

Tópicos de investigación CM072

César Lara Avila

5 de septiembre de 2017

<https://github.com/C-Lara>

1. Introducción al Machine Learning

Acerca del curso:

- **Página inicial al curso**
 - https://c-lara.github.io/Analisis_datos_Python/.
- **Repositorio en Github y página web**
 - <https://github.com/C-Lara/M-L>.
 - <http://c-lara.github.io/M-L/>.
- **Horarios:**
 - Miércoles de 16-20pm Sala1.
- **Evaluación:**
 - En el curso de tomaran 6 asignaciones.
 - Tareas de reforzamiento.
 - Proyecto parcial y final del curso.
 - En la página web del curso se indica información de las evaluaciones.

Prerequisitos

Requerido:

- Algoritmos básicos
 - Análisis de algoritmos, programación dinámica.
 - Nociones de análisis de datos.

Muy recomendado:

- Álgebra Lineal
 - Matrices, vectores, sistema de ecuaciones lineales.
 - Autovalores y autovectores, rango de una matriz.
 - Valores singulares. Descomposiciones matriciales.
- Cálculo multivariado
 - Derivación, integración, plano tangente.
 - Optimización y multiplicadores de Lagrange.

Libros y referencias

El libro de referencia es el de David Barber, **Bayesian Reasoning and Machine Learning** de Cambridge University Press, 2017.

Otras posibilidades son:

- **Machine Learning: a Probabilistic** de Kevin Murphy (2012).
- **Pattern Recognition and Machine** de Chris Bishop (2006).
- **Machine Learning refined: Foundations, Algorithms, and Applications** 1st Edition Jeremy Watt, Reza Borhani y Aggelos K. Katsaggelos, 2016.
- **Data Science From Scratch: First Principles with Python** de Joel Grus 2015.

2.Ejemplos del Machine Learning

Clasificación: Desde datos a clases discretas

Filtrado de Spam (Bayesiano)

Este filtro está basado en el teorema de Bayes para determinar un correo electrónico como spam o no. De wikipedia

$$\mathbb{P}(S|W) = \frac{\mathbb{P}(W|S) \cdot \mathbb{P}(S)}{\mathbb{P}(W|S) \cdot \mathbb{P}(S) + \mathbb{P}(W|H) \cdot \mathbb{P}(H)}$$

donde:

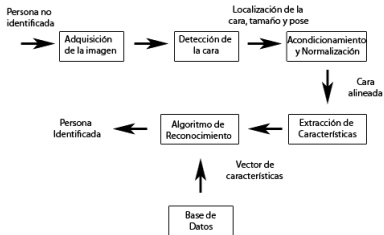
- $\mathbb{P}(S|W)$ es la probabilidad de que un mensaje es spam, a sabiendas de que la palabra a buscar está en el mensaje.
- $\mathbb{P}(S)$ es la probabilidad general de que cualquier mensaje dado es spam.
- $\mathbb{P}(W|S)$ es la probabilidad de que la palabra a buscar aparece en los mensajes de spam.
- $\mathbb{P}(H)$ es la probabilidad general de que cualquier mensaje dado no es spam.
- $\mathbb{P}(W|H)$ es la probabilidad de que la palabra a buscar aparezca en los mensajes legítimos.

Clasificación: Desde datos a clases discretas

Reconocimiento facial

El reconocimiento facial es el proceso de identificación de una o varias personas en imágenes o vídeos mediante análisis y comparación de patrones. Los algoritmos de reconocimiento facial normalmente extraen las características faciales y las comparan con una base de datos para obtener la mejor coincidencia.

Proceso de reconocimiento facial y un ejemplo de imágenes de entrenamiento para cada orientación



Clasificación: Desde datos a clases discretas

Técnicas de reconocimiento facial

El reconocimiento facial aprovecha la visión artificial para extraer información discriminada de imágenes faciales y reconocimiento de patrones o técnicas del machine learning para modelar la apariencia de las caras y para clasificarlas. Por ejemplo:

- SVM
- Árboles de decisión.
- Métodos de ensamblado.
- Redes neuronales profundas.

De wikipedia:

El reconocimiento facial tridimensional, utiliza imágenes $3D$ tanto en el entrenamiento como en el reconocimiento. Esta técnica utiliza sensores en $3D$ para captar información sobre la forma de la cara. Se utiliza un derivado del PCA parcial, llamado, **partial principal components analysis**.

Clasificación: Desde datos a clases discretas

Predicción del tiempo

No existen solamente procedimiento que recurren únicamente a los modelos meteorológicos y climatológicos desarrollados para predecir el tiempo.



Técnicas del machine learning, incluyendo modelos físicos y redes neuronales profundas empleando datos históricos de las distintas variables meteorológicas (presión atmosférica, temperatura, punto de rocío, campo de vientos, etc) y de situaciones meteorológicas anteriores, para con los datos recabados en el momento presente, realizar previsiones con una buena precisión .

Regresión que predice un valor numérico

Bolsa de valores

Una de las formas tanto en condiciones adversas, como en momentos normales en el mercado de predecir hacia donde irá la bolsa así como cada uno de sus valores se llama **análisis técnico**.



El análisis técnico es un estudio de indicadores que se presentan de forma gráfica información básica muy necesaria con el objetivo final de tomar una decisión a corto plazo de compra o venta en esas acciones.

Regresión que predice un valor numérico

Análisis técnico y regresión

Principalmente el análisis técnico se basa en un conjunto de operaciones estadísticas y matemáticas con los precios que van resultando en el mercado, para determinar y detectar situaciones en las tendencias que siguen las cotizaciones de esas acciones a corto y mediano plazo.

- Econometría
- Matemática financiera avanzada.

El mercado se comporta de una manera gráfica tal que el análisis técnico se puede comparar con un conjunto modelo de variables en las que intentamos predecir el comportamiento futuro con una recta de regresión del tipo simple: $Y = a + bX$.

Es una forma muy básica de predicción, pero es la más cercana a una explicación matemática simple de cómo se realizan las predicciones de los futuros valores bursátiles.

Regresión que predice un valor numérico

Predicción del tiempo

La predicción meteorológica ha sido tradicionalmente realizada por modelos físicos de la atmósfera, que son inestables a las perturbaciones, y por lo tanto son imprecisos para grandes períodos de tiempo. Las técnicas del machine learning son más robustas a las perturbaciones, por lo que generan potencialmente pronósticos meteorológicos más precisos durante grandes períodos de tiempo.



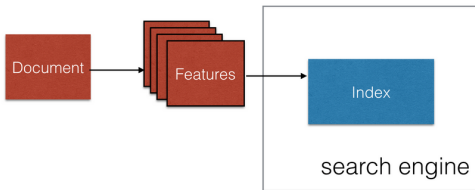
En este contexto, podremos encontrar el valor de la temperatura en determinado escenario, utilizando regresión lineal.

Ranking: Comparación de cosas

¿ Qué es la ingeniería de relevancia?

La **ingeniería de relevancia** es el proceso de identificar las características más importantes del conjunto de documentos para los usuarios de esos documentos y utiliza esas características para afinar el motor de búsqueda para devolver los documentos que mejor se adapten a cada usuario en cada búsqueda.

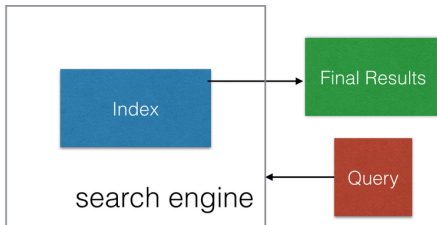
Para recapitular cómo funciona un motor de búsqueda: los documentos en tiempo de índices se analizan en tokens; estos tokens se insertan entonces en un índice como se ve en la figura:



Ranking: Comparación de cosas

Funcionamiento de una máquina de búsqueda

En el tiempo de búsqueda, las consultas individuales también se analizan en tokens. El motor de búsqueda busca los tokens de la consulta en el índice invertido (**inverted index**), clasifica los documentos coincidentes, recupera el texto asociado con esos documentos y devuelve los resultados clasificados al usuario, como se mostrará en la siguiente figura.

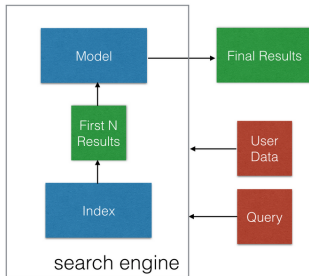


Ranking: Comparación de cosas

LTR

El **LTR** aplica el Machine Learning para construir modelos de clasificación para sistemas de recuperación de información.

Esto significa que en vez de reemplazar el motor de búsqueda por un modelo de Machine Learning, estamos ampliando el proceso con un paso adicional. Después de que se emita la consulta al índice, los mejores resultados de esa consulta se pasan al modelo y se vuelven a ordenar antes de ser devueltos al usuario, como se ve en la siguiente figura:



Ranking: Comparación de cosas

Learning to Rank (LTR) es una clase de técnicas que aplican el machine learning para resolver **problemas de ranking**. La principal diferencia entre el LTR y ML tradicional supervisado es la siguiente:

- El ML tradicional resuelve un problema de predicción (clasificación o regresión) en una sola instancia a la vez. Por ejemplo, si estas realizando detección de spam en el correo electrónico, verá todas las características asociadas con ese correo electrónico y lo clasificará como spam o no. El objetivo del ML tradicional es llegar a una clase (spam o no spam) o una sola puntuación numérica para esa instancia.
- El LTR resuelve un problema de rankings en una lista de elementos. El objetivo de estas técnicas es llegar a un ordenamiento óptimo de esos elementos. Como tal, LTR no le importa mucho acerca de la puntuación exacta que cada elemento obtiene, pero se preocupa más por el orden relativo entre todos los elementos.

Algunos algoritmos LTR son: AdaRank, RankCosine, IntervalRank, etc.

Filtrado colaborativo y basado en contenido

Recomendadores y tipos de filtros

Un **recomendador** es un sistema que selecciona un producto que, si se compra, maximiza el valor tanto para el comprador como para el vendedor en un determinado momento del tiempo.

Los recomendadores habitualmente son de dos tipos: los **filtros colaborativos** y los **filtros basados en contenido**. En este contexto, un filtro es el algoritmo matemático que decide cuál es la recomendación óptima basado en los datos que le entreguemos.

- Los filtros colaborativos generalmente basan su lógica en las características del usuario. Los datos que se tienen del usuario se convierten en el centro de un filtro colaborativo. Ejemplo: reddit, youtube y Last.fm.
- Los filtros basados en contenido tienen el producto como base de la predicción, en lugar de tener al usuario. Es decir, utiliza las características del artículo para hacer las recomendaciones. Ejemplo: IMDb o Rotten Tomatoes.

Algoritmos

- Algoritmos de filtrado colaborativo basados en memoria, o algoritmos de vecinos cercanos (Nearest Neighbour).
- Algoritmos de filtrado colaborativo basados en Modelo. Clustering o redes neuronales como las Redes de Funciones de Base Radial (RBFN).

Un sistema de recomendaciones es mucho más que un algoritmo o un filtro que selecciona productos con más o menos acierto. Podemos dividir un recomendador en 4 partes:

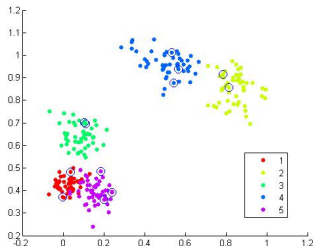
- La base de conocimiento (la información, los datos).
- El procesamiento de la base de conocimientos (tecnología, algoritmos, filtros).
- Analítica y control de negocio (medir todo, estrategia de negocio).
- Interface del usuario.

Agrupamiento: descubrimiento estructuras en los datos

El problema del agrupamiento

El problema del agrupamiento puede definirse como sigue: dados n puntos en un espacio n -dimensional particionar los mismos en k grupos tales que los puntos dentro de un grupo son más similares que cada uno a los de los otros grupos, dicha similaridad se mide atendiendo a alguna función distancia (función de disimilaridad) o alguna función de similaridad.

El conocimiento de los grupos puede permitir una descripción sintética de un conjunto de datos multidimensional complejo.



Agrupamiento: descubrimiento estructuras en los datos

Algoritmos de agrupamiento

Existen varias técnicas para el agrupamiento de casos:

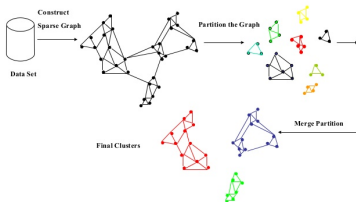
- **Agrupamiento jerárquico**, en los que se va particionando el conjunto de datos por niveles, de modo tal que en cada nivel generalmente , se unen o se dividen dos grupos del nivel anterior , según si es un algoritmo aglomerativo o divisivo.

Las estrategias jerárquicas más conocidas son:

- **Single Link (SL)** En cada paso se unen los dos grupos cuyos elementos más cercanos tienen la mínima distancia.
- **Average Link (AL)** En cada paso se unen los dos grupos tal que tienen la mínima distancia promedio entre sus puntos.
- **Complete Link (CL)** En cada paso se unen los dos grupos tal que su unión tiene el diámetro mínimo o los dos grupos con la menor distancia máxima entre sus elementos.

Agrupamiento: descubrimiento estructuras en los datos

- **Chamaleon** Consta de dos fases:
 - Construye el grafo de los k vecinos más cercanos y usa un algoritmo de particionamiento de grafo para agrupar los puntos en subgrupos.
 - Establece el par de subgrupos más similares considerando interconectividad y cercanía usando un algoritmo jerárquico aglomerativo.



- **Agrupamiento no jerárquico**, en los que el número de grupos se determina de antemano y las observaciones se van asignando a los grupos en función de su cercanía. Por ejemplo el método **k-mean**.

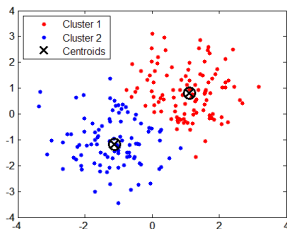
Agrupamiento: descubrimiento estructuras en los datos

k-mean

Es uno de los más simples y conocidos algoritmos de agrupamiento, sigue una forma fácil y simple para dividir una base de datos dada en k grupos (fijados a priori).

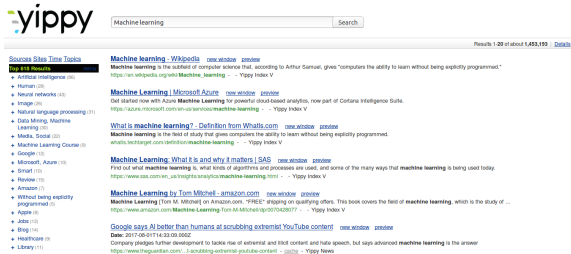
La idea principal es definir k centroides (uno para cada grupo) y luego tomar cada punto de la base de datos y situarlo en la clase de su centroide más cercano. El próximo paso es recalcular el centroide de cada grupo y volver a distribuir todos los objetos según el centroide más cercano.

El proceso se repite hasta que ya no hay cambio en los grupos de un paso al siguiente.



Agrupamiento: descubrimiento estructuras en los datos

- **Algoritmos basados en densidad**, Estos algoritmos usan diversas técnicas para determinar dichos grupos las que pueden ser por grafos, basadas en histogramas, kernels, aplicando la **regla k-NN**,compleando los conceptos de punto central, borde o ruido.
Entre ellos podemos mencionar los algoritmos **DBSCAN**, **OPTICS**, **KNNCLUST** y **SNN**

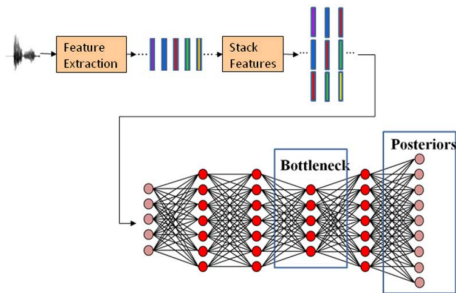


Clusty es un metabuscador y la separación por categorías de los resultados es buen ejemplo de agrupación por búsqueda web.

Predicción estructurada de datos a clases discretas

Redes neuronales y Machine Learning

Los sistemas de inteligencia artificial basados en redes neuronales se usan en sistemas de reconocimiento de imágenes o, incluso, en los sistemas de dictado de dispositivos móviles. En el fondo, funcionan de la misma manera que una persona aprende en base a su propia experiencia (y se producen conexiones entre nuestras neuronas cerebrales), una máquina podría aprender y saber qué hacer en base a información previa (o histórica) suministrada como si fuesen ejemplos (generándose también conexiones en la red neuronal artificial).

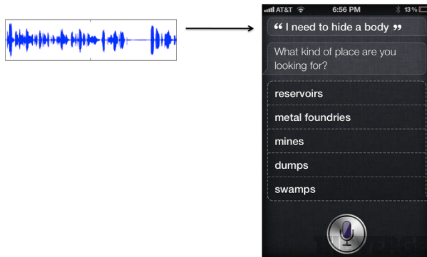


Predicción estructurada de datos a clases discretas

Reconocimiento automático de voz

Nuestra voz tiene patrones característicos como nuestra cadencia a la forma de hablar, las distintas frecuencias que generamos o los fonemas (sonidos que generamos para formar las palabras).

Si un sistema es capaz de aislar estas características y compararlas con un banco previo de patrones que ya conoce, igual que hace nuestro cerebro, sería capaz de realizar la identificación de una voz y asociarla con una persona conocida. Eso lo puede realizar el Machine Learning con una serie de API's **proyecto Oxford**.



Crecimiento del Machine Learning

El machine learning es un enfoque de:

- Reconocimiento de voz, Procesamiento del lenguaje natural.
- Visión por computador
- Control de robots
- Biología computacional
- Redes de sensores
- ...

Esta tendencia se está acelerando en:

- Big Data
- Algoritmos mejorados de Machine Learning
- Computadoras más rápidas
- Buen software de código abierto
- ...

Mapa del curso

Primera parte: Aprendizaje supervisado

- SVM, métodos del Kernel.
- Teoría del aprendizaje.
- Árboles de decisión, boosting, aprendizaje profundo.

Segunda parte: Ciencia de datos

- Aprendizaje no supervisado. Algoritmo EM.
- Reducción de la dimensión.

3. Algunos conceptos básicos

Conceptos básicos

- **Aprendizaje:** a aquello que la máquina pueda aprender a partir de la experiencia, no a partir del reconocimiento de patrones programados a priori. Por tanto, una tarea central de cómo aplicar esta definición al contexto de la computación va a consistir en alimentar la experiencia de la máquina por medio de objetos con los que entrenarse (ejemplos) para, posteriormente, aplicar los patrones que haya reconocido sobre otros objetos distintos.
- **Machine Learning** en su uso más básico es la práctica de usar algoritmos para parsear datos, aprender de ellos y luego ser capaces de hacer una predicción o sugerencia sobre algo. Dependiendo de los objetos que intentan predecir, podemos tener algunos tipos de problemas.
 - **Regresión:** Intentan predecir un valor real.
 - **Clasificación (binaria o multiclase):** Intentan predecir la clasificación de objetos sobre un conjunto de clases prefijadas. Si solo se permiten 2 posibles clases, entonces se llama **clasificación binaria**; si se permiten más de 2 clases, estamos hablando de **clasificación multiclase**.
 - **Ranking:** Intentar predecir el orden ptimo de un conjunto de objetos segn un orden de relevancia predefinido.

Conceptos básicos

Dependiendo del tipo de salida que se produzca y de cómo se aborde el tratamiento de los ejemplos, los diferentes algoritmos de Machine Learning se pueden agrupar en:

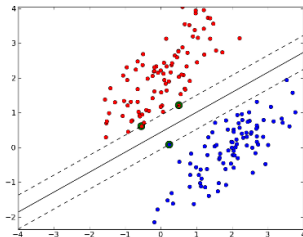
- **Aprendizaje supervisado:** se genera una función que establece una correspondencia entre las entradas y las salidas deseadas del sistema, donde la base de conocimientos del sistema está formada por ejemplos etiquetados a prior.
- **Aprendizaje no supervisado:** donde el proceso de modelado se lleva a cabo sobre un conjunto de ejemplos formados únicamente por entradas al sistema, sin conocer su clasificación correcta. Por lo que se busca que el sistema sea capaz de reconocer patrones para poder etiquetar las nuevas entradas.
Un algoritmo de aprendizaje sin supervisar es organizar los datos de alguna manera o describir su estructura. Esto puede significar agruparlos en clústeres o buscar diferentes maneras de examinar datos complejos para que parezcan más simples o más organizados.
- **Aprendizaje por refuerzo:** en este caso el algoritmo aprende observando el mundo que le rodea y con un continuo flujo de información en las dos direcciones (del mundo a la máquina, y de la máquina al mundo) realizando un proceso de ensayo-error, y reforzando aquellas acciones que reciben una respuesta positiva en el mundo.

Consideraciones al elegir un algoritmo

- **Precisión:** Se pueden usar métodos más aproximados y reducir el tiempo de procesamiento. Estos métodos tienen a evitar el **sobreajuste**.

- **Linealidad**

Muchos algoritmos de ML hacen uso de la linealidad. Los algoritmos de clasificación lineal suponen que las clases pueden estar separadas mediante una línea recta (o su análogo de mayores dimensiones).



- **Tiempo de entrenamiento:** depende de la precisión. Si el tiempo es limitado, esto puede determinar la elección del algoritmo, especialmente cuando el conjunto de datos es grande.

Consideraciones al elegir un algoritmo

- **Cantidad de parámetros:**

Los parámetros son números que afectan al comportamiento del algoritmo, como la tolerancia a errores o la cantidad de iteraciones.

Normalmente, los algoritmos con parámetros de números grandes requieren la mayor cantidad de pruebas y errores posibles para encontrar una buena combinación.

- **Cantidad de características**

Para ciertos tipos de datos, la cantidad de características puede ser muy grande en comparación con la cantidad de puntos de datos. Este suele ser el caso de la genética o los datos textuales.

Una gran cantidad de características puede trabar algunos algoritmos de ML y provocar que el tiempo de entrenamiento sea demasiado largo. Las SVM son especialmente adecuadas para este caso.

Algunos algoritmos de ML hacen determinadas suposiciones sobre la estructura de los datos o los resultados deseados.

Notaciones

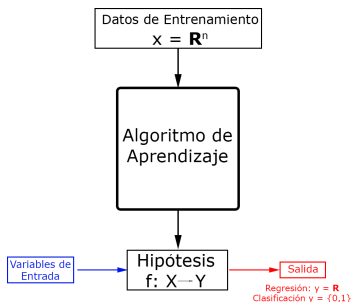
Usamos x_i para denotar **variables de entrada**, también llamadas **características** de entrada y y_i denota la salida o la variable **objetivo**.

El par (x_i, y_i) es llamado **ejemplo de entrenamiento** y el conjunto de datos que vamos a usar para aprender, una lista de m ejemplos de entrenamiento $\{(x_i, y_i); i = 1, \dots, m\}$ es llamado **conjunto de entrenamiento**. También denotaremos X como el espacio de valores de entrada e Y el espacio de valores de salida.

Cuando la variable objetivo que estamos tratando de predecir es continua, llamamos al problema de aprendizaje un problema de **regresión**. Cuando y puede tomar sólo un pequeño número de valores discretos, lo llamamos un problema de **clasificación**.

Notaciones

Para describir el problema de aprendizaje supervisado, nuestro objetivo es, dado un conjunto de entrenamiento, aprender una función $f : X \rightarrow Y$ para que $f(x)$ sea un buen predictor para el valor correspondiente de y . Esta función f se llama **hipótesis**. El proceso se ve así por ejemplo:



Aprendizaje supervisado: Encuentra f

- **Dado:** Un conjunto de entrenamiento $\{x_i, y_i\} | i = 1, \dots, N$
- **Encuentra:** Una buena aproximación a $f : X \rightarrow Y$.

Ejemplos:

- **Detección de spam:**
 - Mapea email a $\{ \text{Spam}, \text{no es Spam} \}$.
- **Reconocimiento de dígitos**
 - Mapea pixeles a $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.
- **Predicción de valores**
 - Nuevas funciones, histórico de precios, etc a \mathbb{R} (los números reales).

Un problema de Aprendizaje supervisado

Conjunto de datos:

Example	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

- Nuestro propósito es encontrar una función $f : X \rightarrow Y$
 - $X = \{0, 1\}^4$
 - $Y = \{0, 1\}$
- **Pregunta 1** ¿Cómo elegir el espacio de hipótesis, el conjunto de funciones posibles f ?
- **Pregunta 2** ¿Cómo encontramos el mejor f en el espacio de hipótesis?

Espacio de hipótesis general

Considere todas las posibles funciones booleanas sobre cuatro características de entrada.

- 2^{16} posibles hipótesis
- 2^9 son consistente con nuestro conjunto de datos
- ¿Cómo elegimos el mejor?

x_1	x_2	x_3	x_4	y
0	0	0	0	?
0	0	0	1	?
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	?
1	0	0	0	?
1	0	0	1	1
1	0	1	0	?
1	0	1	1	?
1	1	0	0	0
1	1	0	1	?
1	1	1	0	?
1	1	1	1	?

Conjunto de datos:

Example	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

Espacio de hipótesis restringido

Considera todas las funciones booleanas conjuntivas.

- 16 posibles hipótesis
- Ninguno es consistente con nuestros datos
- ¿Cómo elegimos el mejor?

Rule	Counterexample
$\Rightarrow y$	1
$x_1 \Rightarrow y$	3
$x_2 \Rightarrow y$	2
$x_3 \Rightarrow y$	1
$x_4 \Rightarrow y$	7
$x_1 \wedge x_2 \Rightarrow y$	3
$x_1 \wedge x_3 \Rightarrow y$	3
$x_1 \wedge x_4 \Rightarrow y$	3
$x_2 \wedge x_3 \Rightarrow y$	3
$x_2 \wedge x_4 \Rightarrow y$	3
$x_3 \wedge x_4 \Rightarrow y$	4
$x_1 \wedge x_2 \wedge x_3 \Rightarrow y$	3
$x_1 \wedge x_2 \wedge x_4 \Rightarrow y$	3
$x_1 \wedge x_3 \wedge x_4 \Rightarrow y$	3
$x_2 \wedge x_3 \wedge x_4 \Rightarrow y$	3
$x_1 \wedge x_2 \wedge x_3 \wedge x_4 \Rightarrow y$	3

Conjunto de datos:

Example	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

Principio de la navaja de Ockham

- Guillermo de Ockham, fue un monje que vivió en los años 1280 – 1349.
- Principio de parsimonia:
No se debe aumentar, más allá de lo necesario, el número de entidades necesarias para explicar cualquier cosa
- Cuando hay **muchas** soluciones disponibles para un problema dado, debemos seleccionar la más **simple**. Pero, ¿qué entendemos por **simple**?
- Usaremos el **conocimiento previo (prior)** del problema para resolver para definir lo que es una solución simple.
Ejemplo de un prior: smoothness

Aspectos claves del Machine Learning

- ¿Cómo elegimos un espacio de hipótesis?
 - Con frecuencia utilizamos el conocimiento previo para guiar esta elección.
- ¿Cómo podemos medir la exactitud de una hipótesis sobre datos nuevos?
 - **Principio de la navaja de Ockham:** utiliza la hipótesis más simple compatible con los datos!.Esto nos ayudará a evitar el sobreajuste.
 - **La teoria del aprendizaje** nos ayudará a cuantificar nuestra capacidad de generalizar como una función de la cantidad de datos de entrenamiento y el espacio de hipótesis.
- ¿Cómo encontramos la mejor hipótesis?
 - Esta es una pregunta algorítmica, el tema principal de la ciencia de la computación.
- ¿Cómo modelar aplicaciones como problemas de Machine Learning? (desafío de la ingeniería)