Métodos actuales en machine learning

23º Escuela de Verano
de Ciencias Informáticas

Gracias a los organizadores!

Métodos actuales en machine learning

Pablo M. Granitto
Dr. en Física
Docente FCEIA – UN Rosario
Investigadador en CIFASIS (CONICET)

Lucas C. Uzal
Dr. en Física
Docente FCEIA – UN Rosario
Investigadador en CIFASIS (CONICET)

Outline general

- Intro y cosas generales (Lu-Pablo)
- Ensambles (Ma-Pablo)
- Métodos de Kernel (Mi-Pablo)
- Redes neuronales profundas (Ju-Vi-Lucas)

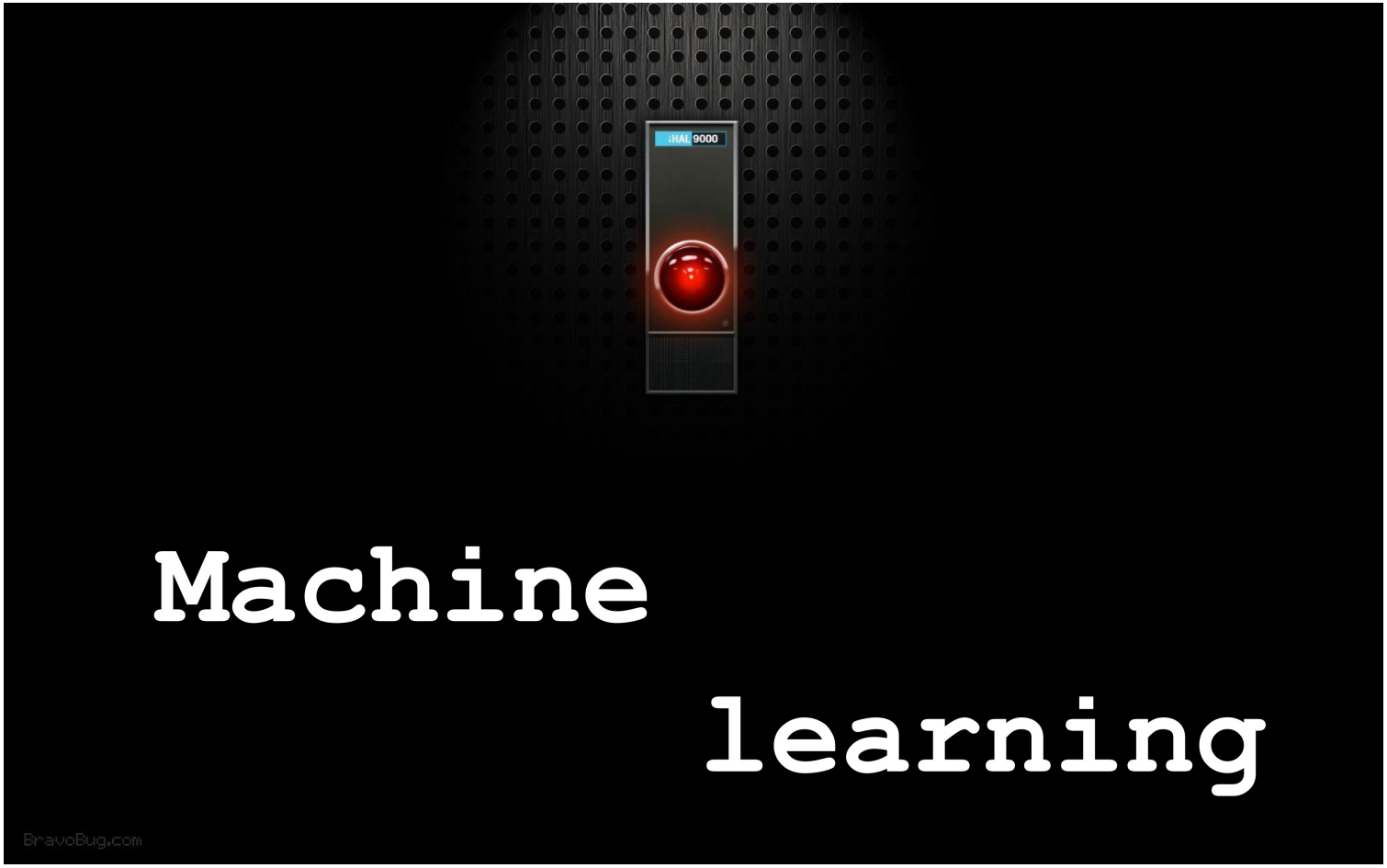
Hoy

- Introducción a ML
- Problemas
- Métodos básicos
- Evaluación de resultados

Extra:

- Selección de inputs

Introducción



Qué es Machine Learning?

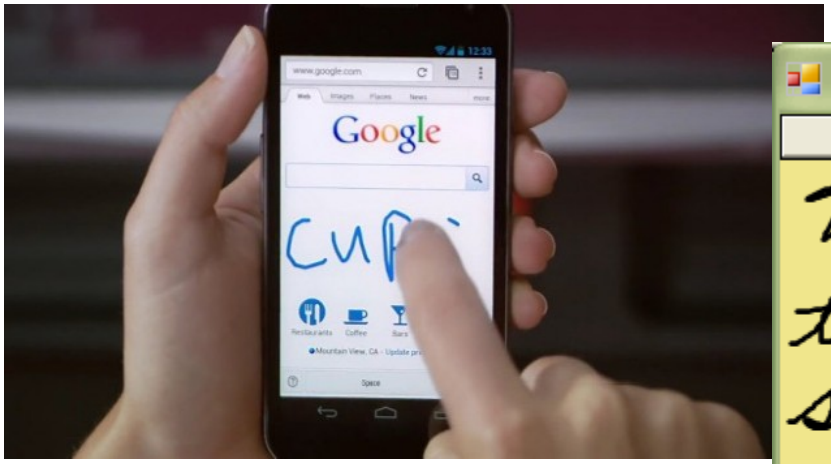
Introducción

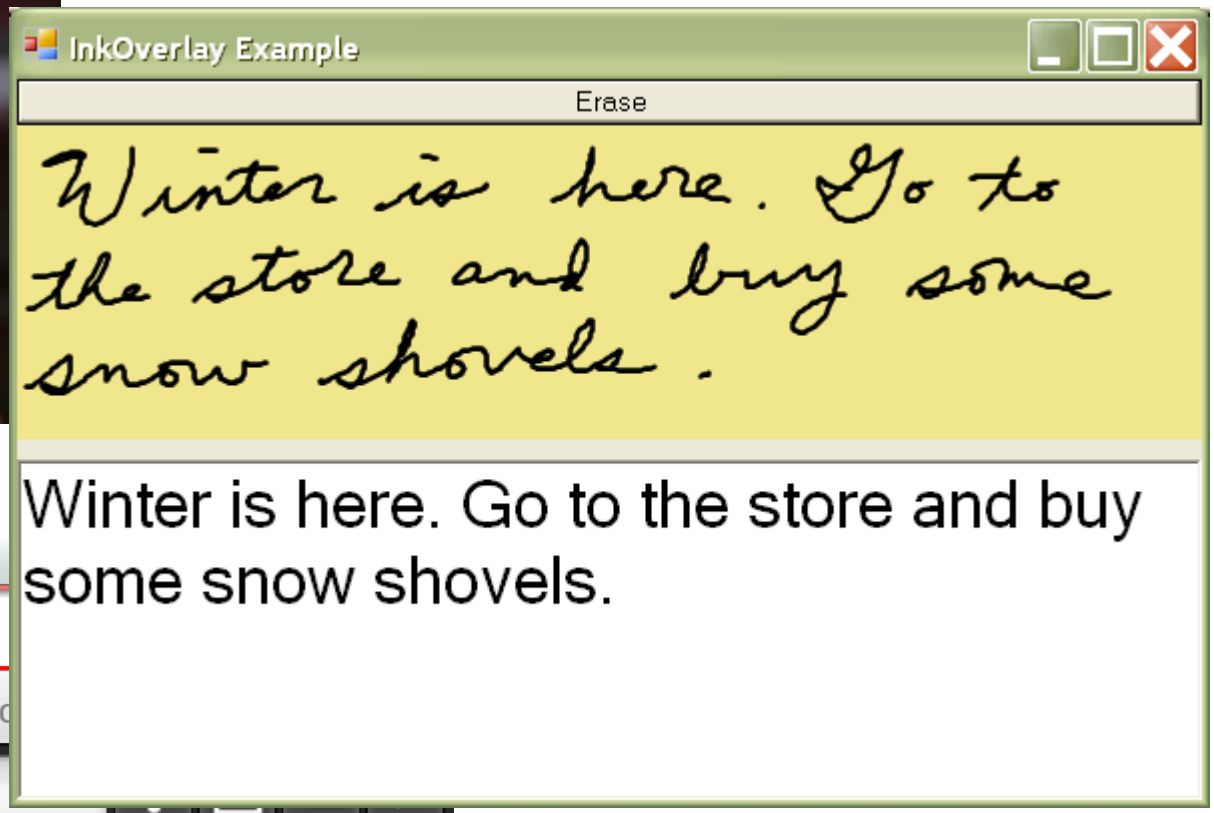
- Hay problemas en Informática que se pueden “definir” concretamente y son simples de convertir en un algoritmo
  - Ejemplo: Ordenar alfabéticamente una lista, calcular el balance de una cuenta.

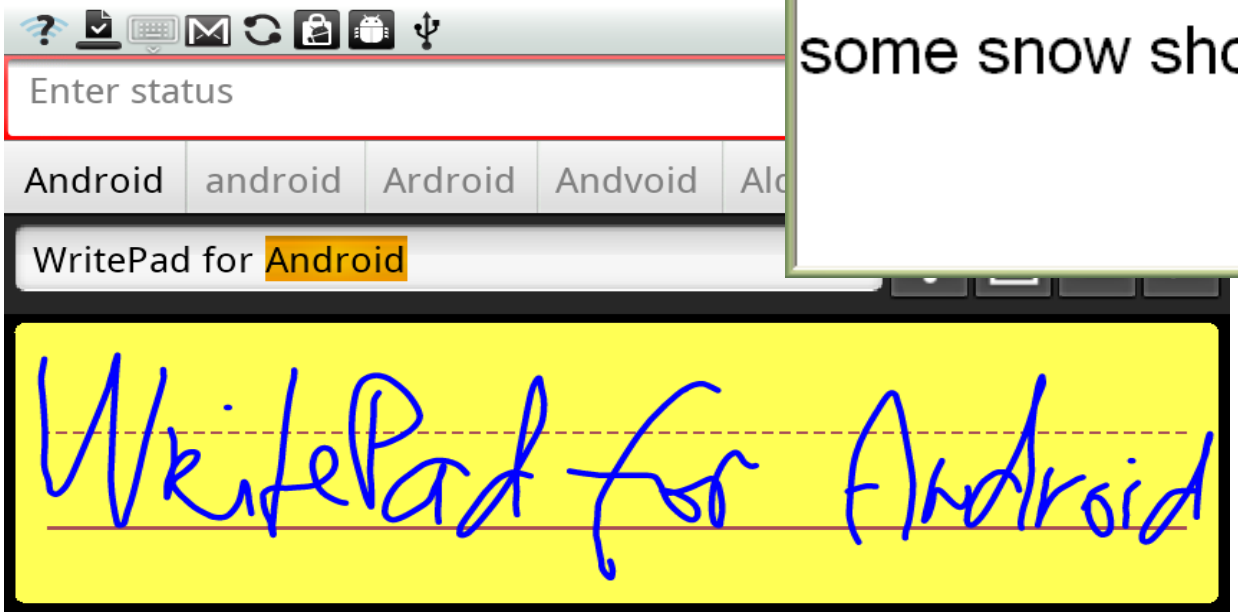
- Hay otros que son simples de “entender” pero muy difíciles de “definir” y convertir en algoritmo
  - Ejemplo: Detectar una sonrisa en una cara, interpretar un gesto del lápiz como una letra dada

El Aprendizaje Automatizado introduce métodos que pueden resolver esas tareas “aprendiendo” la solución a partir de ejemplos de como se realiza la misma

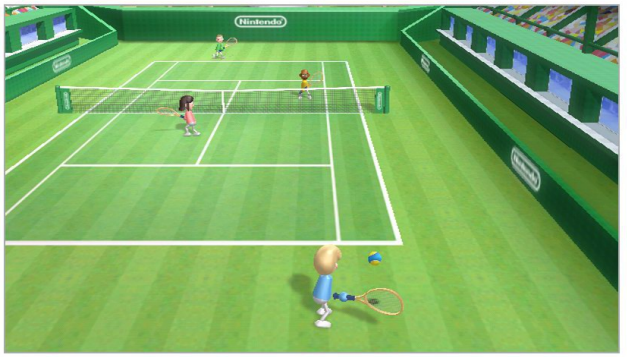
Introducción



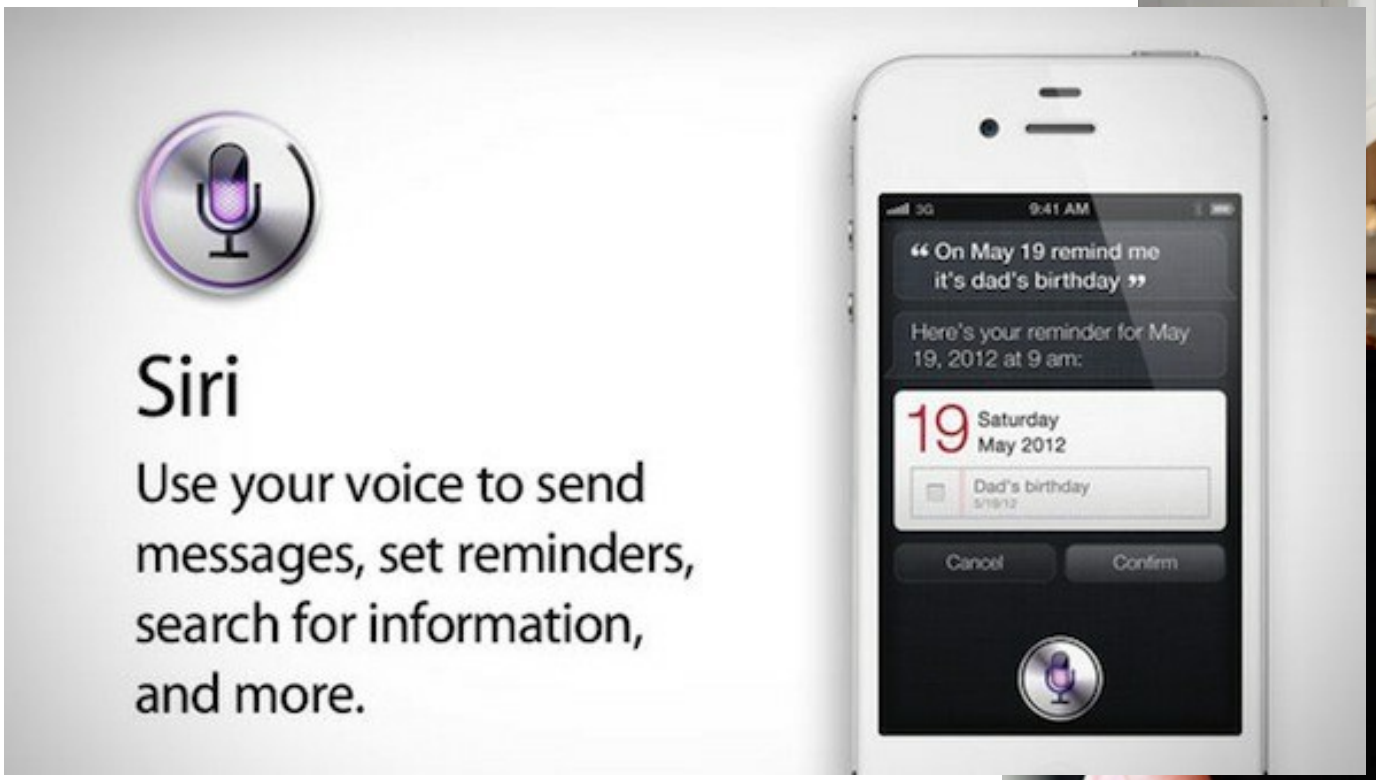


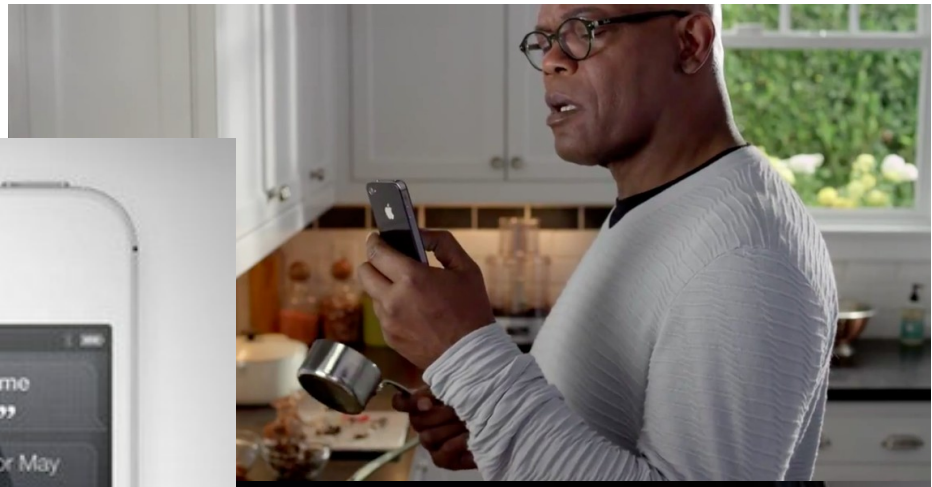


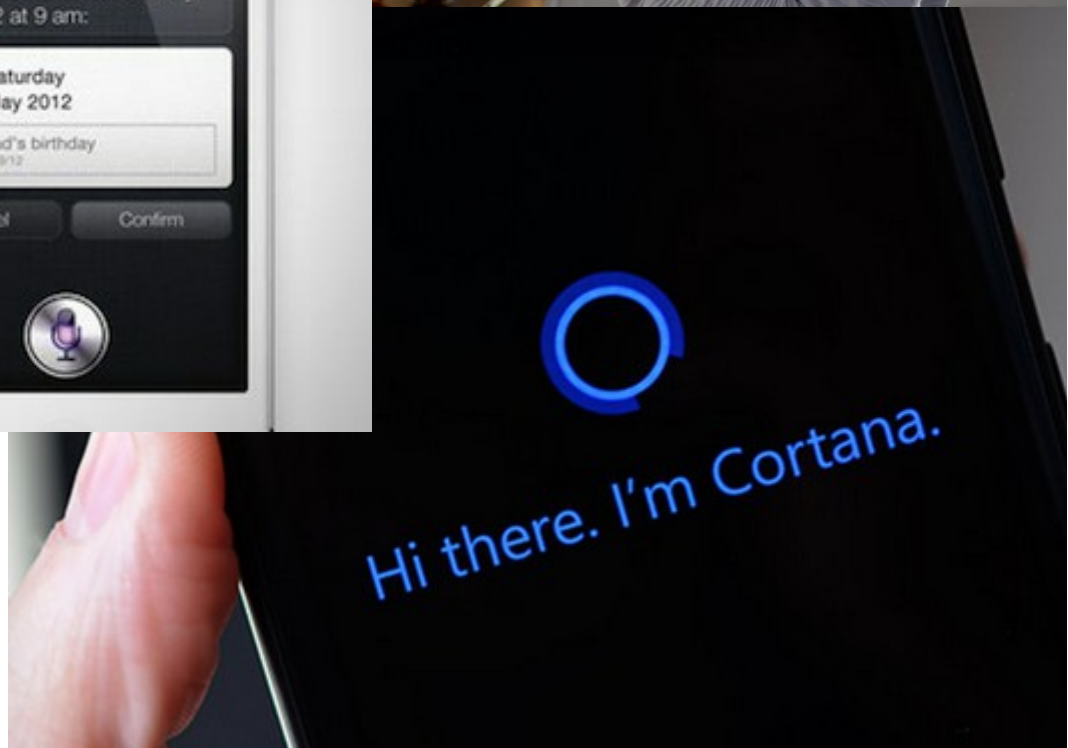
Introducción



Introducción







Introducción

Problemas en ML

- Clasificación
- Regresión
- Ranking-Retrieval
- Detección de novedades
- Clustering
- Identificación de inputs relevantes
- Etc, etc.

Clasificación

Problema:

Dado un objeto (conjunto de características medidas de alguna forma) asignarle una (o varias) etiqueta de un conjunto finito.

Ejemplo:

asignar un símbolo alfanumérico a una secuencia de movimientos del lápiz en la pantalla táctil

Asignar automáticamente una noticia a diferentes grupos de interés (una o más clases)

Regresión

Problema:

Dado un objeto asignarle un número real.

Ejemplo:

Predecir la relación euro-dolar de mañana.

Predecir niveles de stock/ventas a futuro.

Búsqueda y Ranking

Problema:

Dado un objeto, asignarle y ordenar las respuestas más probables dentro de una base de datos.

Ejemplo:

Buscadores en Internet

Sistemas de recomendación

Detección de novedades

Problema:

Detectar "outliers", objetos que son diferentes a los demás.

Ejemplo:

Alarmas de comportamiento en compras con tarjeta.

Detección de fallas en equipos críticos.

Clustering

Problema:

Detectar grupos de objetos que tienen características similares.

Ejemplo:

Segmentación de consumidores/clientes a partir de sus patrones de compra/búsqueda. Marketing "dirigido".

Detección de inputs relevantes

Problema:

Dado uno de los problemas anteriores (u otro) y sus datos, averiguar cuales de las variables son responsables de la solución.

Ejemplo:

El "nuevo método científico": tomar muestras sanas y con alguna enfermedad. Analizar miles de variables con un método automático (MALDI-TOF, DNA-microchips) y buscar cuales de las variables monitoreadas son relevantes al problema.

Programas que aprenden?

“Se dice que un programa aprende si mejora su performance en una cierta tarea al incorporar experiencia”

Programas que aprenden?

Memorizar no es aprender

Generalizar es aprender

Como logramos generalizar?

Tengo estos datos:

8 – T

2 – T

5 – F

9 – F

4 – T

13 – F

Cual es la respuesta para 12?

Y si agrego los datos:

14 – F

16 - T

Como logramos generalizar?

Para generalizar incorporamos “algo” a los datos: un bias.

En general usamos la “navaja de Occam”: La respuesta más simple que explica las observaciones es la válida

Distintos métodos de ML usan distintos bias

Métodos básicos

Arboles de decisión

- Probablemente el método más conocido para resolver problemas de clasificación
- Muy asociado a nuestra forma de proceder
- Uno de los primeros desarrollos en ML

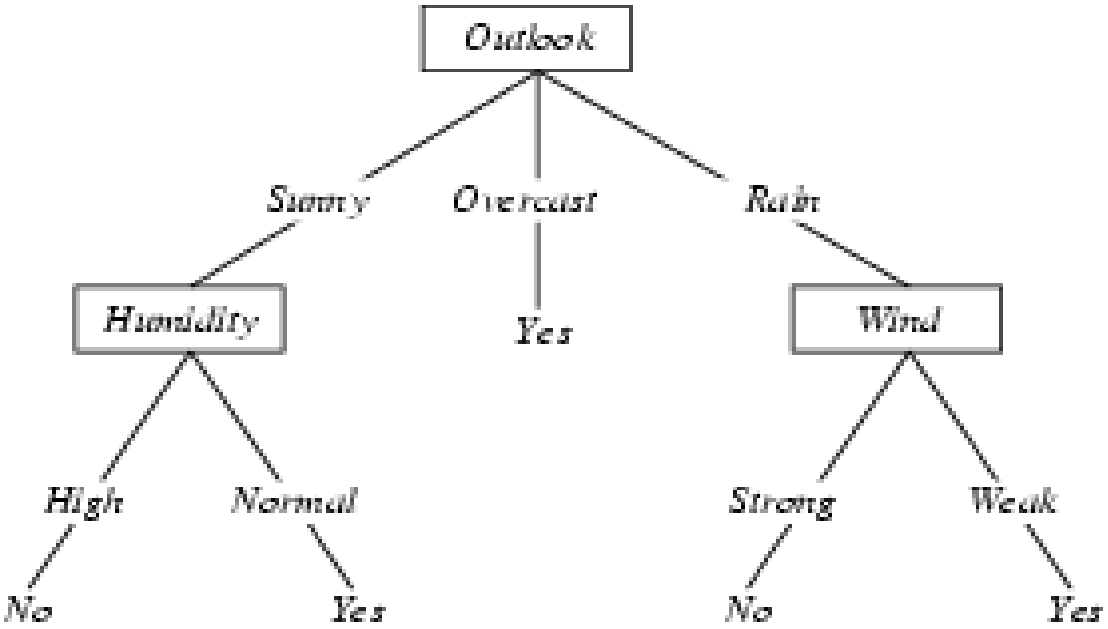
Arboles de decisión

Ejemplo: “Play Tennis”

Day	Outlook	Temperature	Humidity	Wind	PlayTenn
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Arboles de decisión

Ejemplo: “Play Tennis”



Arboles de decisión

Cómo construímos el árbol?

- Hay variables más relevantes que otras.
- Si ponemos más alto las más relevantes, seguramente el árbol llegara antes a la solución, será mas simple.
- Un árbol más simple seguramente generalizará mejor (Occam).

Arboles de decisión

Cómo elegir que variable usar para dividir?

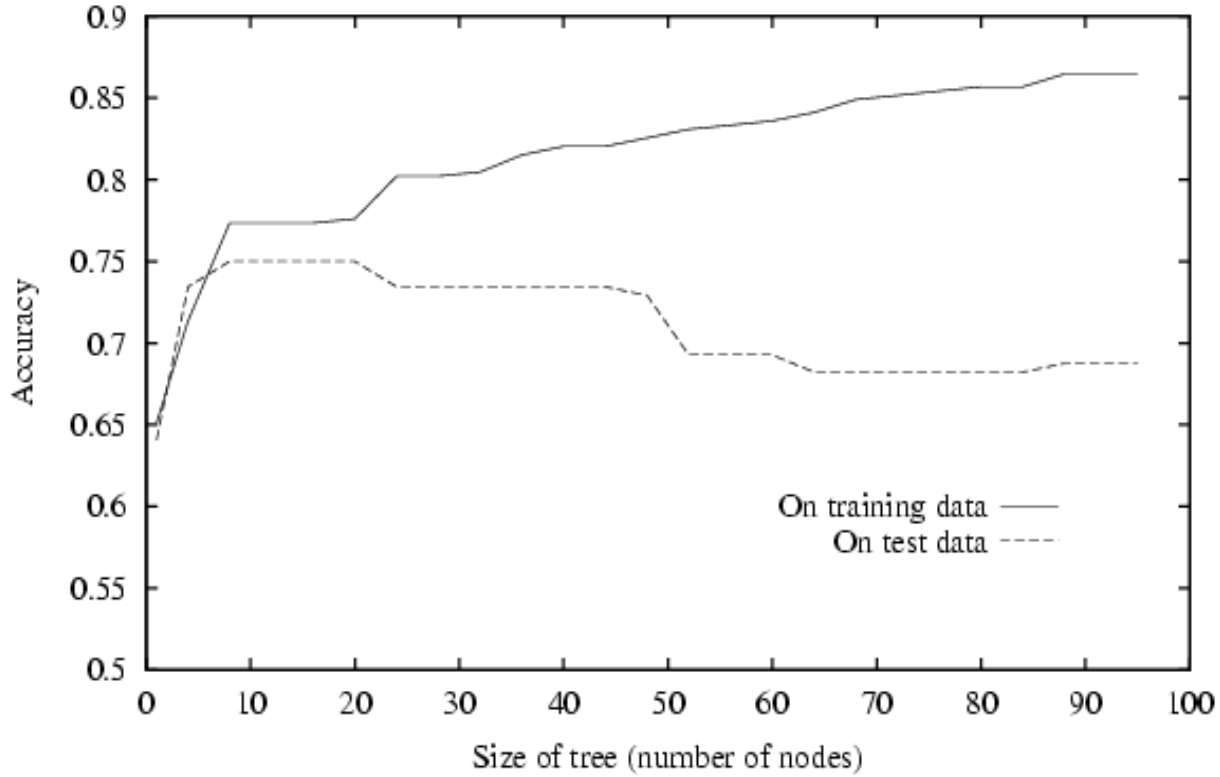
- Necesitamos la más “relevante”
- Busquemos la que dá más información sobre la clase->Information Gain, correlación, etc.
- Dividimos e iteramos

Arboles de decisión

Hasta cuando dividimos?

- Hasta que tenga clases puras en todos los nodos
- Hasta que no tenga más variables disponibles
- O hay algo mejor?

Arboles de decisión



Sobreajuste!

Arboles de decisión

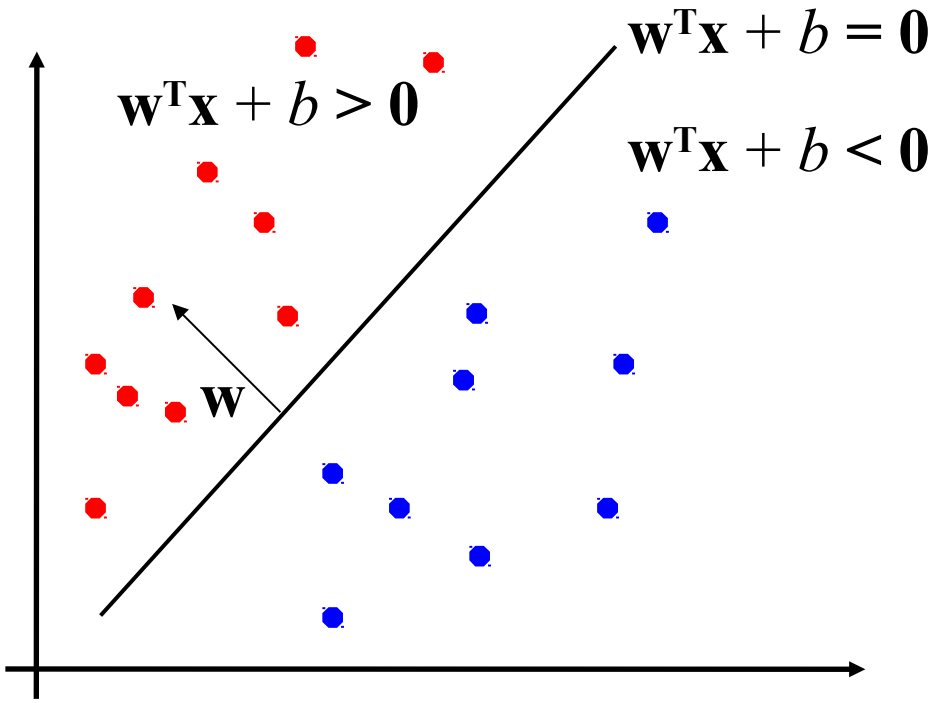
Cómo controlar el sobreajuste?

- Necesitamos un conjunto independiente de datos, sobre el que podamos controlar la capacidad de generalización (“Validación”).
- Cuando deja de mejorar, detenemos el proceso.

Este proceso se suele llamar “Model Selection”

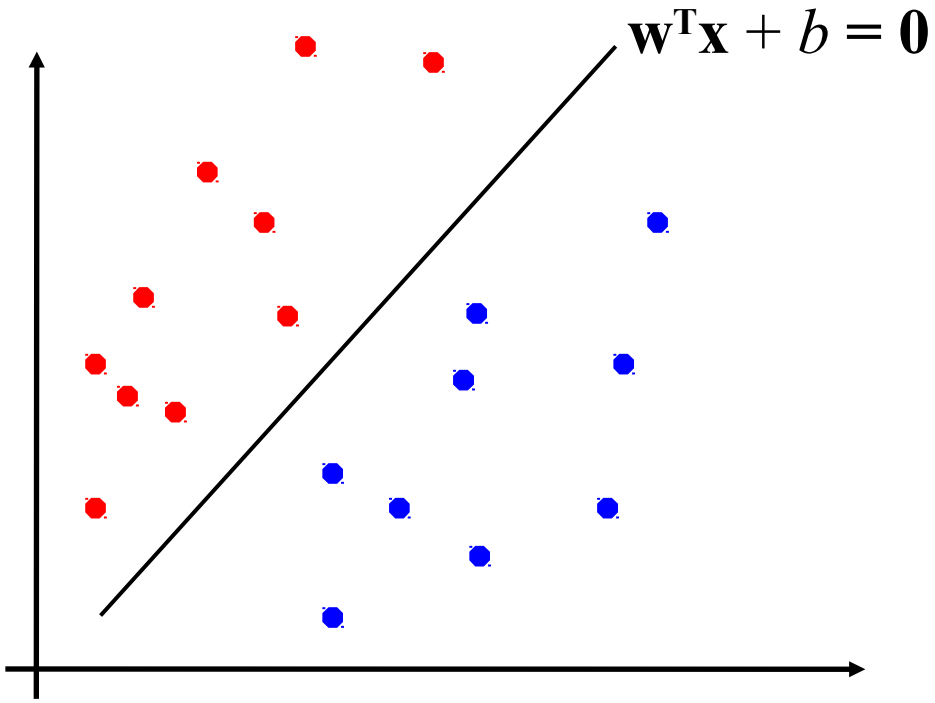
El perceptron

Si los datos están en un espacio vectorial...



$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x} + b)$$

El perceptron

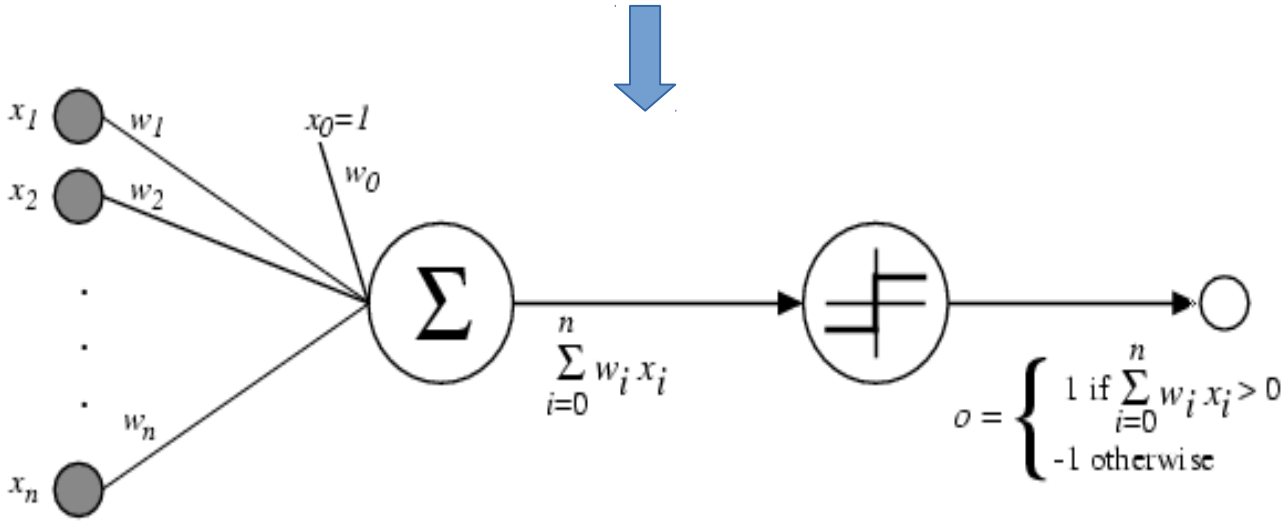


$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x} + b)$$

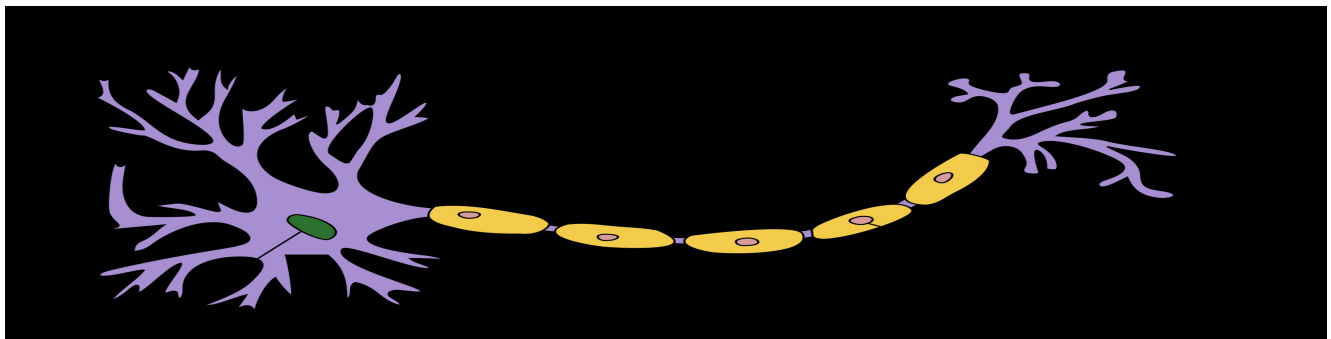
- Tenemos una regla de clasificación de forma fija
- Aprender: encontrar el mejor W y b para el problema
- Regla de aprendizaje del perceptron: si es incorrecto muevo W hacia el ejemplo
- Converge a la solución

El perceptron

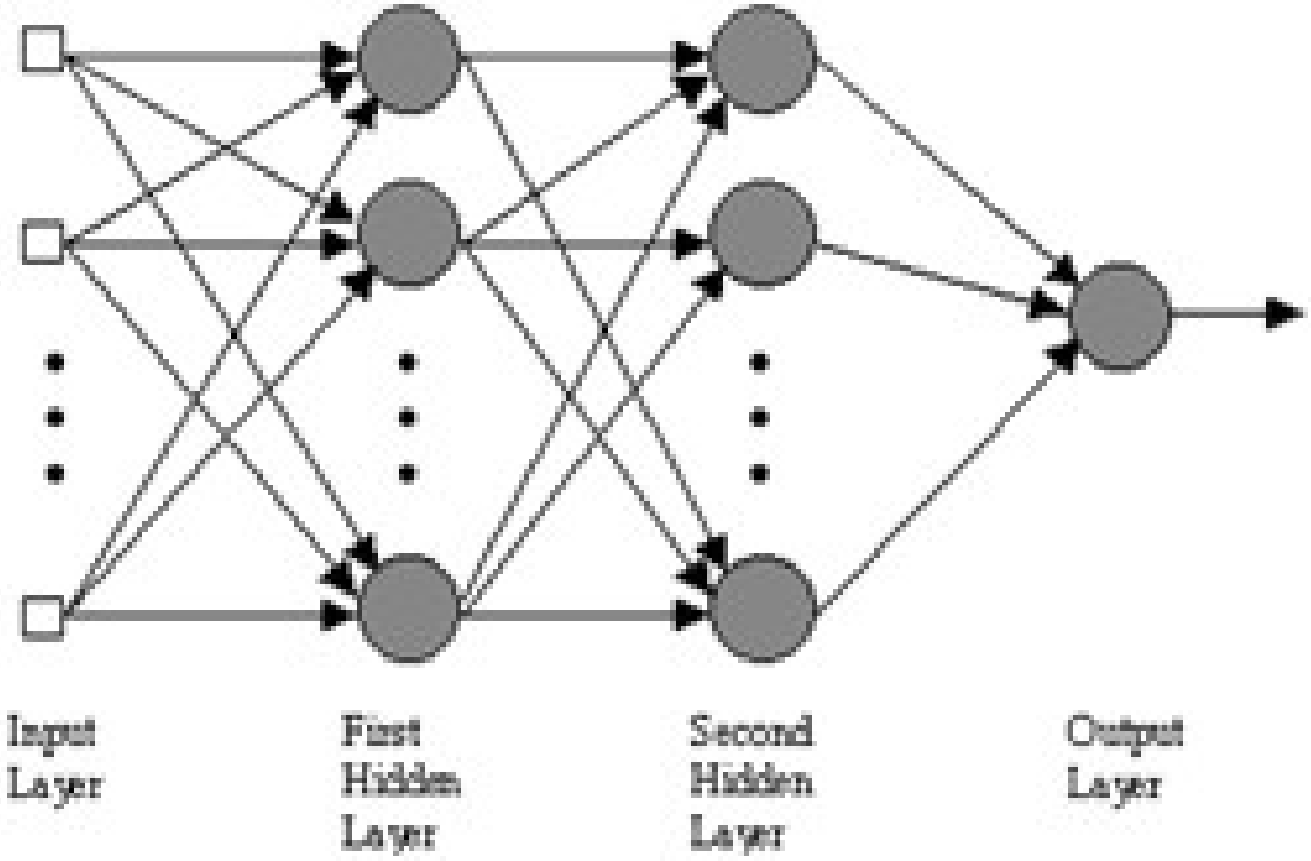
$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^{\mathrm{T}}\mathbf{x} + b)$$



$$o(x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } w_0 + w_1 x_1 + \cdots + w_n x_n > 0 \\ -1 & \text{otherwise.} \end{cases}$$



El perceptron multicapa o red neuronal



El perceptron multicapa

Como elegir los parámetros para algo tan complejo como la función implementada por un MLP?

Hay que buscar por algún método valores que hagan mínimo el error sobre los datos, que tengan el valor correcto en la salida

Descenso por el gradiente!

El perceptron multicapa

Descenso por el gradiente:

Definir la función error a minimizar

- Calcular el gradiente
- Mover los parámetros en la dirección que minimiza el error
- Iterar hasta converger

Heurísticas para mejorar la convergencia

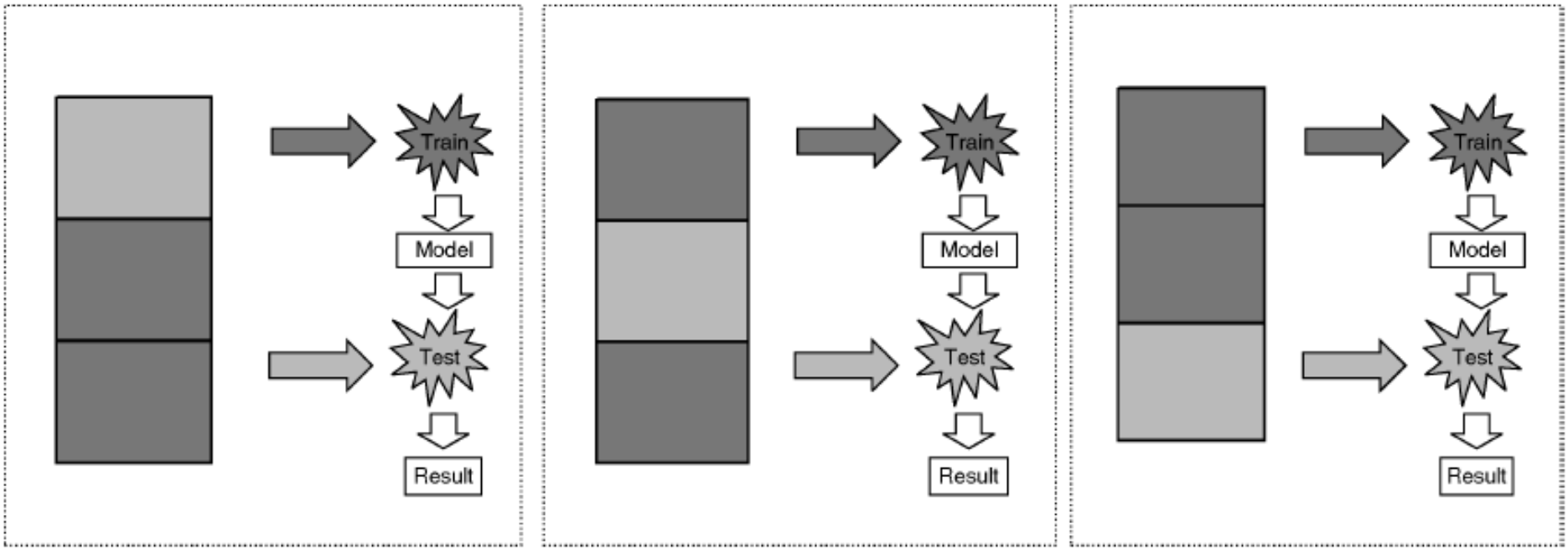
Estimación del error

Como estimar qué tan bueno es lo que aprendí?

La solución más simple es medirlo en una muestra de datos similar a la que voy a usar a futuro: Conjunto de test

Estimación del error

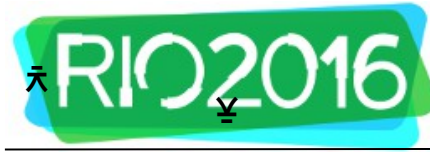
Si no tengo un conjunto de test, estimo con Cross-Validation



Cross-Validation. Figure 1. Procedure of three-fold cross-validation.

Límite: Leave-one-out

Como resolver un problema en ML



- Identificar el problema y conseguir conocimiento experto
- Conseguir datos, muchos datos!
- Elegir un método adecuado (o varios)
- Entrenar varios modelos con el conjunto de train, evaluarlos con el conjunto de validación, elegir el mejor
- Estimar el error con el conjunto de test

Selección de inputs

23º Escuela de Verano
de Ciencias Informáticas

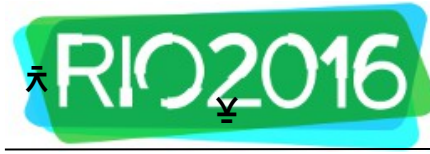
Selección de variables: Por qué?

- Muchos problemas actuales tienen cientos o miles de variables medidas (sobre pocos ejemplos)
- Modelar esos problemas “directamente” suele ser sub-óptimo.
  - Tanto en calidad como en interpretabilidad.
- En algunos casos la “extracción de variables” (pca, ica, etc.) no es una opción válida.

Selección de variables: Para qué?

- Para mejorar la performance de los métodos de aprendizaje:
  - Algunos métodos trabajan mucho mejor con menos variables.
    - Aunque los métodos modernos de ML suelen ser muy resistentes al problema de la dimensionalidad.
  - En ciertos casos muchas variables no son informativas del problema (ruido o redundancias).
    - Al eliminarlas reducimos el riesgo de sobreajuste.

Selección de variables: Para qué?



- Para descubrir:
  - Cuáles son las variables más importantes en un problema.
  - Cuáles variables están correlacionadas, co-reguladas, o son dependientes y cuáles no.

- La selección de variables no es más una técnica de pre-procesado, actualmente es una herramienta para descubrir información de un problema.

Métodos

- **Univariados** consideran una variable a la vez.
- **Multivariados:** consideran subconjuntos de variables al mismo tiempo.
- **Filtros:** Ordenan las variables con criterios de importancia independientes del predictor.
- **Wrappers:** Usan el predictor final para evaluar la utilidad de las variables.

Métodos

- **Problema Base:**

  Seleccionar un subconjunto óptimo de p variables de las n variables originales, dado un criterio.

- Por qué no evaluar todas las posibilidades?

  Explosión combinatoria: $$\sum_{p=1}^{n} C_n^p = \sum_{p=1}^{n} \frac{n!}{p!(1-p)!}$$

Se usan soluciones sub-óptimas sobre eurísticas.

Métodos de Filtro

- Elige las mejores variables usando criterios razonables de “importancia”.
- El criterio es generalmente independiente del problema real.
- Usualmente se usan criterios univariados.
- Se ordenan las variables en base al criterio y se retienen las más importantes (criterio de corte!)

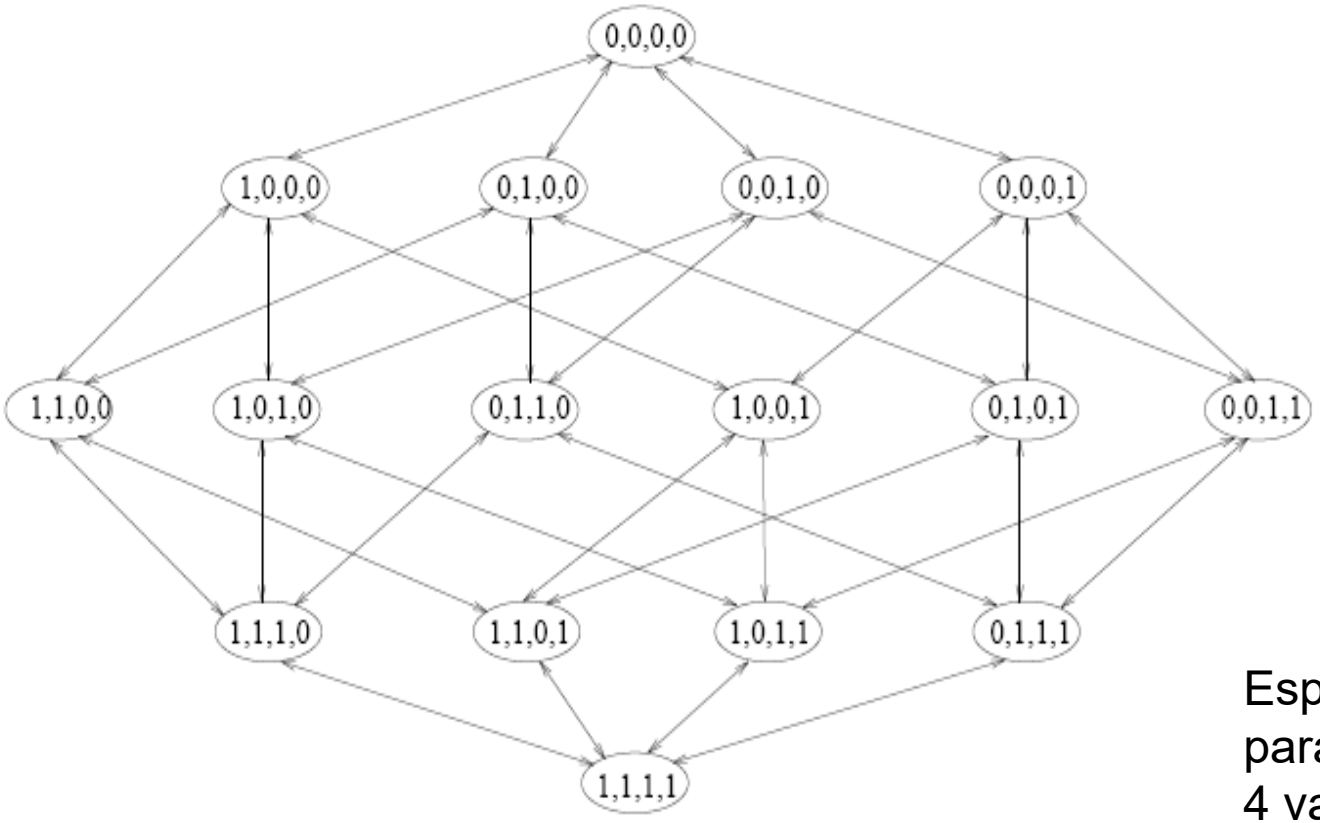
Wrappers. Claves

- Seleccionar las mejores variables para modelar (usando el criterio final)
- Para cada subconjunto de variables resolver el problema de modelado. Conservar la mejor solución.
- Como ya discutimos, la búsqueda completa es exponencialmente larga.

Wrappers. Alternativas

- Búsquedas Greedy:
  - forward selection
  - backward elimination
  - combinaciones de ambas
- Búsquedas pseudo-random:
  - Simulated annealing
  - genetic algorithm

Wrappers. Ejemplo

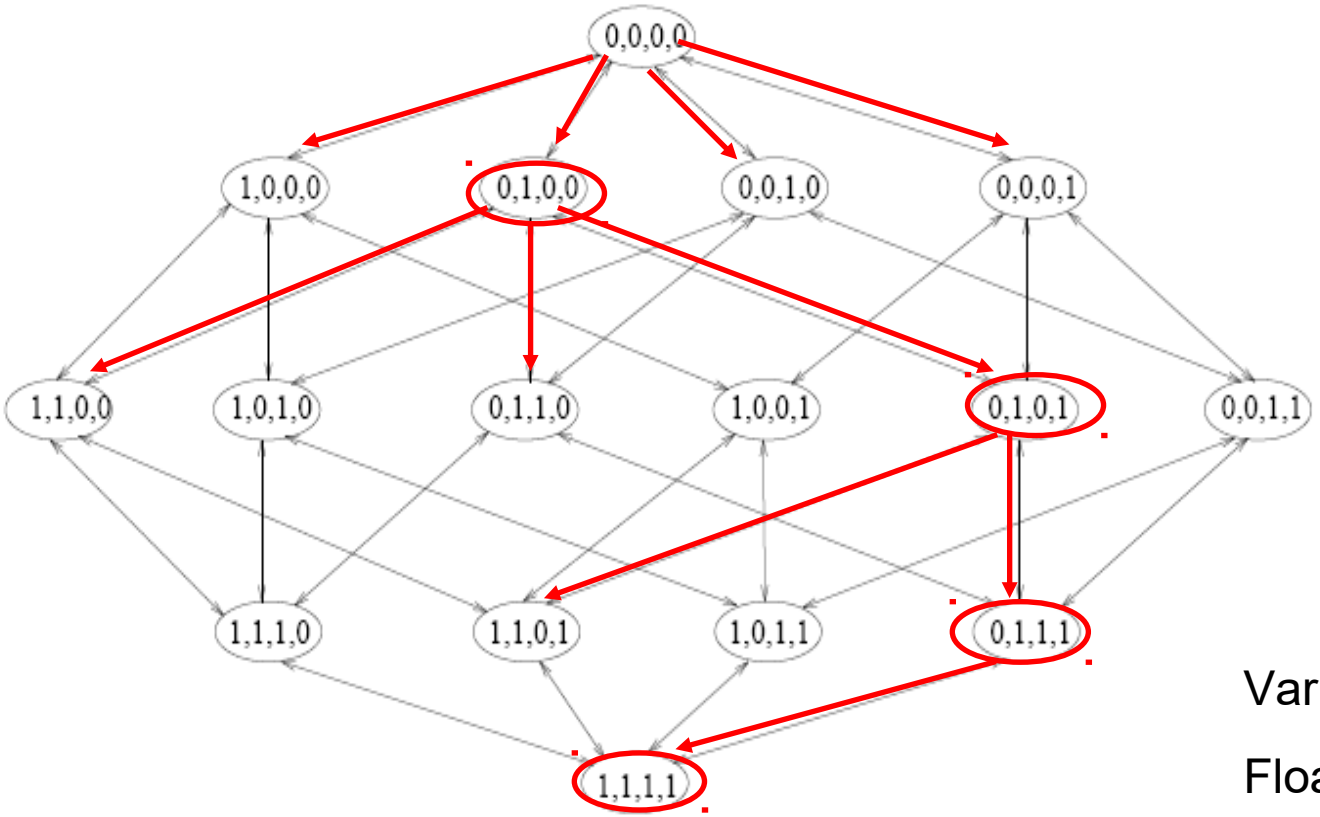


Kohavi-John, 1997

Espacio de búsqueda para un problema con 4 variables.

0 ausente - 1 presente

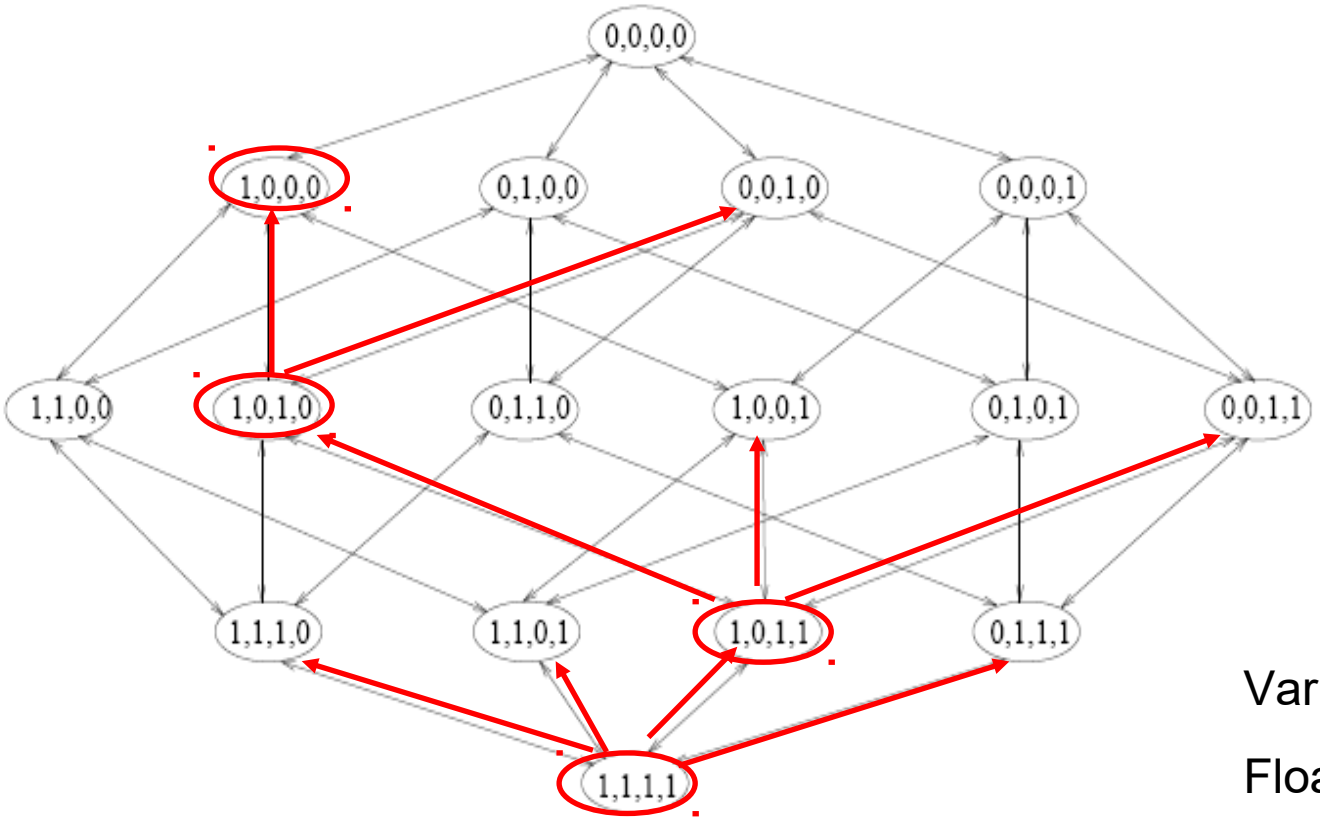
Wrappers. Forward search



Variantes:

Floating search. 1 paso adelante, 1 atrás

Wrappers. Backward search



Variantes:

Floating search

Métodos embebidos

- Los wrappers backward son potencialmente los mejores métodos de selección.
- Son computacionalmente muy pesados.
  - A cada paso construyen todos los clasificadores intermedios posibles para evaluarlos.
  - Para rankear p variables crean $O(p^2)$ modelos.
- La solución ideal sería un método backward, basado directamente en el modelo final, pero eficiente.

Métodos embebidos

- Para evaluar cuál es la próxima variable a eliminar, el wrapper back construye todos los modelos con una variable menos.
- Los evalúa a todos y “da un paso” en la dirección de máximo descenso del error.
- Se puede hacer algo parecido sin calcular todos los modelos target?

Métodos embebidos

- Lo que necesitamos conocer es la derivada del error respecto de cada variable.
  - O alguna aproximación a la derivada
  - Se las llama “medidas internas de importancia”
- Si la función error es razonablemente suave, dar el paso en la dirección de máximo descenso de la derivada debería ser lo mismo que el máximo descenso del error.

Métodos embebidos

- Recursive Feature elimination (RFE):
  - Ajustar un modelo a los datos
  - Rankear las variables usando una medida interna de importancia.
    - Más importante es la que más empeora al modelo al ser eliminada
  - Eliminar la variable (o un grupo) con el ranking más bajo
  - iterar

Medidas de importancia

- Ejemplos de medidas de importancia:
  - SVM: componentes de W
  - Random Forest:
    - Shuffling OOB
    - Variación del GINI index.
  - LDA o PDA o Regresión logística: weights
  - Partial Least Squares (PLS): Scores
  - ANN:
    - Saliency (Solla et al, NIPS)
    - Shuffling OOB (Izetta, CACIC09)

Problemas con el RFE

- Variables importantes pero correlacionadas
  - Si el modelo original usa las dos, las dos “comparten” la importancia.
  - En la práctica aparecen como menos importantes que otras variables.
  - Al eliminar una de ellas, la otra toma toda la importancia y suele subir bruscamente en el ranking (por esto es necesario iterar).

Problemas con el RFE

- Variables importantes pero correlacionadas
  - Cuál es eliminada y cuál promocionada es casi chance.
  - Como resultado, el ranking de variables tiende a ser inestable.

Método sugerido

- Usar el train para elegir las variables.
- Usar validación independiente para determinar la cantidad de variables a retener.
- Hacer una selección final con train+validación.
- Estimar el error con el test set.