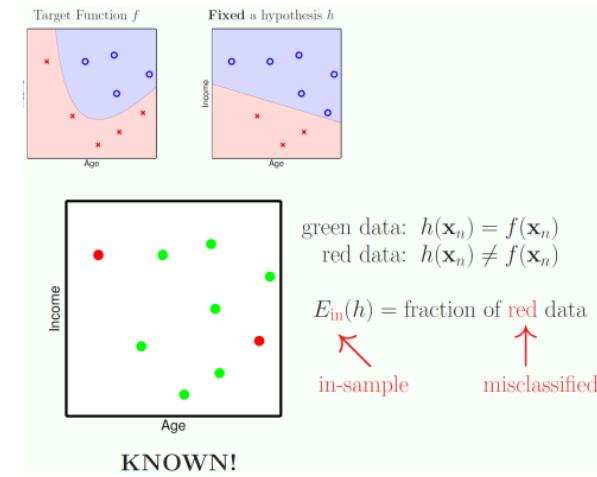


En la misma muestra, la función objetivo f y la hipótesis h nos proporcionan diferentes etiquetas

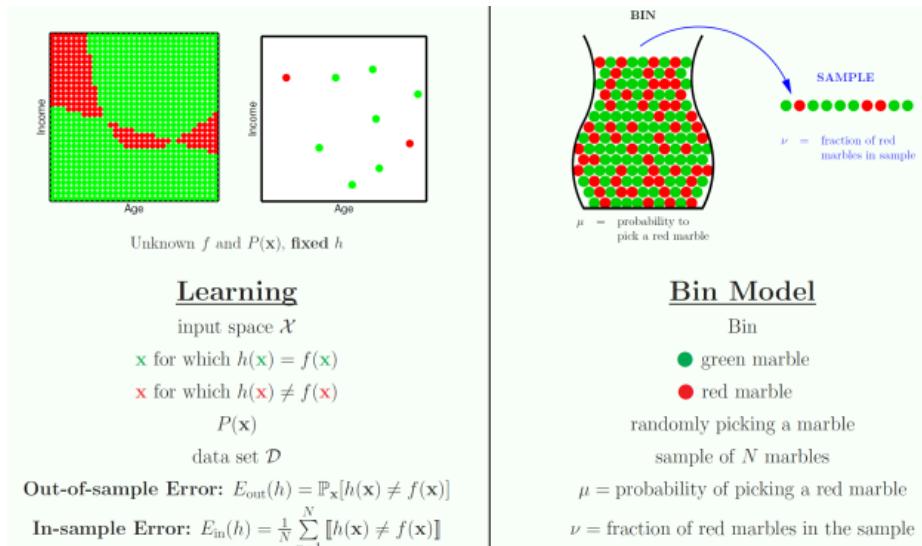
Tenemos puntos en diferentes zonas de la función de error.

Si la muestra se extrae de forma independiente de acuerdo con P , cada punto será rojo con probabilidad μ y verde con probabilidad $1-\mu$

Relacionar el contenedor con el aprendizaje: los datos



Modelo de contenedor y aprendizaje



Hoeffding la desigualdad en el aprendizaje

- Consideremos $H = \{h\}$, solo una función, y $f(x)$ la función verdadera desconocida.
 - Vamos a representar $[f(x) = h(x)]$ y $[f(x) \neq h(x)]$ nuevas variables binarias en la población. Ahora $\mu = \Pr([f(x) \neq h(x)])$
 - Para cualquier muestra de entrenamiento D de tamaño N , $v = \text{Fracción}([f(x) \neq h(x)])$ en D
 - Ahora μ y v representan la población y el error muestral respectivamente.
 - Denotemos por $E_{out}(h) = \mu$ y $E_{in}(h) = v$ el error global y muestral de h respectivamente
 - La desigualdad de Hoeffding se puede reescribir como:
- $$\mathbb{P}(D: |E_{out}(h) - E_{in}(h)| > \epsilon) \leq 2e^{-2\epsilon^2 N} \quad \text{para todo } \epsilon > 0$$
- Esto se denomina resultado "Probablemente aproximadamente correcto (PAC)".
- IMPORTANTE: Tenga en cuenta que h es fijo antes de conocer la muestra de datos

Hoeffding dice que $E_{in}(h) \approx E_{out}(h)$

$$\mathbb{P}(D: |\mu - v| > \epsilon) \leq 2e^{-2\epsilon^2 N} \iff \mathbb{P}(D: |E_{out}(h) - E_{in}(h)| > \epsilon) \leq 2e^{-2\epsilon^2 N}$$

E_{in} es aleatorio, pero conocido; E_{out} fijo, pero desconocido.

$N \gg 1$

- Si $E_{in}(h) \approx 0 \Rightarrow E_{out}(h) \approx 0$ (con alta probabilidad), es decir, $\mathbb{P}_x[h(x) \neq f(x)] = 0$
 - Hemos aprendido algo sobre todo $f: f \approx h$ sobre X (fuera de D)
- Si $E_{in} \gg 0$, no tenemos suerte.
 - Pero, todavía hemos aprendido algo sobre todo $f: f \neq h$ sobre X ; aunque no es muy útil.

Preguntas:

- Suponga que $E_{in} = 1$, ¿hemos aprendido algo sobre la f completa que sea útil?
- ¿Cuál es la peor E_{in} para inferir acerca de f ?

Comprensión de los resultados de la PAC

$$\mathbb{P}(D: |E_{out}(h) - E_{in}(h)| > \epsilon) \leq 2e^{-2\epsilon^2 N} \quad \text{para todo } \epsilon > 0$$

- Consideremos $\delta = 2e^{-2\epsilon^2 N}$ entonces

$$\mathbb{P}(D: |E_{out}(h) - E_{in}(h)| > \epsilon) \leq \delta \Leftrightarrow \mathbb{P}(D: |E_{out}(h) - E_{in}(h)| < \epsilon) \geq 1 - \delta$$

- O equivalente:

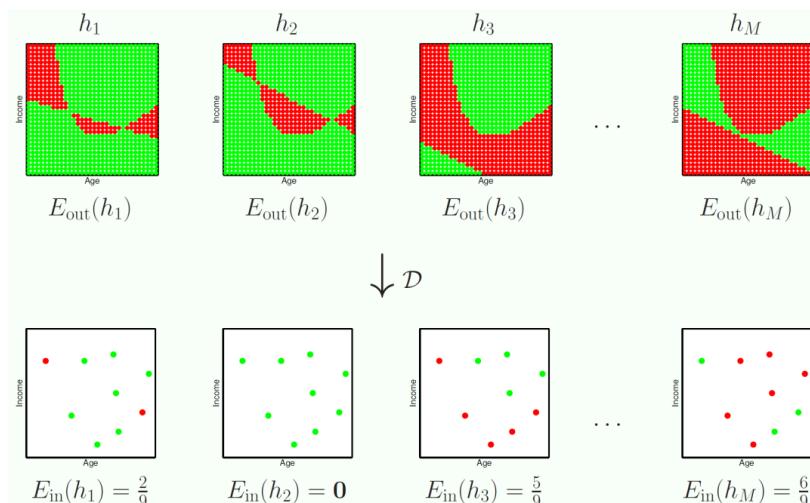
$$E_{out}(h) \leq E_{in}(h) + \epsilon, \text{ con probabilidad de al menos } 1 - \delta \text{ en } D$$

- Escribamos ϵ en función de N y δ , luego

$$E_{out}(h) \leq E_{in}(h) + \sqrt{\frac{1}{2N} \log \frac{2}{\delta}} \text{ con probabilidad de al menos } 1 - \delta \text{ en } D$$

- Cuanto mayor sea N , más estrecho será el intervalo (el tamaño de la muestra es importante!)
- Cuanto menor sea δ , mayor será el intervalo (cuanto mayor sea la garantía, menor precisión)

Aprendizaje real: modelo de aprendizaje finito



Elegir la hipótesis con E_{in} mínimo; ¿ E_{out} será pequeño?

La desigualdad de Hoeffding para hipótesis múltiples

LAS COSAS SON DIFERENTES: En la desigualdad de Hoeffding la h se fija antes de conocer los datos, PERO en PROBLEMAS REALES la hipótesis elegida, $g \in H$ se identifica usando los datos.

LA BÚSQUEDA CAUSA EL SESGO DE SELECCIÓN: LA ANALOGÍA DE LA MONEDA

Pregunta: si lanza una moneda justa diez veces, ¿cuál es la probabilidad de que obtenga diez caras?

Respuesta: ≈ 0.1 (pruébalo)

Pregunta: si lanza 1000 monedas justas diez veces, ¿cuál es la probabilidad de que alguna moneda obtenga diez caras?

Respuesta: ≈ 0.63 (pruébalo)

Identificación de monedas con funciones: cuanto mayor sea el tamaño de H , mayor será la probabilidad de tener una hipótesis con E en un error 0, PERO, ¿podemos esperar que E sea pequeña?

¿Ahora qué?

- Adaptar la desigualdad de Hoeffding al **caso de H finito**
 1. La solución de hipótesis g **debe fijarse antes de conocer la muestra de datos.** (**CONDICIÓN OBLIGATORIA**)
 2. No obstante, el algoritmo de aprendizaje utiliza los datos de entrenamiento para buscar g .
- Una solución simple es considerar un evento válido para todas las funciones en H .
 - Sea g una solución de hipótesis genérica, entonces,

$$\{D:|E_{in}(g) - E_{out}(g)| > \epsilon\} = \bigcup_{h \in H} \{D:|E_{in}(h_i) - E_{out}(h_i)| > \epsilon\}$$

- Usando $P(\bigcup_{i=1:|H|} B_i) \leq \sum_{i=1}^{|H|} P(B_i)$

$$P(D:|E_{in}(g) - E_{out}(g)| > \epsilon) < 2|H|e^{-2\epsilon^2 N} \quad \text{para todo } \epsilon > 0$$

Convergencia uniforme

Interpretando el límite

$$P(D:|E_{in}(g) - E_{out}(g)| > \epsilon) < 2|H|e^{-2\epsilon^2 N} \quad \text{para todo } \epsilon > 0$$
$$P(D:|E_{in}(g) - E_{out}(g)| \leq \epsilon) \geq 1 - 2|H|e^{-2\epsilon^2 N} \quad \text{para todo } \epsilon > 0$$

Resultado: Con probabilidad de al menos $1-\delta$, $E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{1}{2N} \ln \frac{2|H|}{\delta}}$

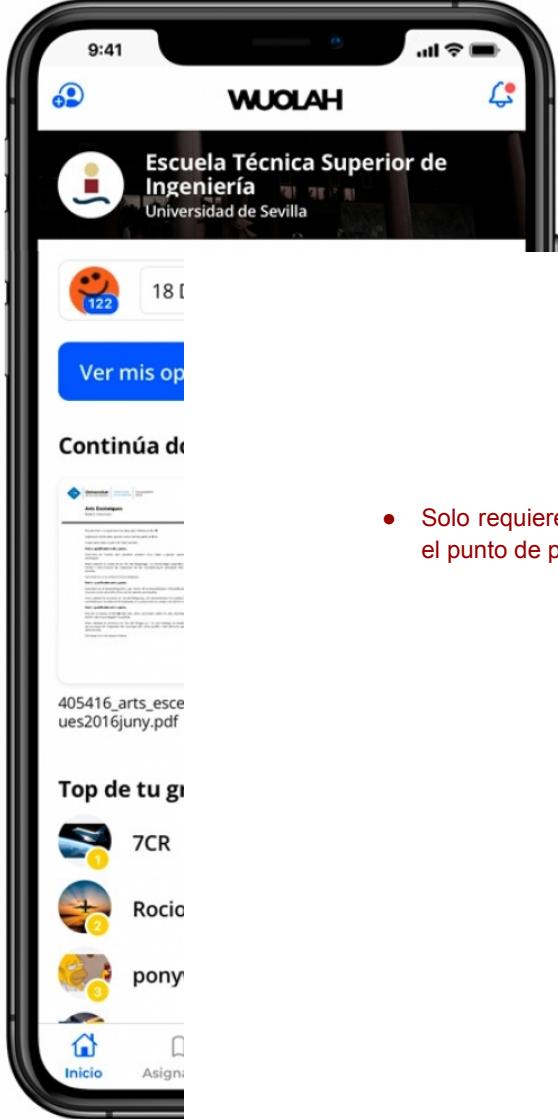
↳ Error de la barra de generalización

Denotemos $\delta = 2|H|e^{-2\epsilon^2 N}$ y escribiendo ϵ en función de N , δ y $|H|$

E_{in} llega fuera de E_{out} cuando $|H|$ es pequeño $E_{out}(g) \leq E_{in}(g) + O(\sqrt{\frac{\ln |H|}{N}})$

Si $N \gg \ln |H|$, entonces $E_{out}(g) \approx E_{in}(g)$

- **Este límite no depende de X , $P(x)$, f ni de cómo se encuentra g .** (un destino en el peor de los casos)



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.

Available on the
App Store

GET IT ON
Google Play

Un caso particular: la hipótesis de la realizabilidad

- Definición (H-realizable): existen $h^* \in H$ s.t. $E_{\text{out } P, f}(h^*) = 0$
- Esto significa que la clase de funciones H incluye al menos una función con error cero para cualquier P y f .
- Al ser una hipótesis matemática se puede aplicar en tareas reales.
 - Ejemplo: Clases separadas en clasificación
- Bajo esta hipótesis se puede demostrar que la regla ERM sobre clases finitas H siempre proporciona una función h_s s.t. $E_{\text{en}}(h_s) = 0$, obteniendo el límite de error más pequeño

$$E_{\text{out}}(h_s) \leq \frac{1}{N} \log \frac{|H|}{\delta} \text{ con probabilidad al menos } 1 - \delta$$

Definición formal de aprendizaje PAC realizable

- Una clase de hipótesis **realizable** H es PAC si existe una función $m_H(\epsilon, \delta) \rightarrow N$ y un algoritmo de aprendizaje A con la siguiente propiedad:
 - Para cada, $\epsilon, \delta \in (0, 1)$
 - Para cada distribución P sobre X
 - Para cada función de etiquetado f

Si la suposición realizable se cumple con respecto a H, P, f , entonces cuando se ejecuta el algoritmo de aprendizaje en $N \geq m_H(\epsilon, \delta)$ i.i.d muestras generadas por P y etiquetadas por f , el algoritmo devuelve una hipótesis h tal que, con probabilidad al menos $1 - \delta$ (sobre la elección de las m muestras de entrenamiento)

$$ERM_{P, f}(h) \leq \epsilon$$

Esto significa que el algoritmo de aprendizaje siempre devuelve una hipótesis h con un error cercano a cero.

Muestra de complejidad en PAC-Learnability

- ¿Cuántos ejemplos se requieren para garantizar una solución PAC en clases realizables?
- Esto depende de ϵ, δ, H y del rango de la función de pérdida
- Si H es un PAC realizable, existen muchas funciones $m_H(\epsilon, \delta)$ que satisfacen los requisitos dados en la definición de capacidad de aprendizaje del PAC realizable.
- La complejidad de la muestra para aprender H se define como el “número entero mínimo” $m_H(\epsilon, \delta)$ que satisface los requisitos del aprendizaje PAC realizable con precisión ϵ y confianza δ
- Resultado de la función de pérdida con rango en $[0, 1]$: cada clase de hipótesis finita realizable se puede aprender con PAC con complejidad de muestra: $m_H(\epsilon, \delta) \leq \frac{1}{\epsilon} \log \frac{|H|}{\delta}$

Definición formal de aprendizaje PAC agnóstico

- Una clase de hipótesis H es apta para el PAC si existe una función $m_H(\epsilon, \delta) \rightarrow N$ y un algoritmo de aprendizaje A con la siguiente propiedad:
 - Para cada, $\epsilon, \delta \in (0, 1)$
 - Para cada distribución P sobre X

Cuando se ejecuta el algoritmo de aprendizaje en $N \geq m_H(\epsilon, \delta)$ i.i.d muestras generadas por P , el algoritmo devuelve una hipótesis h tal que, con probabilidad de al menos $1 - \delta$ (sobre la elección de las m muestras de entrenamiento)

$$E_{\text{in } P}(h) \leq \min_{h' \in H} E_{\text{in } P}(h') + \epsilon$$

Esto significa que el algoritmo de aprendizaje siempre devuelve una hipótesis h más cercana a la mejor posible dentro de la clase H

La regla ERM es un aprendiz de PAC agnóstico exitoso para clases finitas H

Complejidad de la muestra Agnostic-PAC

- ¿Cuántos ejemplos se requieren para garantizar una solución PAC?
- Esto depende de ϵ , δ , H y del rango de la función de pérdida
- Si H es PAC aprendible, existen muchas funciones $m_H(\epsilon, \delta)$ que satisfacen los requisitos dados en la definición de aprendibilidad PAC
- La complejidad de la muestra de aprendizaje H se define como el "número entero mínimo" $m_H(\epsilon, \delta)$ que satisface los requisitos del aprendizaje PAC con precisión ϵ y confianza δ
- Resultado de la función de pérdida con rango en $[0,1]$: cada clase de hipótesis finita se puede aprender con PAC con complejidad de muestra

$$m_H(\epsilon, \delta) \leq \frac{2}{\epsilon^2} \log \frac{2|H|}{\delta}$$
- ¡Compare con el caso PAC-realizable!

Viabilidad del aprendizaje versus complejidad

- El aprendizaje sólo es posible en un entorno probabilístico (bajo condiciones):
 - Las muestras de X deben ser i.i.d
 - Misma distribución de probabilidad en entrenamiento y prueba
- Tener éxito en el aprendizaje significa encontrar una función g , s.t. $E_{out}(g) \approx 0$
- Sin embargo, solo podemos garantizar,

$$\mathbb{P}(D: |E_{in}(g) - E_{out}(g)| > \epsilon) < 2|H|e^{-2\epsilon^2 N} \quad \text{para todo } \epsilon > 0$$
- La viabilidad del aprendizaje debe responder a dos preguntas:
 1. ¿Podemos asegurarnos de que $E_{out}(g)$ esté lo suficientemente cerca de $E_{in}(g)$?
 2. ¿Podemos hacer que $E_{in}(g)$ sea lo suficientemente pequeño?
- ¿Cuál es la relación entre la viabilidad del aprendizaje y la complejidad de H y f ?

Viabilidad de aprendizaje: $E_{out} \approx 0$

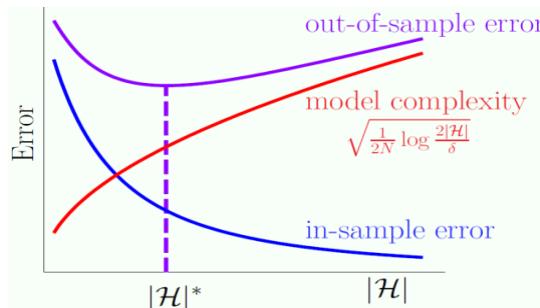
Dos condiciones:

1. $E_{in} \approx E_{out}$ -> Se verifica gracias a la desigualdad de Hoeffding.
2. $E_{in} \approx 0$ -> Se logra a través del algoritmo de aprendizaje.

Juntos, estos aseguran $E_{out} \approx 0$

PERO hay una compensación en H :

- Pequeño $|H|$ $E_{in} \approx E_{out}$
- Grande $|H|$ $E_{in} \approx 0$ es más probable



¿Qué pasa con la complejidad DESCONOCIDA de f :

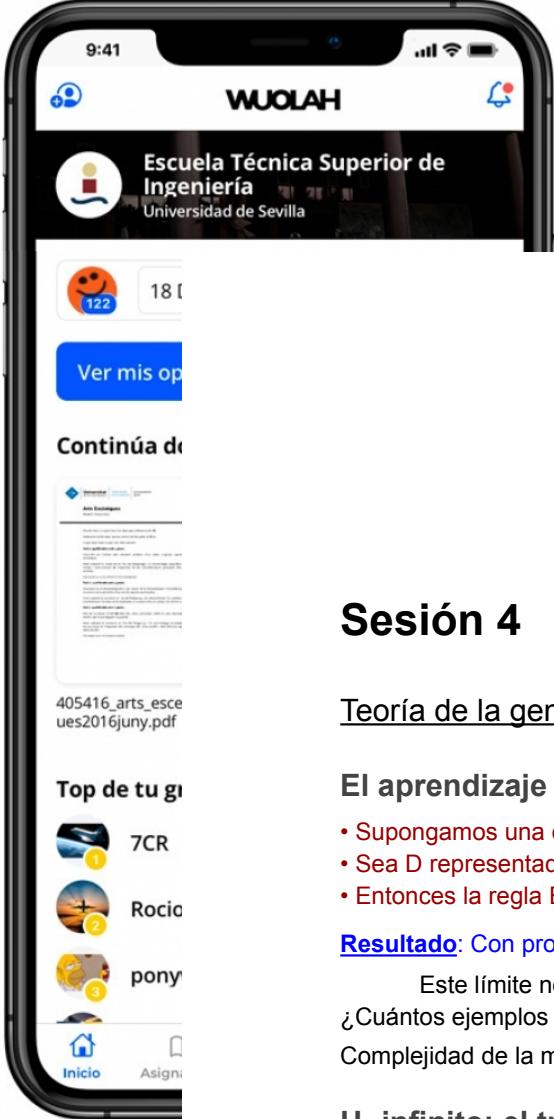
- f simple -> puede usar una H pequeña para obtener $E_{in} \approx 0$ (necesita una N más pequeña).
- f complejo -> necesita una H grande para obtener $E_{in} \approx 0$ (necesita una N más grande).

Viabilidad del aprendizaje (H finito): Resumen

- Fuera de D , no se puede garantizar nada sobre f
- Si D es una muestra independiente de $P(x)$.
 $E_{out} \approx E_{in}$ (E_{in} puede llegar fuera de los datos establecidos en E_{out}).
- Pero, lo que queremos es $E_{out} \approx 0$.
- La solución de dos pasos. Intercambiamos $E \approx 0$ por 2 goles:
 - (i) $E_{out} \approx E_{in}$
 - (ii) $E_{in} \approx 0$.

Conocemos E_{in} , no E_{out} , pero podemos asegurar (i) si $|H|$ es pequeño.

Cualquier regla de ERM es un aprendiz exitoso de PAC para clases finitas H



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.

Available on the
App Store

GET IT ON
Google Play

Sesión 4

Teoría de la generalización ERM: La dimensión Vapnik-Chervonenkis

El aprendizaje es factible para H finito

- Supongamos una distribución de probabilidad $P(x)$ en X
- Sea D representado como una muestra i.i.d de $P(x)$
- Entonces la regla ERM nos da:

Resultado: Con probabilidad de al menos $1-\delta$, $E_{\text{out}}(g) \leq E_{\text{in}}(g) + \sqrt{\frac{1}{2N} \ln \frac{2|H|}{\delta}}$

Este límite no depende de X , $P(x)$, f ni de cómo se encuentra g .

¿Cuántos ejemplos se requieren para garantizar una convergencia uniforme?

Complejidad de la muestra: $N(\epsilon, \delta, H) \geq [\frac{1}{\epsilon^2} \ln \frac{2|H|}{\delta}] = O(\frac{\ln |H|}{\epsilon^2})$

H -infinito: el truco de la discretización

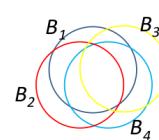
- El truco de discretización nos permite tener una estimación de la desigualdad de complejidad de la muestra en clases infinitas
 - Ejemplo:
 - Una computadora moderna usa una representación de 64 bits para cada escalar.
 - Ya sea que tengamos que ajustar funciones con un solo parámetro libre, solo tenemos 2 64 valores posibles
 - El tamaño de H ahora es 2 64
 - En el caso de d parámetros libres el tamaño será 2 64d
 - Aplicando la desigualdad para clases finitas, obtenemos un límite para la complejidad de la muestra dada por
- $$m_H(\epsilon, \delta) \leq [\frac{1}{\epsilon^2} \log \frac{2|H|}{\delta}] = \frac{64d + 2\log(2/\delta)}{\epsilon^2}$$
- Este límite nos permite obtener una estimación muy aproximada de la complejidad de la muestra requerida en situaciones prácticas
 - Hay algo mejor...?

¿Cuál es el problema de la desigualdad uniforme?

- Recordemos el límite simple que usamos: $P(\bigcup_{i=1:|H|} B_i) \leq \sum_{i=1}^{|H|} P(B_i)$
- y su consecuencia $P(D: |E_{\text{in}}(g) - E_{\text{out}}(g)| > \epsilon) < 2|H|e^{-2\epsilon^2 N}$ para todo $\epsilon > 0$
- Pero en la mayoría de los casos, $B_i \cap B_j \neq \emptyset$ para casi todos (i, j) , por lo tanto

$$\bigcup_{i=1:|H|} B_i = \bigcup_{j=1:|H|} B_j \quad |V| \leq |H|$$

- ¡Esto significa que contar solo unas pocas hipótesis podría ser suficiente!
- Se necesita un mejor límite para el número efectivo de hipótesis en H
- ¡La dimensión Vapnik-Chervonenkis es la respuesta!

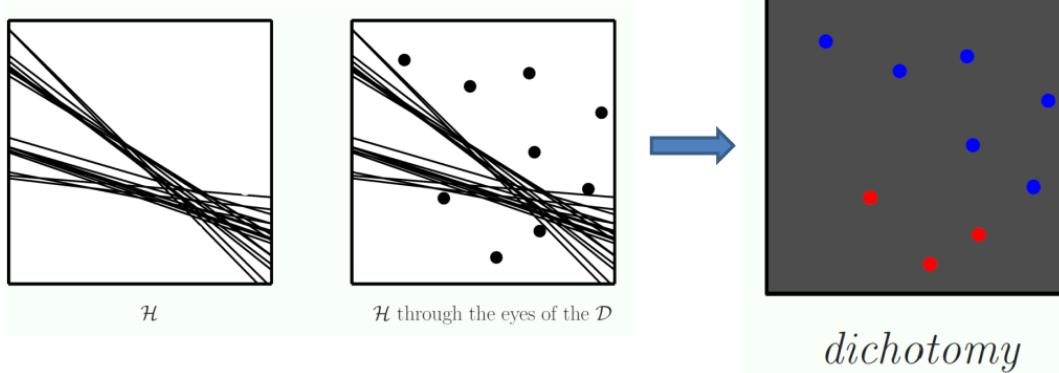


VC Generalización Bound

- El límite de generalización de VC es $E_{\text{out}}(h) \leq E_{\text{in}}(h) + \sqrt{\frac{8}{N} \log \frac{4((2N)^{d_{\text{vc}}} + 1)}{\delta}}$
- o equivalente $E_{\text{out}}(h) \leq E_{\text{in}}(h) + O(\sqrt{d_{\text{vc}} \frac{\log N}{N}})$
- Esto muestra que para d_{vc} finito y $N >> 0$, la generalización está garantizada
- Como conclusión, cualquier modelo puede considerarse bien modelo o modelo desconocido.
 - Buenos modelos: podemos obtener una buena generalización
 - Modelos desconocidos: d_{vc} es infinito (no hay respuesta en la teoría de VC)

Midiendo la diversidad de H

- Necesitamos una forma de medir la diversidad de H
- Aquí nos centramos en funciones de destino binarias $\{-1, +1\}$ y muestra finita de puntos
- El enfoque es combinatorio:
 - Considere una muestra de tamaño fijo N
 - Explore si H puede implementar TODAS las funciones posibles (etiquetado) en ESTOS N puntos
 - Evaluar para todos los valores de N



La función de crecimiento

Ese es el número efectivo de funciones en la clase.

- La función de crecimiento m_H : dado un tamaño de muestra N y una clase H , m_H devuelve el **número máximo de patrones binarios** generados por H en N puntos.

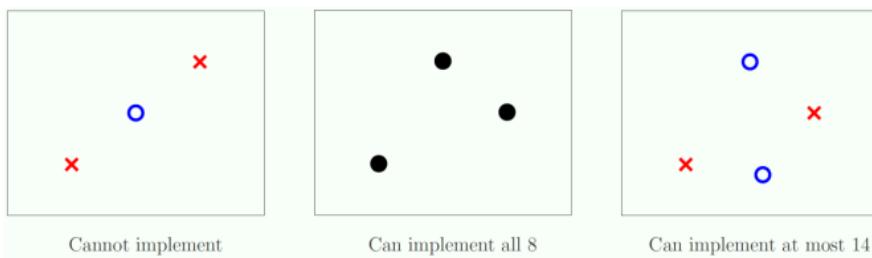
$$m_H(N) = \max_{x_1, \dots, x_N} |H(x_1, x_2, \dots, x_N)|$$

donde $| |$ representa el número de elementos del conjunto

- El **máximo** se calcula en todas las muestras posibles de tamaño N
- En general $m_H(N) \leq 2^N$
- Cuando $m_H(N) = 2^N$ decimos que **H rompe el conjunto** $\{x_1, x_2, \dots, x_N\}$
- Es independiente de P y, por lo tanto, es un análisis del peor de los casos.

Función de crecimiento: ejemplo 1

- Sea H la clase de perceptrón 2D



- ¿Cuál es el valor de $m_H(N)$?

$$m_H(2) = 4 = 2^2$$

$$m_H(3) = 8 = 2^3$$

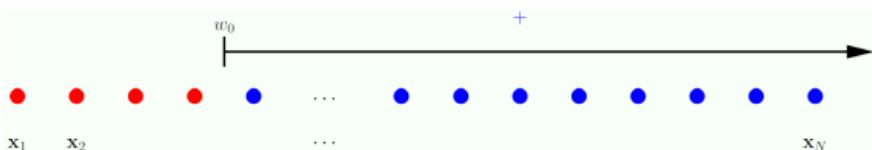
$$m_H(4) = 14 < 2^4$$

- Sea H la clase de perceptrón (predictores lineales binarios) y $X = \mathbb{R}^3$

- ¿Se puede hacer añicos una muestra de 2 puntos? , ¿3 puntos ? , ¿4 puntos ? , etc.

- ¿Puedes adivinar alguna regla para los puntos en \mathbb{R}^k ?

Función de crecimiento: ejemplo 2



- Sea H la clase de $h: \mathbb{R} \rightarrow \{-1, +1\}$ (Rayos positivos)

$$h(x) = \text{signo}(x-w_0)$$

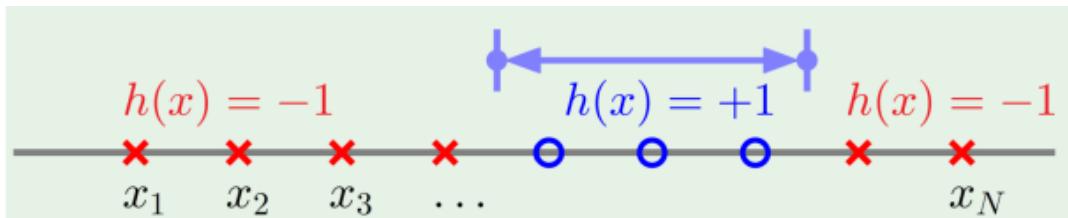
- Consideraremos una muestra de N puntos de \mathbb{R} .

- Pregunta: ¿Cuál es el valor de $m_H(N)$?

- Respuesta: $\{N+1, N, N-1\}$, ¿cuál de ellos?

- ¿Cuántos puntos se pueden romper?

Función de crecimiento: ejemplo 3



- Sea H la clase de funciones $h: \mathbb{R} \rightarrow \{-1, +1\}$ (Intervalos)

$$h_{a,b}(x) = \begin{cases} +1 & \text{si } x \in [a,b] \\ -1 & \text{si } x \notin [a,b] \end{cases}$$

- Consideraremos una muestra de N puntos de \mathbb{R} .

$$\bullet \text{ Ahora: } m_H(N) = \binom{N+1}{2} = \frac{1}{2}N^2 + \frac{1}{2}N + 1 \quad \text{¿Por qué?}$$

- ¿Cuántos puntos se pueden romper?

Reservados todos los derechos.
No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

Compra Coins y descarga sin publicidad.

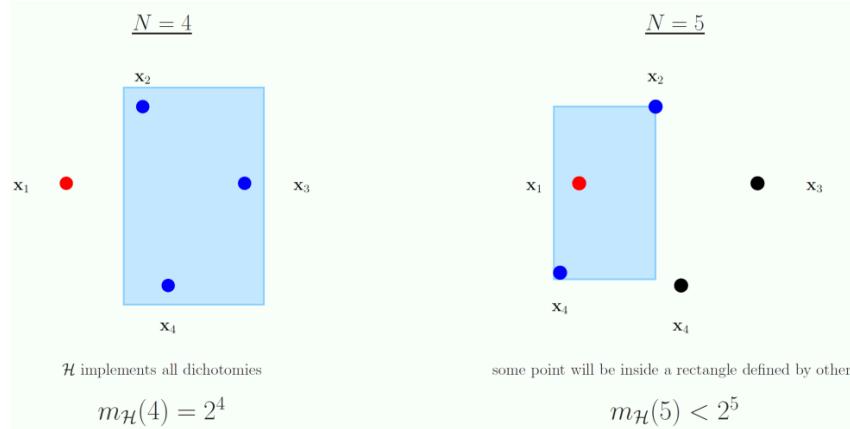
Estudiar sin publi es posible.

Compra Wuolah Coins y que nada te distraiga durante el estudio.



Función de crecimiento: ejemplo 4

- Sea H la clase de rectángulos positivos

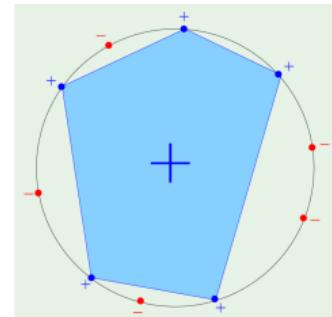


¡Calcular $m_H(5)$ NO es fácil!

Función de crecimiento: ejemplo 5

- Sea H la clase de conjunto convexo
- Considere el caso en el que todos los puntos se encuentran en un círculo.

\mathcal{H} is set of $h: \mathbb{R}^2 \rightarrow \{-1, +1\}$
 $h(\mathbf{x}) = +1$ is convex
 $m_{\mathcal{H}}(N) = 2^N$



Función de crecimiento y generalización

- Echemos un vistazo al nuevo límite que obtenemos:

$$P(D: |E_{in}(h) - E_{out}(h)| > \epsilon) \leq 4m_{\mathcal{H}}(2N)e^{-\frac{N\epsilon^2}{8}}$$

$$E_{out}(h) \leq E_{in}(h) + \sqrt{\frac{8}{N} \log \frac{4m_{\mathcal{H}}(2N)}{\delta}}$$

- Esta expresión proporciona una cota mejor pero necesaria para calcular la función de crecimiento.
- Un límite superior constante en $m_{\mathcal{H}}(N)$ resolverá el problema
- El nuevo límite **no es** un reemplazo directo de $|H|$ por $m_{\mathcal{H}}(N)$!!



WUOLAH

Punto de ruptura

- No es práctico tratar de calcular $m_H(N)$ para cada conjunto de hipótesis que usamos, un límite superior será suficiente.
- Concepto de punto de ruptura:
 - si para algún valor k , $m_H(N)k < 2^k$, entonces k es un **punto de ruptura** para H
 - **Es decir, H NO PUEDE romper una muestra de tamaño k**
- **Ejemplos:**
 - ¿Cuál es el punto de quiebre del perceptrón 2D? $k = 4$
 - ¿Cuál es el punto de quiebre de los Positive Rays? $k = 2$
 - ¿Cuál es el punto de quiebre del intervalo? $k = 3$
 - ¿Cuál es el punto de ruptura del rectángulo positivo? $k = 5$
 - ¿Cuál es el punto de quiebre del conjunto convexo? $k = \infty$

Limitando la función de crecimiento

- **Resultado principal (Vapnik y Chervonenkis, 1971):** si k es un punto de quiebre para H , entonces para todos los N

$$m_H(N) \leq \sum_{i=0}^{k-1} \binom{N}{i}$$

El RHS **es polinomio en N** de grado $k-1$: $O(N^{k-1})$

Este resultado dice: **si H tiene un punto de ruptura: $m_H(N)$ es polinomial en N**

si H NO tiene punto de ruptura: $m_H(N) = 2^N$

- Con respecto al límite de generalización, reemplazamos $\log(m_H(N))$ por $O(k \log(N))$.
 - Para $N \gg 0$ podemos garantizar una buena generalización ya que $\log(N) / N \rightarrow 0$
- ¿Qué sucede en el límite de generalización cuando **H NO tiene un punto de ruptura?**

Vapnik&Chervonenkis: dimensión VC:

- **Definición:** La dimensión VC de un conjunto de hipótesis H , denotado por $d_{VC}(H)$ o simplemente d_{VC} , es el valor más grande de N para el cual $m_H(N) = 2^N$. Si $m_H(N) = 2^N$ para todo N , entonces $d_{VC} = \infty$.
- Se puede demostrar que el **límite principal** se puede escribir como

$$m_H(N) \leq \sum_{i=0}^{k-1} \binom{N}{i} \leq N^{d_{VC}+1} \\ \left(\frac{eN}{d_{VC}}\right)^{d_{VC}}$$

Dos límites para la función de crecimiento

- El valor d_{VC} mide el número "**efectivo**" de parámetros asociados con $h \in H$ (Perceptrón 2D, $d_{VC} = 3$; en **modelos lineales** $d_{VC} = d+1$)

VC Generalization Bound

- Combinando límites

$$E_{\text{out}}(h) \leq E_{\text{in}}(h) + \sqrt{\frac{8}{N} \log \frac{4((2N)^{d_{\text{VC}}} + 1)}{\delta}}$$

o equivalente

$$E_{\text{out}}(h) \leq E_{\text{in}}(h) + O(\sqrt{d_{\text{VC}} \frac{\log N}{N}})$$

- Esto muestra que para d_{VC} finito y $N >> 0$, la generalización está garantizada
- Una conclusión de estos resultados es que hay una división de modelos en dos clases: modelos "buenos" y modelos "inútiles" (en términos de aprendizaje ERM)
 - Modelos "buenos": d_{VC} es finito podemos obtener una buena generalización
 - Modelos "inútiles": d_{VC} es infinito (¡¡no podemos aprender usando la regla ERM !!)

Complejidad de la muestra

- **Recuerde:** la complejidad de la muestra es la cantidad mínima de ejemplos de entrenamiento (N) necesarios para lograr un cierto rendimiento de generalización
 - ϵ, δ tienen que ser fijos
 - La rapidez con que crece $N(\epsilon, \delta)$ indica cuántos datos se necesitan para obtener una buena generalización.
- Fije $\delta > 0$ y suponga que el error de generalización es como máximo

$$\sqrt{\frac{8}{N} \log \frac{4m_H(2N)}{\delta}} \leq \epsilon \Rightarrow N \geq \frac{8}{\epsilon^2} \ln \left(\frac{4m_H(2N)}{\delta} \right) \Rightarrow N \geq \frac{8}{\epsilon^2} \ln \left(\frac{4((2N)^{d_{\text{VC}}} + 1)}{\delta} \right)$$

- Esta es una ecuación implícita en N , la resolvemos iterativamente (ejemplo en diapositiva 20 T4)

VC significa penalización por complejidad del modelo

- En la mayoría de situaciones prácticas, se proporciona el conjunto de datos de muestra, por lo que N es fijo.
- La pregunta relevante ahora es qué rendimiento podemos esperar dado este N particular

$$E_{\text{out}}(g) \leq E_{\text{in}}(g) + \sqrt{\frac{8}{N} \log \frac{4m_H(2N)}{\delta}} \leq E_{\text{in}}(g) + \sqrt{\frac{8}{N} \ln \left(\frac{4((2N)^{d_{\text{VC}}} + 1)}{\delta} \right)}$$
$$\Omega(N, H, \delta) = \sqrt{\frac{8}{N} \ln \left(\frac{4((2N)^{d_{\text{VC}}} + 1)}{\delta} \right)} = O\left(\sqrt{\frac{d_{\text{VC}} \ln N - \ln \delta}{N}}\right)$$

- Este término puede verse como una **penalización debido a la complejidad H** .

$$E_{\text{out}} \leq E_{\text{in}} + \Omega(d_{\text{VC}})$$

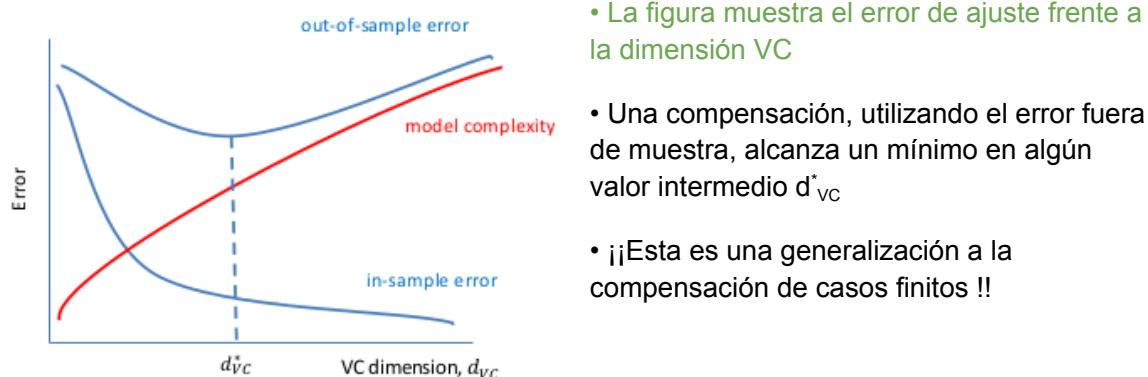
Aproximación-Generalización Tradeoff: una nueva perspectiva para elegir g

VC Bound cuantifica la aproximación frente a la generalización

- De hecho, tenemos una compensación: los modelos más complejos ayudan a E_{in} y perjudican a $\Omega(N, H, \delta)$
- $d_{vc} \uparrow \Rightarrow$ mayor probabilidad de aproximar f ($E_{in} \approx 0$).
- $d_{vc} \downarrow \Rightarrow$ mayor probabilidad de generalizar fuera de muestra ($E_{in} \approx E_{out}$).

$$E_{out} \leq E_{in} + \Omega(d_{vc})$$
- El análisis de VC solo depende de H.**
 - Independiente de f, P(X, A (algoritmo de aprendizaje))
 - Aplicable principalmente a problemas de clasificación y regresión.
 - Sin embargo, para la pérdida de cuadratura, la compensación B-V proporciona una mejor percepción.
 - Atado bastante suelto

Penalización por complejidad del modelo

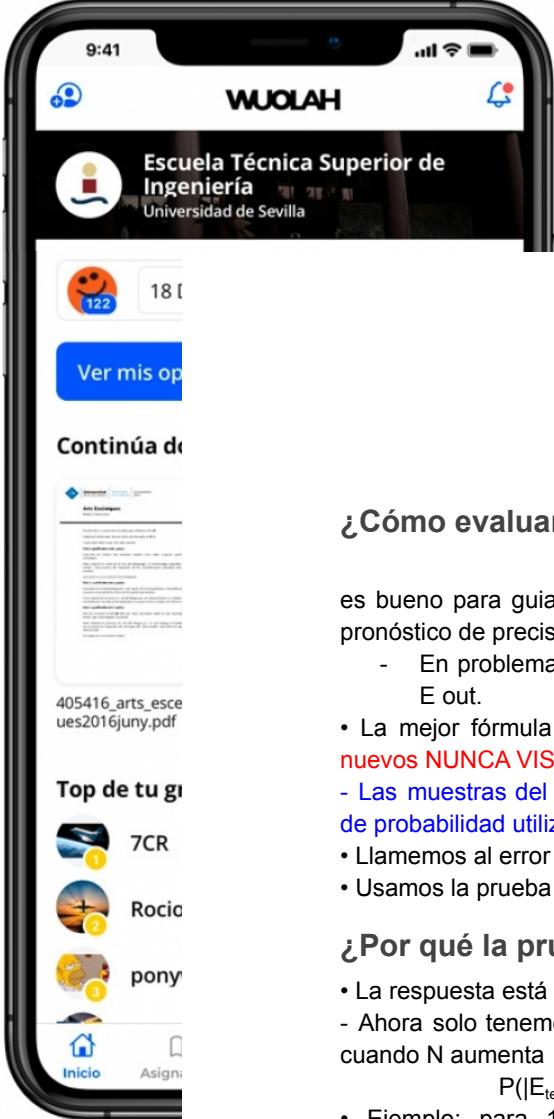


Complejidad del modelo (d_{vc}) $\uparrow \rightarrow E_{in} \downarrow \rightarrow$ mayor probabilidad de aproximar f

Complejidad del modelo (d_{vc}) $\downarrow \rightarrow E_{out} - E_{in} \downarrow \rightarrow$ mayor probabilidad de una buena generalización

Resumen del VC Bound

- Si $d_{vc}(H)$ es finito \Rightarrow La clase H se puede aprender con "PAC"
- El límite de VC es independiente de: f, $\mathbb{P}(X)$, A
 - (Función de destino binario, distribución de entrada, algoritmo de aprendizaje)
- La dimensión VC nos da una medida de la complejidad de la clase H
 - Cuanto mayor sea la complejidad, mayor será el conjunto de entrenamiento para un error fijo
- La dimensión VC de una clase H está relacionada con el número "efectivo" de parámetros libres de sus elementos.
- El análisis de VC se desarrolló para una función de pérdida 0-1 (clasificación)
 - Pero puede extenderse a funciones de pérdida de valor real (regresión)



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.

Available on the
App Store

GET IT ON
Google Play

¿Cómo evaluar nuestro ajuste?

$$E_{out} \leq E_{in} + \Omega(N, H, \delta)$$

es bueno para guiar el proceso de entrenamiento PERO es inútil si queremos obtener un pronóstico de precisión de E_{out} .

- En problemas reales, lo que el cliente espera obtener es una estimación precisa de E_{out} .
- La mejor fórmula es desafiar nuestra hipótesis entrenada con ejemplos **absolutamente nuevos NUNCA VISTOS ANTES**. A esto se le llama **SET DE PRUEBA**
- Las muestras del conjunto de prueba **DEBEN** ser muestras i.i.d. de la misma distribución de probabilidad utilizada en el entrenamiento.
- Llamemos al error en el equipo de prueba E_{test} .
- Usamos la prueba E_{test} como estimador de E_{out}

¿Por qué la prueba E debería ser un buen estimador de E_{out} ?

- La respuesta está en la desigualdad de Hoeffding simple
 - Ahora solo tenemos una hipótesis y la desigualdad de Hoeffding es mucho más ajustada cuando N aumenta
- $$P(|E_{test}(g) - E_{out}(g)| > \epsilon) \leq 2e^{-2Ne^2}$$
- Ejemplo: para 1000 ejemplos de prueba, E_{test} estará dentro del 5% de E_{out} con probabilidad $\geq 98\%$
 - Además, la estimación del conjunto de pruebas no está sesgada. Esto significa independiente de E_{in}
 - Pero nada es gratis, hay que pagar un precio por usar un equipo de prueba
 - Perdimos datos de entrenamiento Mayor error en la muestra

Discusión NLT

- ¿Cómo afecta la transformación de características a PLA VC-bound?
- Si honestamente arreglamos la transformación antes de ver los datos, entonces $d_{VC}(H_\Phi) = d_{VC}(H)$ al menos con probabilidad $1-\delta$
- ¿Qué pasa si primero intentamos separar con líneas, fallamos y luego usamos los círculos?
 - Esto es equivalente a usar una transformación donde se conservan las características originales y agregamos el cuadrado de todas ellas.
 - ¡Hemos aumentado la dimensión del espacio de funciones!
- ¿Qué pasa si exploramos los datos pero no probamos ningún modelo?
 - Aún peor !! Nuestra mente ha explorado un enorme espacio de hipótesis que debemos agregar a la dimensión de las transformaciones reales.
 - ¡¡Sin darte cuenta, has decidido que tus datos son el problema y no una muestra !!
- En los problemas de clasificación, si insistimos en obtener una separabilidad total entre clases, podemos vernos obligados a utilizar transformaciones de alto grado.
 - Sin embargo, esto aumenta drásticamente la dimensión del espacio de características y la dimensión de VC
- Analicemos con más detalle estas implicaciones

Computación y generalización

- Denotemos por Q la transformada polinomial de Q -ésimo orden
 - $\Phi_4(x) = (1, x_1, x_2, x_1^2, x_2^2, x_1x_2, x_1^3, x_2^3, x_1x_2^2, x_2x_1^2, x_1^4, x_2^4, x_1^2x_2^2, x_2^1x_1^3, x_1^1x_2^3)$
- Una Q más grande proporciona una mayor flexibilidad en términos de la forma del límite de decisión, pero hay un precio que pagar.
 1. El cálculo es un problema porque la transformación de características Φ_Q asigna x (el vector inicial) a $d = \frac{Q(Q+3)}{2}$ dimensiones, lo que aumenta la memoria y el costo computacional.
 2. La dimensión VC puede aumentar hasta $\frac{Q(Q+3)}{2} + 1$ y el límite VC puede crecer significativamente
 - Para $Q = 50$ el VC-dim es $\frac{Q(Q+3)}{2} + 1 = 1326$ en lugar de 3 (inicial)
 3. De acuerdo con la regla: “.. el número de muestras necesarias es proporcional al VC-dim”, cuanto mayor sea el valor Q , mayor (orden cuadrático) será el número de muestras que necesitaremos para obtener el mismo nivel de error de generalización.
- En general, al elegir la dimensión adecuada para la transformación de características, debemos utilizar una compensación de aproximación-generalización:
 - mayor d , mayor probabilidad de ser linealmente separables $E_{in} \downarrow$ y $E_{out} \uparrow$
 - menor d , E_{in} y E_{out} posiblemente no separables linealmente \downarrow

¿Qué sucede cuando $d_{VC} = \infty$?

- APRENDIZAJE UNIFORME: A partir del análisis de la dimensión VC sabemos que la regla ERM es una regla general de aprendizaje para d_{vc} finito
- APRENDIZAJE NO UNIFORME
- Ahora consideramos $H = U_n H_n$, $d_{vc}(H_n) < \infty$, $n = 1, 2, 3, \dots$
 - Esto significa una clase con una dimensión de VC infinita pero definida como la unión de un número infinito de clases, cada una con $d_{vc} < \infty$
- Ejemplo:
 - Clase de todos los polinomios en R . $H = U_n H_n$ donde H_n representa la clase de los polinomios de grado n . No es difícil demostrar que $VCdim(H) = \infty$ y $VCdim(H_n) = n+1$

Regla de aprendizaje no uniforme: SRM

$$\Omega(N, H, \delta) = O\left(\sqrt{\frac{d_{vc} \ln N - \ln \delta}{N}}\right)$$

- Que pasa cuando $\frac{N}{d_{vc}} < 20$?
 - pequeño número de muestras con respecto al número de parámetros efectivos.
- En este caso, la regla ERM no es una garantía para el aprendizaje.

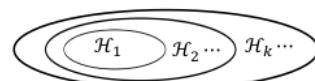
Se introduce una nueva regla de inducción: **Minimización del RIESGO estructural (SRM)**

$$g^* = \arg \min_{i=1,2,\dots} E_{in}(g_i) + \Omega(H_i)$$

$$d_{vc}(H_1) \leq d_{vc}(H_2) \leq \dots \leq d_{vc}(H_k) \leq \dots$$

SMR

1. Seleccione una secuencia anidada de conjunto de hipótesis
2. Estima g de cada conjunto de la secuencia.



Criterios de implementación de SRM

- Mantiene fija la complejidad del modelo y minimiza el error empírico
- Mantiene el error empírico constante (pequeño) y minimiza la dimensión de VC

Válido para enfoques que minimizan el verdadero error en lugar de empíricos

Another understanding for E_{out}

Compensación de sesgo-varianza

- Descomposición BIAS-VARIANCE

$$E_{\text{out}}(g^{(D)}) = E_x[(g^{(D)}(\mathbf{x}) - f(\mathbf{x}))^2]$$

- E_x denota el valor esperado con respecto a \mathbf{x} (basado en $P(X)$)
- **Ese es el error cuadrático medio (MSE) de $g^{(D)}$**

- El análisis de la varianza de sesgo se basa en la medida de los errores cuadrados, pero se aplica a la clasificación y la regresión.

• El análisis de la varianza de sesgo tiene en cuenta H y A

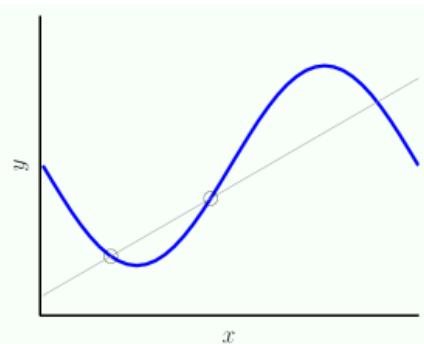
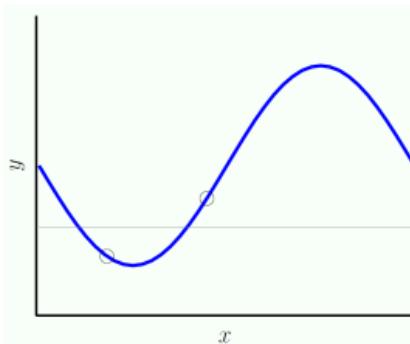
- ¡Diferentes algoritmos de aprendizaje A pueden tener diferentes E_{out} cuando se aplican a la misma H !!

Un simple problema de aprendizaje

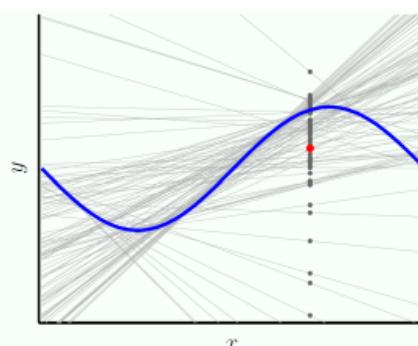
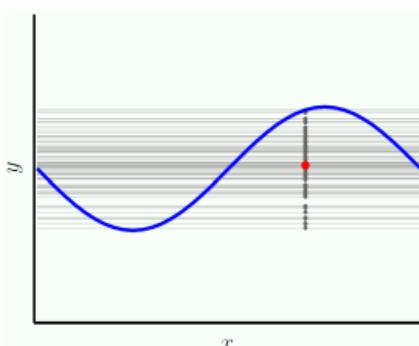
- 2 puntos de datos. 2 conjuntos de hipótesis

• $H_0: h(x) = b$

• $H_1: h(x) = ax + b$



Repita el experimento varias veces ...



- Para cada conjunto de datos D , obtiene un g^D diferente.
- Entonces, para una \mathbf{x} fija, $g^D(\mathbf{x})$ es un valor aleatorio, dependiendo de D .

9:41

WUOLAH

Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

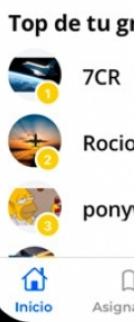
Available on the App Store GET IT ON Google Play

Descarga la APP de Wuolah.

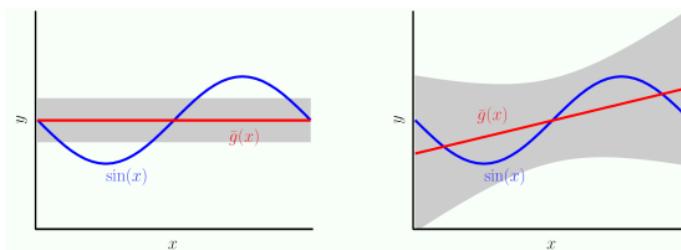
Ya disponible para el móvil y la tablet.



Continúa d



¿Qué está pasando en promedio?



Podemos definir

$$g^D(x) \quad \leftarrow \text{valor aleatorio, según D}$$

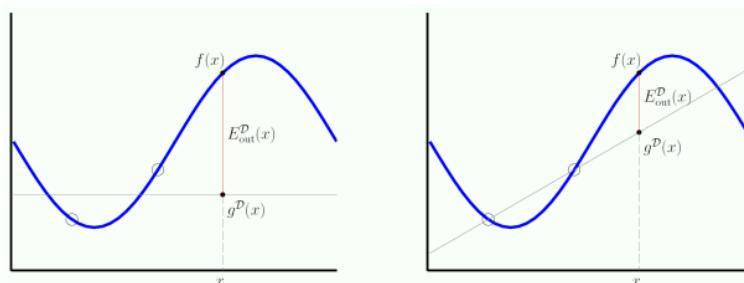
$$\bar{g}(x) = E_D[g^D(x)]$$

$$\approx \frac{1}{K} (g^1(x) + g^2(x) + \dots + g^K(x)) \quad \leftarrow \text{la predicción promedio en } x$$

$$\text{var}(x) = E_D[(g^D(x) - \bar{g}(x))^2]$$

$$= E_D[g^D(x)] - \bar{g}(x)^2 \quad \leftarrow \text{¿Qué tan variable es la predicción?}$$

E en el punto de prueba x para los datos D



$$E^D_{\text{out}}(x) = (g^D(x) - f(x))^2 \quad \leftarrow \text{error al cuadrado, un valor aleatorio que depende de D}$$

$$E_{\text{out}}(x) = E_D[E^D_{\text{out}}(x)] \quad \leftarrow \text{se esperaba } E_{\text{out}}(x) \text{ antes de ver D}$$

Compensación de sesgo-varianza

- Para obtener una estimación del error MSE independiente de D

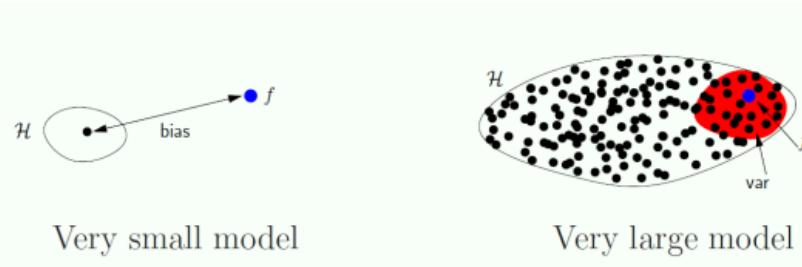
$$E_D[E_{\text{out}}(g^{(D)})] = E_D[E_x[(g^{(D)}(x) - f(x))^2]] = E_x[E_D[(g^{(D)}(x) - f(x))^2]] \\ E_D[(g^{(D)}(x) - f(x))^2] = E_D(g^{(D)}(x)^2) - 2E_D(g^{(D)}(x)f(x)) + f(x)^2$$

- El término $E_D(g^{(D)}(x))$ da una **función promedio** que denotamos por $\hat{g}(x)$

$$E_D[E_{\text{out}}(g^{(D)})] = E_x[E_D(g^{(D)}(x)^2) - 2\hat{g}(x)f(x) + f(x)^2] \\ = E_x[E_D(g^{(d)}(x)^2) - \hat{g}(x)^2 + \hat{g}(x)^2 - 2\hat{g}(x)f(x)+f(x)^2] \\ \text{varianza}(x) \quad (\hat{g}(x)-f(x))^2 \\ \text{sesgo}(x)$$

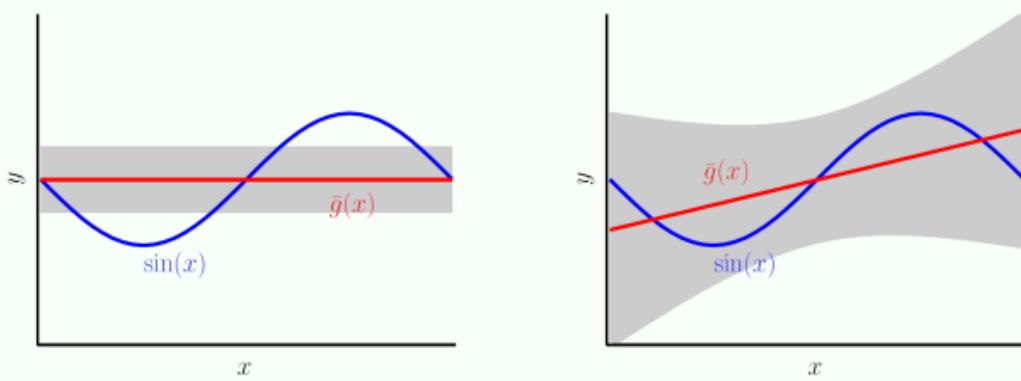
$$E_D[E_{\text{out}}(g^{(d)})] = E_x[\text{sesgo}(x) + \text{varianza}(x)] = \text{sesgo} + \text{varianza}$$

Compensación entre sesgo y varianza: comentarios



- $E_D[E_{\text{out}}(g^{(D)})] = \sigma^2 + \text{sesgo} + \text{varianza}$ (para señales ruidosas)
 - σ^2 es la varianza del ruido
 - El ruido es inevitable sin importar lo que hagamos, por lo que nuestro interés permanece en el sesgo y la varianza.
 - Desafortunadamente, es imposible calcular el sesgo y la varianza. Por tanto, **la descomposición sesgo-varianza es una herramienta conceptual que resulta útil a la hora de desarrollar un modelo.**
- Hay dos objetivos típicos cuando consideramos el sesgo y la varianza:
 - Disminuir la varianza sin aumentar significativamente el sesgo (1)
 - Disminuir el sesgo sin aumentar significativamente la varianza (2)
- **Estos objetivos se logran mediante diferentes técnicas: Regularización (1), conocimientos previos (2)**

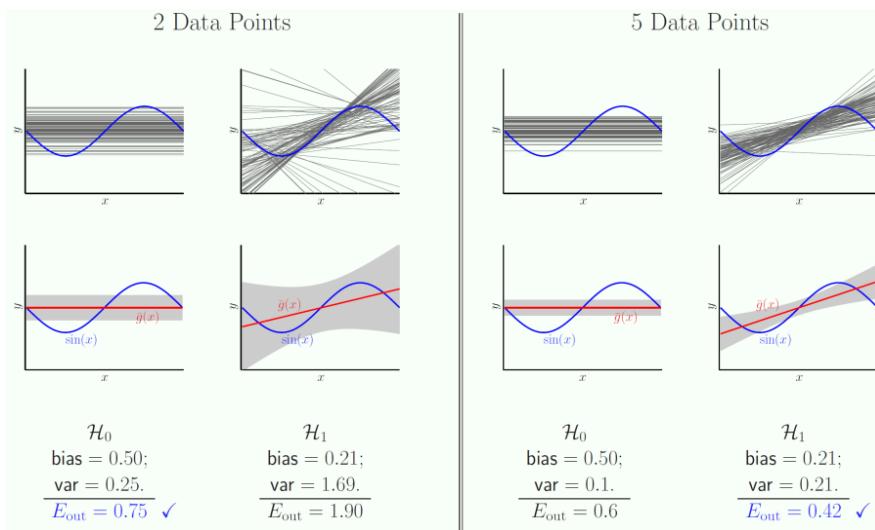
De vuelta a H_0 y H_1 ; y nuestro ganador es...



$$\begin{aligned}
 \mathcal{H}_0 \\
 \text{bias} &= 0.50 \\
 \text{var} &= 0.25 \\
 \frac{E_{\text{out}} = 0.75}{\checkmark}
 \end{aligned}$$

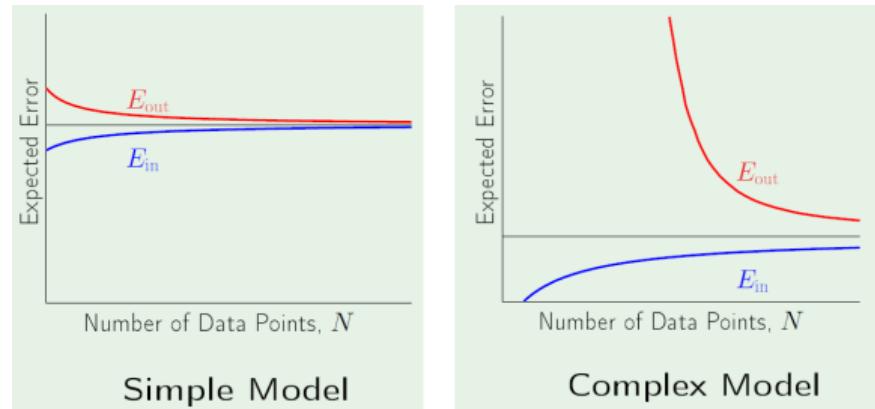
$$\begin{aligned}
 \mathcal{H}_1 \\
 \text{bias} &= 0.21 \\
 \text{var} &= 1.69 \\
 \frac{E_{\text{out}} = 1.90}{}
 \end{aligned}$$

Haga coincidir el poder de aprendizaje con los datos. . .No a la memoria.



Curva de aprendizaje

- Las curvas de aprendizaje resumen el comportamiento de los errores $E_D[E_{in}(g^{(D)})]$ y $E_D[E_{out}(g^{(D)})]$ cuando **variamos el tamaño N del conjunto de entrenamiento**.



La complejidad del modelo influye en el Error Esperado y la velocidad de convergencia
Izquierda: polinomio de segundo orden Derecha. Polinomio de décimo orden

Curvas de aprendizaje para regresión lineal

Solución de regresión lineal: $\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$

Vector de error en la muestra = $\mathbf{X}\mathbf{w} - \mathbf{y}$

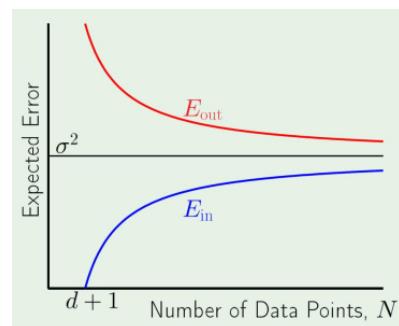
vector de error 'fuera de muestra' = $\mathbf{X}\mathbf{w} - \mathbf{y}$

Mejor error de aproximación = σ^2

Error esperado en la muestra = $\sigma^2(1 - \frac{d+1}{N})$

Error esperado fuera de la muestra = $\sigma^2(1 + \frac{d+1}{N})$

Error de generalización de expectativas = $2\sigma^2 \left(\frac{d+1}{N}\right)$



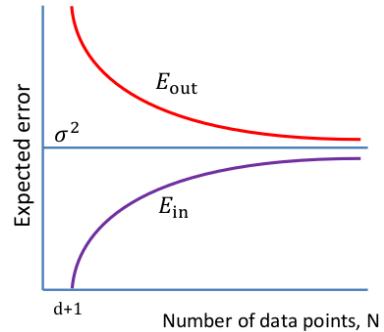
Curva de aprendizaje para regresión lineal

- Consideremos ahora la expresión para los valores esperados de $E_{in}(\mathbf{w}_{lin})$ y $E_{out}(\mathbf{w}_{lin})$

$$E_D[E_{in}(\mathbf{w}_{lin})] = \sigma^2(1 - \frac{d+1}{N}), \text{ para } N \geq d+1$$

$$E_D[E_{test}(\mathbf{w}_{lin})] = \sigma^2(1 - \frac{d+1}{N}) \text{ (approx. a } E_{out} \text{)}$$

La figura muestra la curva de aprendizaje de regresión lineal bajo los supuestos de OLS.



- E_{in} : Cuando N aumenta, el modelo absorbe tanta información como sea posible con los parámetros $d+1$
- E_{out} : Cuando N aumenta, el error fuera de muestra del modelo disminuye al ruido residual.
- Este comportamiento de la curva de aprendizaje es el esperado cuando se ha elegido el modelo de complejidad adecuado.