

# Machine Learning

## Aprendizaje basado en árboles de decisión

Angel Vázquez-Patiño  
angel.vazquezp@ucuenca.edu.ec

Departamento de Ciencias de la Computación  
Universidad de Cuenca

18 de octubre de 2017

# Objetivos

1. Entender la utilidad de los árboles de decisión (ADD) para aprender conceptos
2. Saber para qué tipo de problemas es adecuado usar aprendizaje basado en ADD
3. Entender cómo construir un árbol de decisión con ID3
4. Utilizar un ADD para clasificar datos más allá de  $\mathcal{D}$
5. Entender qué es el bias inductivo
6. Saber a qué se refiere el principio de Occam's razor
7. Entender qué es el overfitting y cómo tratar de evitarlo
8. Ver otros tipos de problemas que se pueden presentar al usar ADD (sobre todo relacionado a  $\mathcal{D}$ )

# Contenido

Representación

Problemas apropiados

Algoritmo básico: ID3

Búsqueda en el hypothesis set

Bias inductivo

Cuestiones importantes

# Árboles de decisión (ADD)

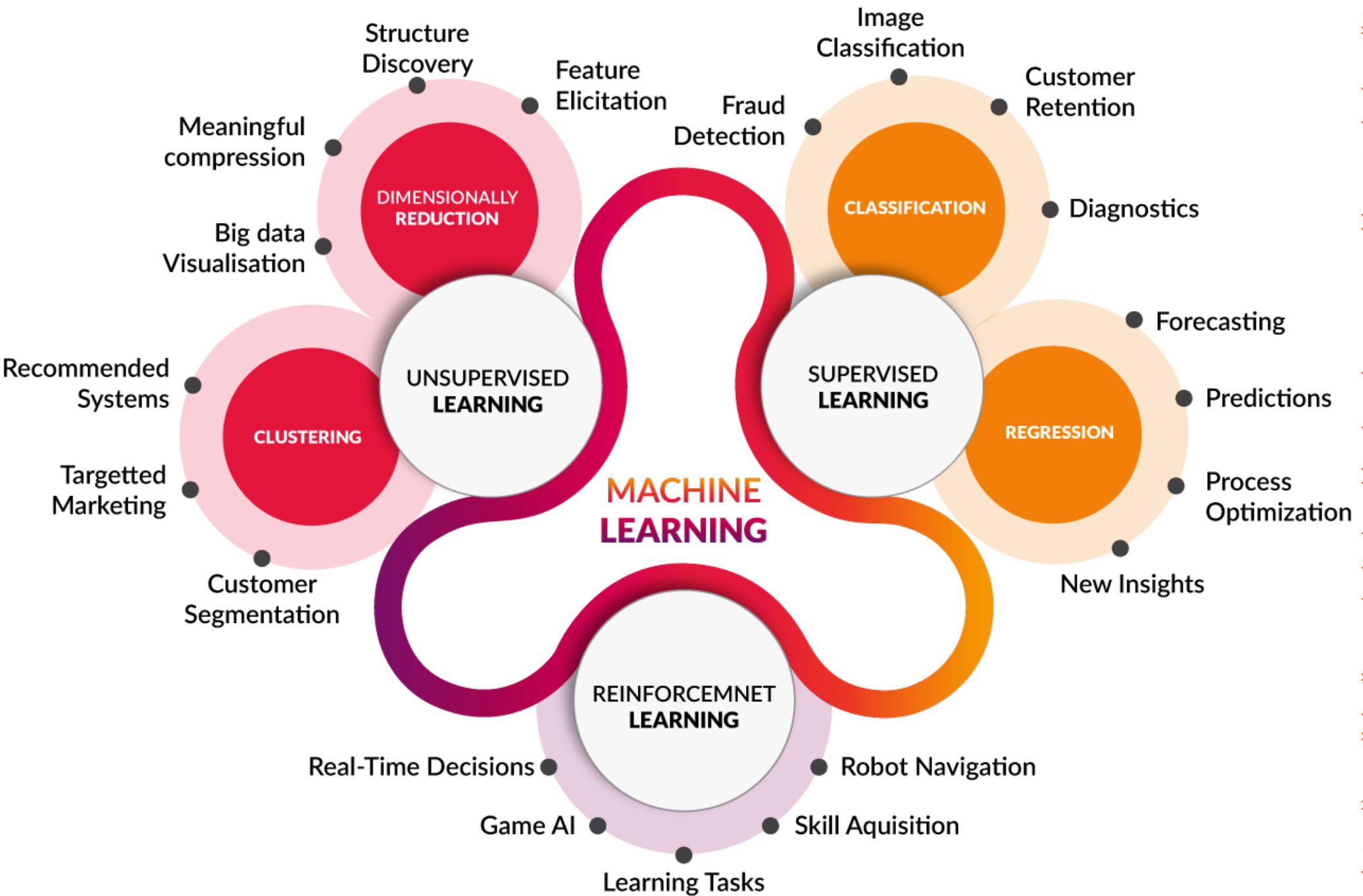
- De los más usados y prácticos en inferencia inductiva
- Aproxima funciones **discretas** de manera robusta para datos con **ruido**
- Capaz de aprender expresiones disjuntivas

## Algoritmos de aprendizaje de árboles de decisión

- ID3
- ASSISTANT
- C4.5

# Árboles de decisión

- Los algoritmos de aprendizaje dan una hipótesis de un conjunto de hipótesis muy expresiva
- La hipótesis  $g$  aprendida es representada por un ADD
- Los ADD aprendidos pueden ser tomados como un conjunto de reglas Si-Entonces facilitando la lectura de los humanos
- Aprendizaje supervisado: clasificación
- Diagnósticos médicos o riesgos crediticios



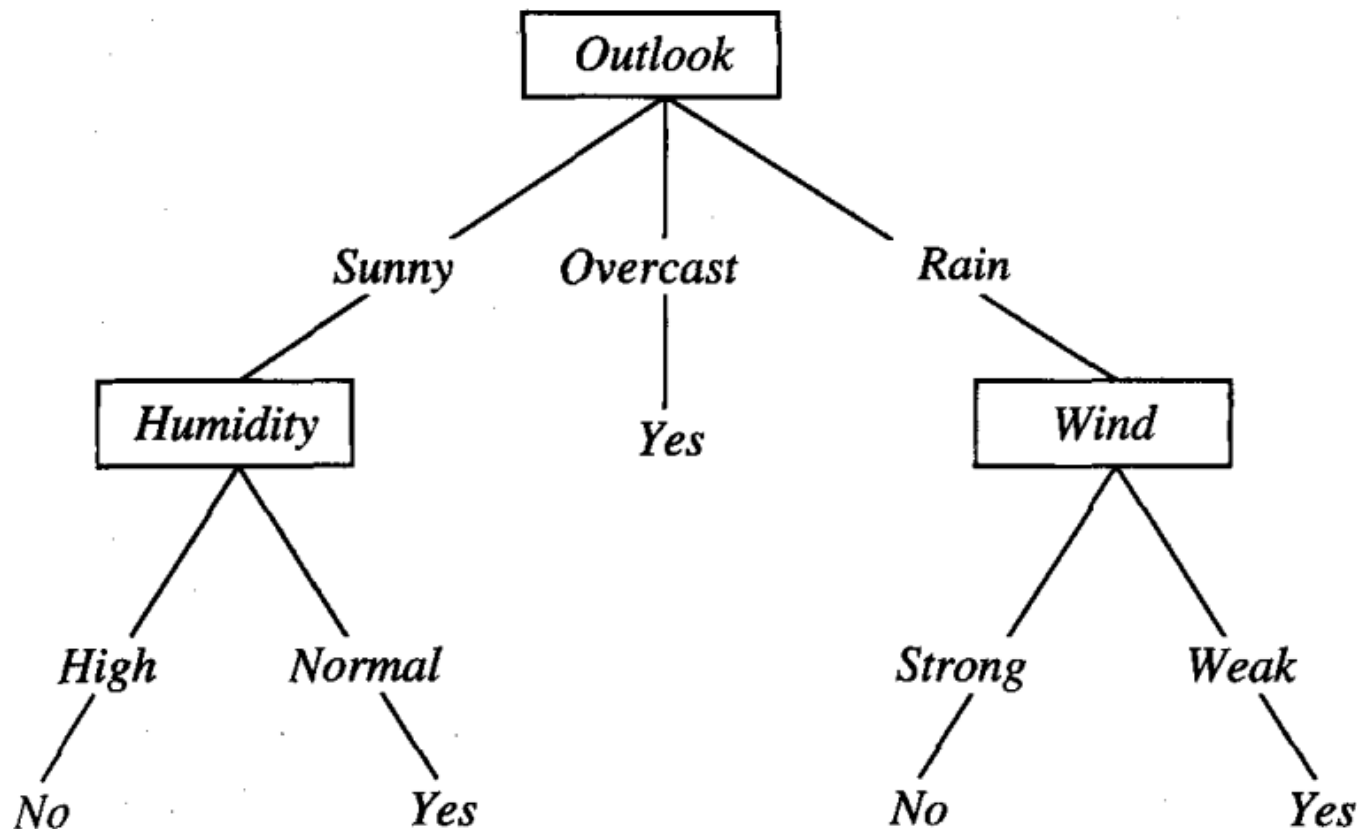
# Representación del árbol de decisión

# Representación

- Clasifica los data points (instancias) ordenándolos de manera descendente desde la raíz hasta algún nodo hoja
- Cada nodo especifica una prueba de algún atributo ( $x_d$ ) de la instancia
- Cada rama descendiendo del nodo corresponde a uno de los posibles valores del atributo



# Representación

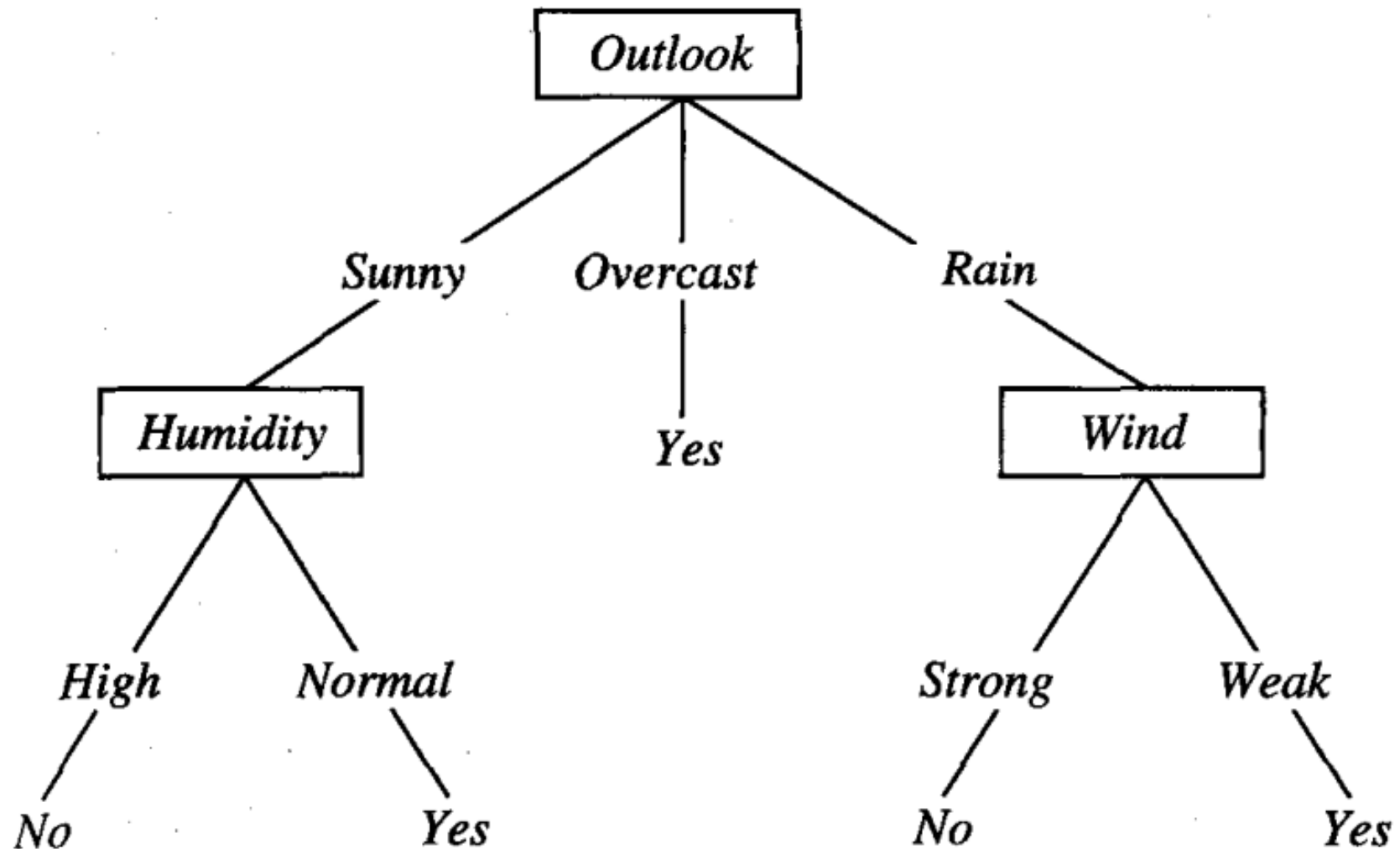


**FIGURE 3.1**

A decision tree for the concept *PlayTennis*. An example is classified by sorting it through the tree to the appropriate leaf node, then returning the classification associated with this leaf (in this case, *Yes* or *No*). This tree classifies Saturday mornings according to whether or not they are suitable for playing tennis.

# Representación

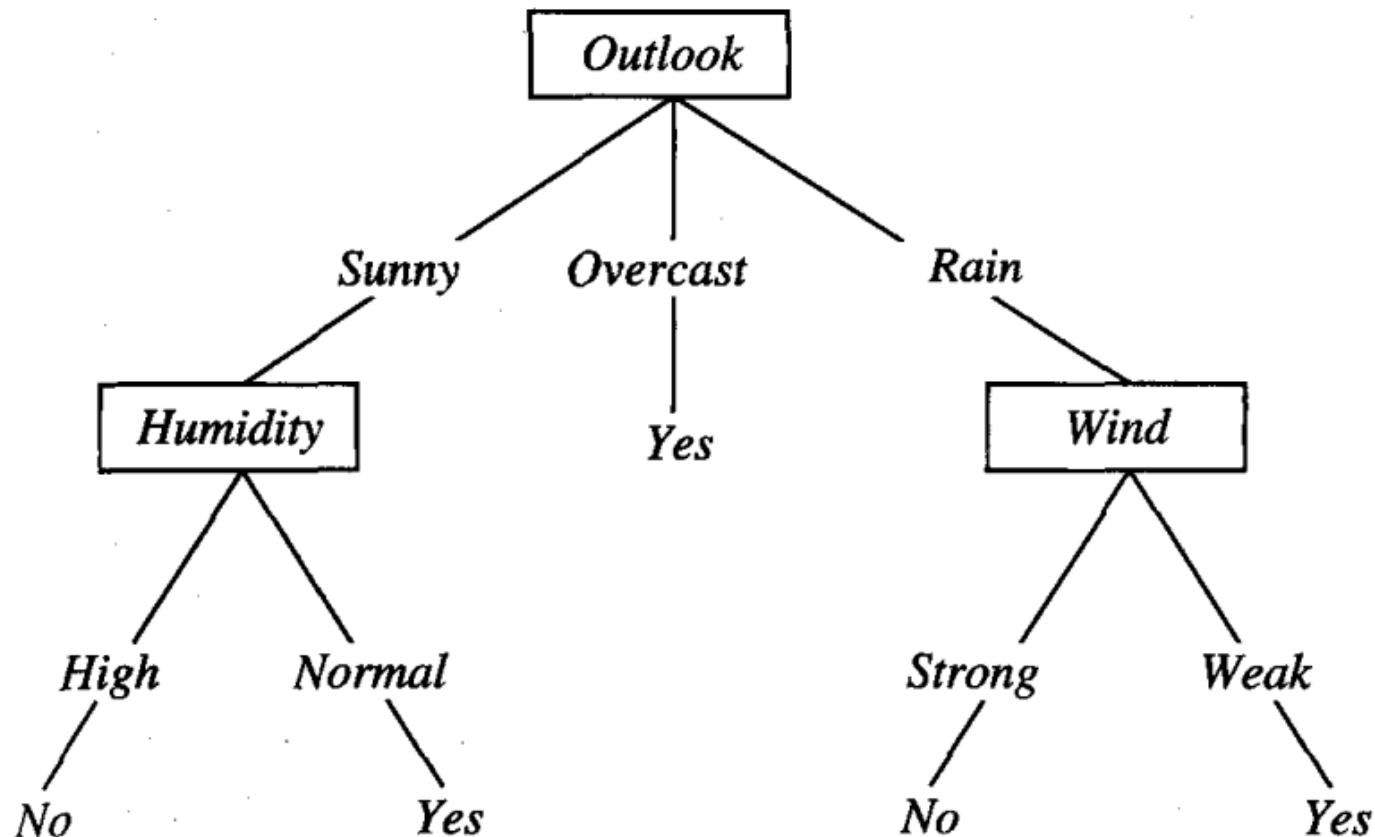
*(Outlook = Sunny, Temperature = Hot, Humidity = High, Wind = Strong)*



# Representación

- En general representan una disyunción de conjunciones de restricciones en los valores de los atributos de las instancias
- Cada camino desde la raíz hasta una hoja corresponde a una conjunción de pruebas en los atributos
- El ADD en sí mismo es una disyunción de las conjunciones

# Representación



$(Outlook = Sunny \wedge Humidity = Normal)$

✓  $(Outlook = Overcast)$

✓  $(Outlook = Rain \wedge Wind = Weak)$

Problemas apropiados en donde se puede usar aprendizaje basado en árboles de decisión

# Problemas apropiados

En **general** para problemas con las siguientes características

- Las instancias son representados por un par atributo-valor (e.g. temperatura: caliente, frío). La mejor situación, valores disjuntos (hay extensiones)
- La función objetivo  $f$  tiene valores discretos de salida ( $y$  discreto)
- Cuando se requiere de descripciones disyuntivas

# Problemas apropiados

En **general** para problemas con las siguientes características

- Los datos de entrenamiento  $\mathcal{D}$  pueden contener errores. Robustez a errores: clasificación de instancias de entrenamiento y errores en los valores de los atributos
- Los datos de entrenamiento pueden contener atributos con valores vacíos

# Problemas apropiados

## Ejemplos

- Aprender a clasificar pacientes médicos por sus enfermedades
- Mal funcionamiento de equipos por sus causas
- Créditos por su historial de pagos



# Algoritmo básico de aprendizaje de árboles de decisión

# ID3

- Aprende un ADD construyéndolo con un enfoque top-down
- ¿Qué atributo debe ser probado en la raíz?
- Cada atributo (solo) de las instancias es evaluado usando un test estadístico para determinar qué tan bien clasifica los ejemplos
- El mejor atributo es tomado para la prueba en la raíz
- Un nodo descendiente se crea por cada posible valor del atributo
- Los ejemplos se disponen apropiadamente debajo de cada rama
- Se repite el proceso usando los ejemplos asociados con cada nodo descendiente para seleccionar el mejor atributo para probar en ese punto del árbol

# ID3

- Es una búsqueda greedy (codiciosa/ambiciosa)
- El algoritmo nunca regresa para reconsiderar opciones anteriores
- Versión simplificada para aprender funciones de valores booleanos (aprendizaje de conceptos) ...

ID3(*Examples*, *Target\_attribute*, *Attributes*)

*Examples* are the training examples. *Target\_attribute* is the attribute whose value is to be predicted by the tree. *Attributes* is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given *Examples*.

- Create a *Root* node for the tree
- If all *Examples* are positive, Return the single-node tree *Root*, with label = +
- If all *Examples* are negative, Return the single-node tree *Root*, with label = -
- If *Attributes* is empty, Return the single-node tree *Root*, with label = most common value of *Target\_attribute* in *Examples*
- Otherwise Begin
  - $A \leftarrow$  the attribute from *Attributes* that best\* classifies *Examples*
  - The decision attribute for *Root*  $\leftarrow A$
  - For each possible value,  $v_i$ , of  $A$ ,
    - Add a new tree branch below *Root*, corresponding to the test  $A = v_i$
    - Let  $Examples_{v_i}$  be the subset of *Examples* that have value  $v_i$  for  $A$
    - If  $Examples_{v_i}$  is empty
      - Then below this new branch add a leaf node with label = most common value of *Target\_attribute* in *Examples*
      - Else below this new branch add the subtree  
ID3( $Examples_{v_i}$ , *Target\_attribute*,  $Attributes - \{A\}$ )
- End
- Return *Root*

---

\* The best attribute is the one with highest *information gain*, as defined in Equation (3.4).

# ID3

¿Qué atributo es el mejor clasificador?

- Seleccionar el atributo más útil para clasificar los ejemplos

Ganancia de información (information gain)

- Mide qué tan bien un atributo separa los ejemplos de entrenamiento de acuerdo a su clasificación objetivo ( $y$ )

# ID3

## Entropía y métrica de homogeneidad de ejemplos

### Entropía

- Caracteriza la (im)pureza de una colección  $S$

### Calificación binaria

$$\text{Entropía}(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

- $p_{\oplus}$  proporción de ejemplos positivos en  $S$
- $p_{\ominus}$  proporción de ejemplos negativos en  $S$
- $0 \log_2 0 = 0$

# ID3

## Ejemplo

- S es un colección de 14 ejemplos de un concepto **booleano** cualquiera

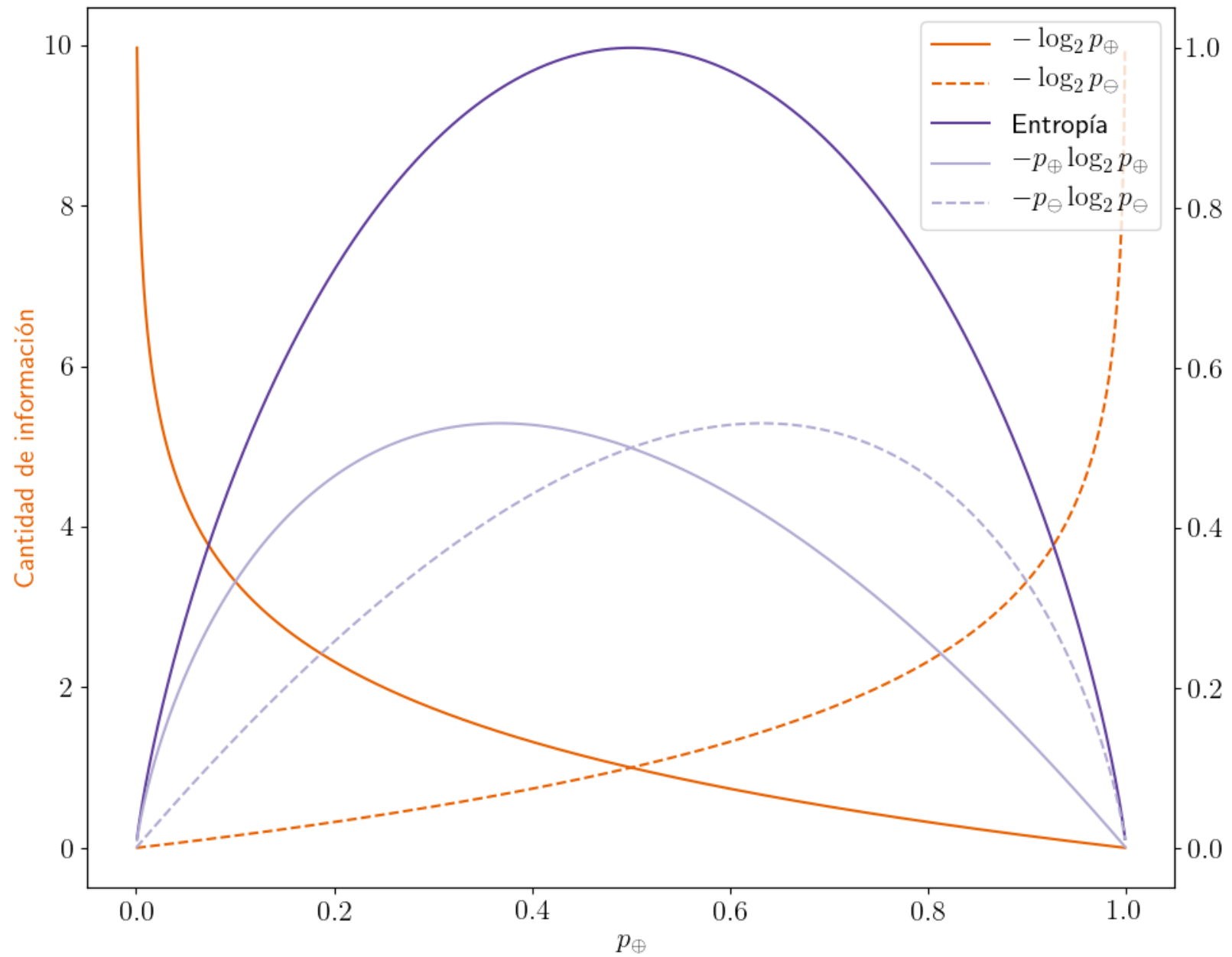
$[9+, 5-]$

$$\text{Entropía}([9+, 5-]) \equiv -\frac{9}{14} \log_2 \left( \frac{9}{14} \right) - \frac{5}{14} \log_2 \left( \frac{5}{14} \right)$$

$$\text{Entropía}([9+, 5-]) \equiv 0.94$$

- Intuición, valor entropía. Vea <https://goo.gl/K4c2Yk>

# Clasificación booleana





# ID3

## Una interpretación de entropía, desde la teoría de la información

- Especifica el número mínimo de bits de información necesarios para codificar la clasificación de un miembro arbitrario de  $S$  (tomado al azar con probabilidad uniforme)
- E.g.,  $p_{\oplus} = 1$ , el receptor conoce que el ejemplo tomado será positivo  $\rightarrow$  no se necesita enviar ningún mensaje (entropía = 0)
- $p_{\oplus} = 0.5$ , se requiere un bit para indicar si el ejemplo es + o -
- $p_{\oplus} = 0.8$ , la colección de mensajes puede ser codificado usando en promedio menos de 1 bit por mensaje, asignando códigos más cortos a colecciones de ejemplos positivos y código más largos para los, menos probables, ejemplos negativos

# ID3

Cuando la función objetivo no es booleana

- Si  $f$  puede dar  $c$  diferentes valores (clases)

$$\text{Entropía}(S) \equiv \sum_{i=1}^c -p_i \log_2 p_i$$

- $p_i$ : proporción de  $S$  perteneciente a la clase  $i$

# ID3

## Information gain

- Medida de efectividad de un atributo para clasificar los datos de entrenamiento
- Reducción esperada en entropía causada al dividir los ejemplos de acuerdo a un atributo

# ID3

## Information gain

- La Ganancia( $S, A$ ) de un atributo  $A$ , relativo a una colección  $S$  de ejemplos es

$$\equiv \text{Entropía}(S) - \sum_{v \in \text{Valores}(A)} \frac{|S_v|}{|S|} \text{Entropía}(S_v)$$

- $\text{Valores}(A)$ : conjunto de todos los posibles valores del atributo  $A$
- $S_v$ : subconjunto de  $S$  para el cual el atributo  $A$  tiene valor  $v$   
$$S_v = \{s \in S \mid A(s) = v\}$$

# ID3

## Information gain

$$\equiv \text{Entropía}(S) - \sum_{v \in \text{Valores}(A)} \frac{|S_v|}{|S|} \text{Entropía}(S_v)$$

Entropía de la colección original S



# ID3

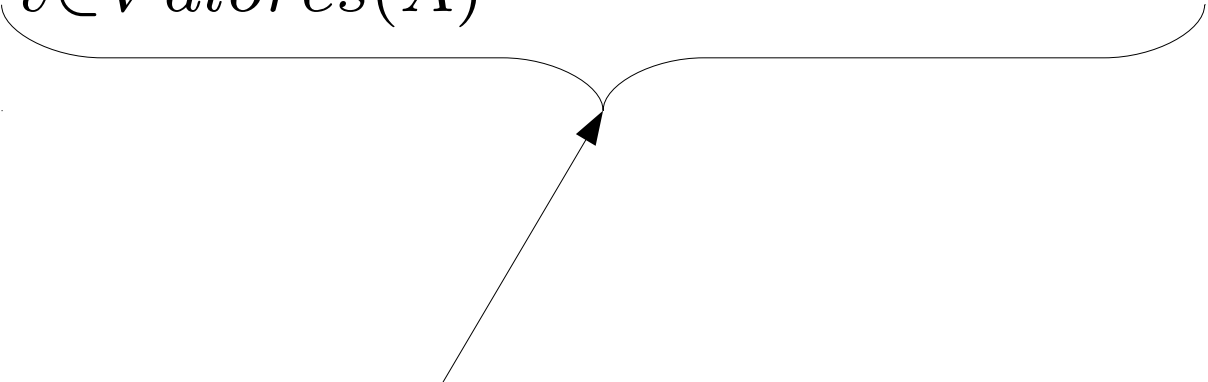
## Information gain

$$\equiv \text{Entropía}(S) - \underbrace{\sum_{v \in \text{Valores}(A)} \frac{|S_v|}{|S|} \text{Entropía}(S_v)}_{\text{Valor esperado de la entropía después de dividir } S \text{ usando el atributo } A}$$

Valor esperado de la entropía después de dividir  $S$  usando el atributo  $A$

# ID3

## Information gain

$$\equiv \text{Entropía}(S) - \sum_{v \in \text{Valores}(A)} \frac{|S_v|}{|S|} \text{Entropía}(S_v)$$


Suma de las entropías de cada subconjunto  $S_v$  con pesos iguales a la fracción de ejemplos de  $S$  que pertenecen  $S_v$

# ID3

## Information gain

$$\equiv \text{Entropía}(S) - \sum_{v \in \text{Valores}(A)} \frac{|S_v|}{|S|} \text{Entropía}(S_v)$$

Gain(S, A) es el número de bits ahorrados cuando se codifica el valor objetivo de un ejemplo arbitrario de S conociendo el valor del atributo A



# ID3

## Ejemplo

- S  
[9+, 5-]

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
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

**TABLE 3.2**

Training examples for the target concept *PlayTennis*.

# ID3

## Ejemplo

- $S = [9+, 5-]$
- $\text{Wind} = \text{Strong}$   
 $S_{\text{Strong}} = [3+, 3-]$
- $\text{Wind} = \text{Weak}$   
 $S_{\text{Weak}} = [6+, 2-]$

Wind	Play Tennis
Strong	No
Strong	No
Strong	No
Strong	Yes
Strong	Yes
Strong	Yes
Weak	No
Weak	No
Weak	Yes
Weak	Yes
Weak	Yes
Weak	Yes
Weak	Yes
Weak	Yes

# ID3

## Ejemplo

- $S = [9+, 5-]$

- $\text{Wind} = \text{Weak}$

$$S_{\text{Weak}} = [6+, 2-]$$

- $\text{Wind} = \text{Strong}$

$$S_{\text{Strong}} = [3+, 3-]$$

$$\text{Gain}(S, A)$$

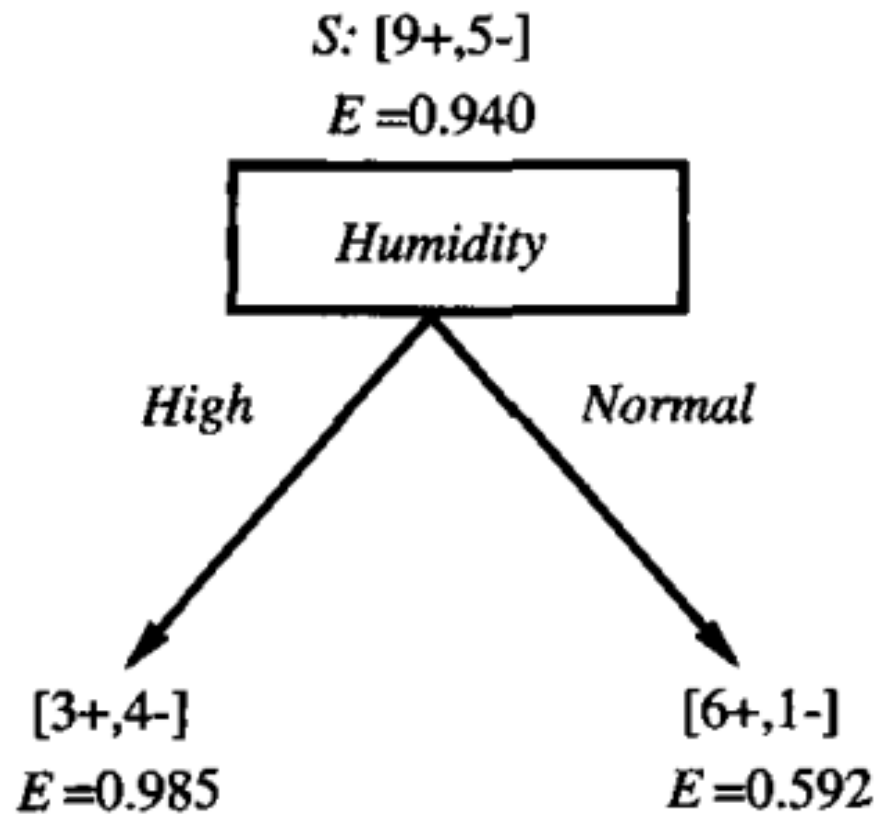
$$= \text{Entropy}(S) - \sum_{v \in \{\text{Weak}, \text{Strong}\}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$$= \text{Entropy}(S) - (8/14)\text{Entropy}(S_{\text{Weak}}) \\ - (6/14)\text{Entropy}(S_{\text{Strong}})$$

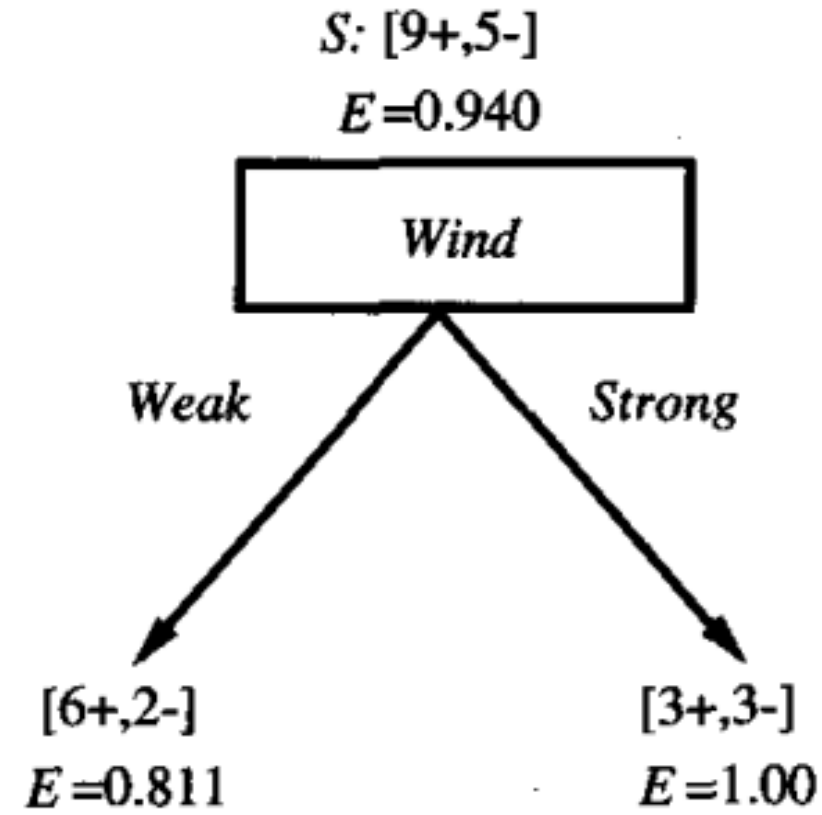
$$= 0.940 - (8/14)0.811 - (6/14)1.00$$

$$= 0.048$$

# ID3



$$\begin{aligned} \text{Gain}(S, \text{Humidity}) &= .940 - (7/14) \cdot .985 - (7/14) \cdot .592 \\ &= .151 \end{aligned}$$



$$\begin{aligned} \text{Gain}(S, \text{Wind}) &= .940 - (8/14) \cdot .811 - (6/14) \cdot 1.0 \\ &= .048 \end{aligned}$$

# ID3

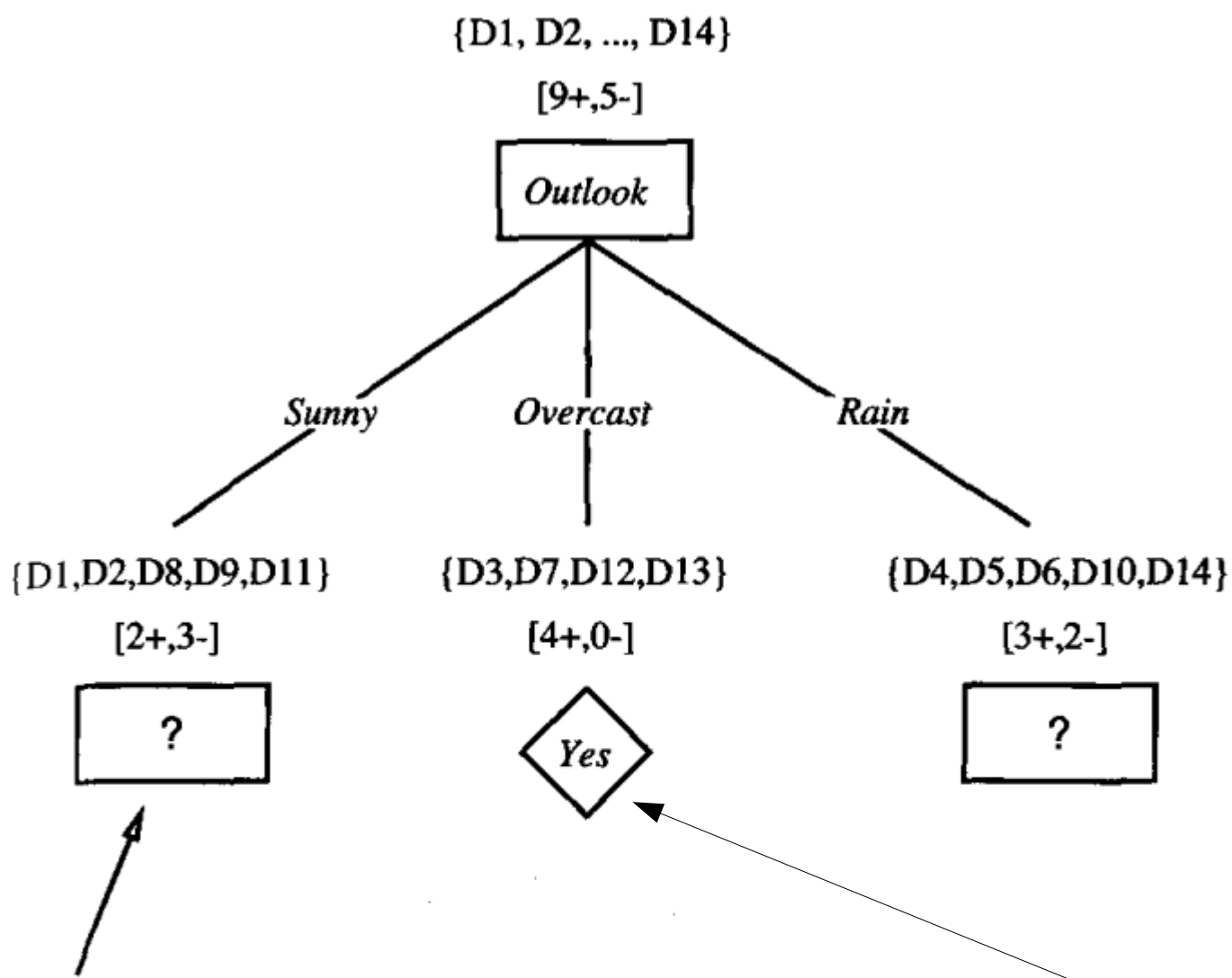
$$\text{Gain}(S, \text{Outlook}) = 0.246$$

$$\text{Gain}(S, \text{Humidity}) = 0.151$$

$$\text{Gain}(S, \text{Wind}) = 0.048$$

$$\text{Gain}(S, \text{Temperature}) = 0.029$$

Outlook	Temperature	Humidity	Wind	Play Tennis
Overcast	Cool	Normal	Strong	Yes
Overcast	Hot	High	Weak	Yes
Overcast	Hot	Normal	Weak	Yes
Overcast	Mild	High	Strong	Yes
Rain	Cool	Normal	Strong	No
Rain	Cool	Normal	Weak	Yes
Rain	Mild	High	Strong	No
Rain	Mild	High	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Cool	Normal	Weak	Yes
Sunny	Hot	High	Strong	No
Sunny	Hot	High	Weak	No
Sunny	Mild	High	Weak	No
Sunny	Mild	Normal	Strong	Yes



*Which attribute should be tested here?*

Hoja  $\equiv$  entropía cero

$$S_{\text{sunny}} = \{D1,D2,D8,D9,D11\}$$

$$\text{Gain}(S_{\text{sunny}}, \text{Humidity}) = .970 - (3/5) 0.0 - (2/5) 0.0 = .970$$

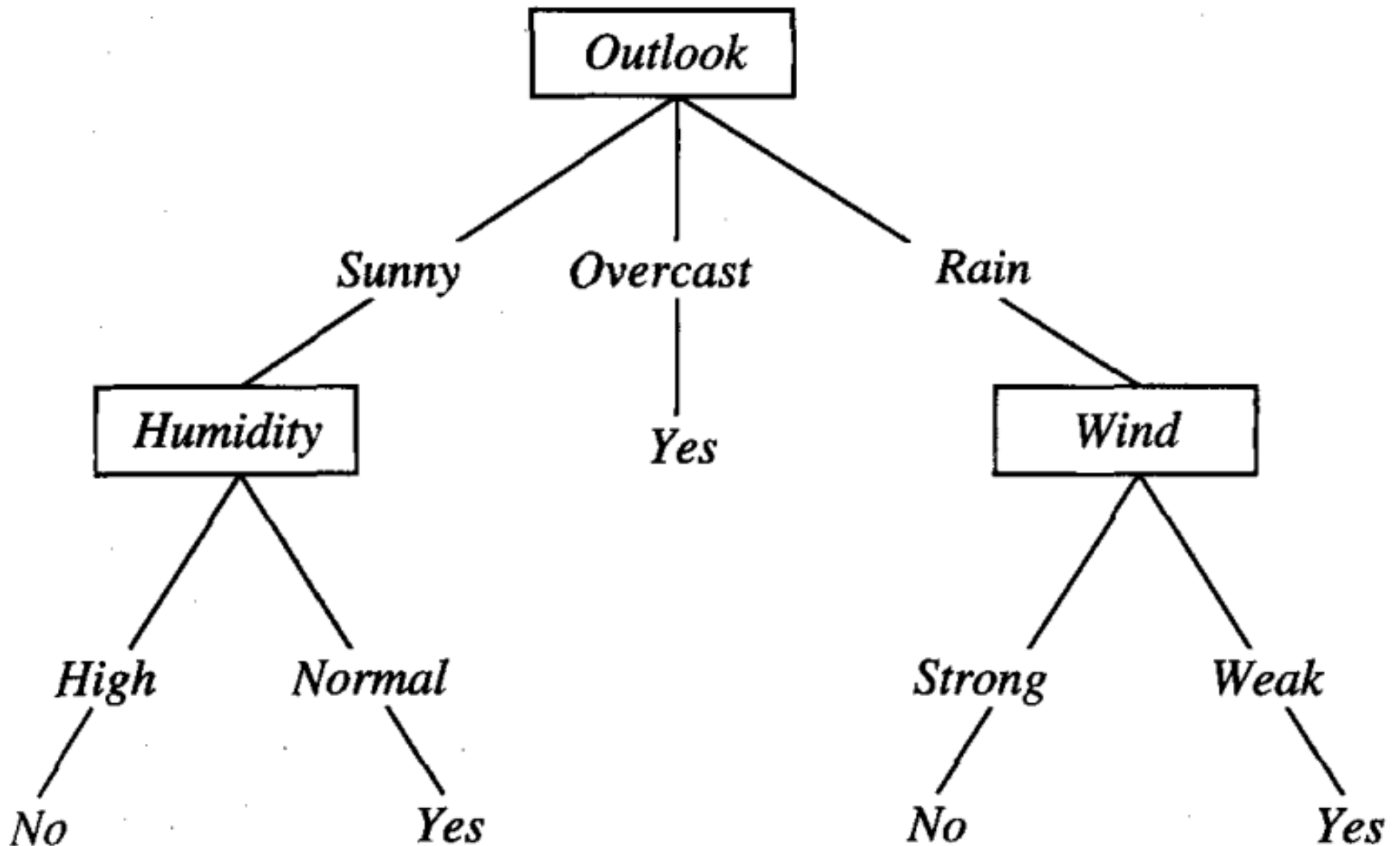
$$\text{Gain}(S_{\text{sunny}}, \text{Temperature}) = .970 - (2/5) 0.0 - (2/5) 1.0 - (1/5) 0.0 = .570$$

$$\text{Gain}(S_{\text{sunny}}, \text{Wind}) = .970 - (2/5) 1.0 - (3/5) .918 = .019$$

# ID3

- Proceso iterativo, hasta que
  - 1) Todo atributo haya sido incluido en el camino del árbol
  - 2) Los ejemplos de entrenamiento asociados con un nodo tengan todos el mismo atributo (entropía cero)

# ID3





# ID3

## Ejercicio

- Encuentre la  $g$  (aprenda el concepto *inflado*  $T/F$ ) que clasifique el  $\mathcal{D}$  de balones ( <http://archive.ics.uci.edu/ml/datasets/Balloons> )
- Encuentre un  $g$  para cada uno de los cuatro escenarios del experimento

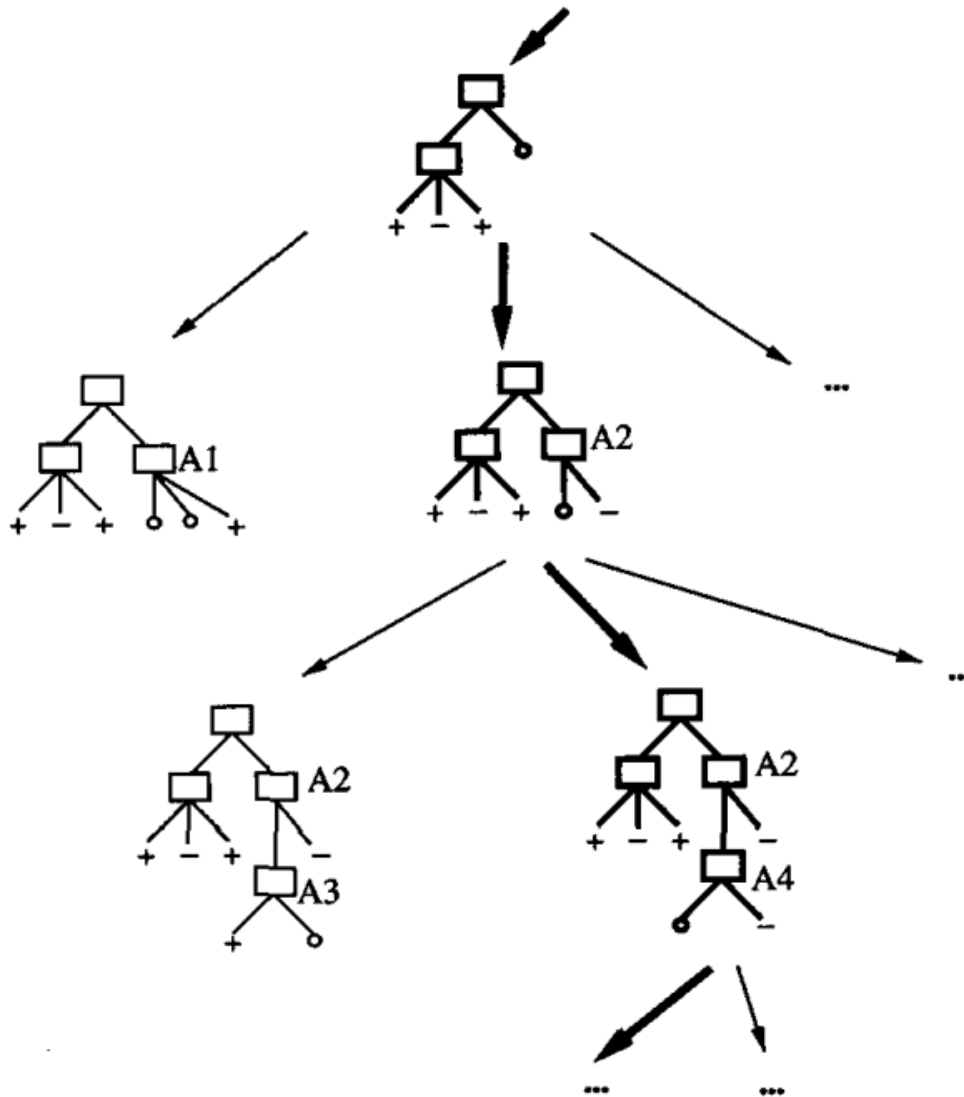
Fuente: Pazzani, M. (1991). The influence of prior knowledge on concept acquisition: Experimental and computational results. Journal of Experimental Psychology: Learning, Memory & Cognition, 17, 3, 416-432.

# Búsqueda en el hypothesis set

# Búsqueda en el hypothesis set

- Como otros métodos, ID3 puede ser caracterizado como una búsqueda en el espacio de hipótesis que trata de encontrar una hipótesis que se ajuste a  $\mathcal{D}$
- ID3 hace una búsqueda hill-climbing simple a compleja
- La función de error (función de evaluación / medida del error) que guía la búsqueda hill-climbing es la ganancia de información

# Búsqueda en el hypothesis set



**FIGURE 3.5**

Hypothesis space search by ID3. ID3 searches through the space of possible decision trees from simplest to increasingly complex, guided by the information gain heuristic.

# Búsqueda en el hypothesis set

## Capacidades y limitaciones de ID3

- Basado en el espacio de búsqueda y estrategia de búsqueda

## El hypothesis set es completo y finito

- Funciones de valores discretos, relativas a los atributos disponibles
- Toda función puede ser representada por un ADD
- Se evita un  $\mathcal{H}$  incompleto (que no contiene a  $f$ )

# Búsqueda en el hypothesis set

## Capacidades y limitaciones de ID3

Se mantiene una sola hipótesis en cada momento

- No representa en todo momento todas las hipótesis consistentes
- E.g. no puede determinar cuántos árboles alternativos son consistentes o indicar nuevas consultas de instancias que resuelvan cuál de esas hipótesis es la mejor

# Búsqueda en el hypothesis set

## Capacidades y limitaciones de ID3

En su forma pura (sin post-pruning) no hace backtracking

- Una vez elegido un atributo nunca hace backtracking para considerar otras opciones
- Susceptible a converger a un **óptimo local** (riesgo usual de búsquedas hill-climbing sin backtracking)
- Óptimo local: de un sólo camino explorado

# Búsqueda en el hypothesis set

## Capacidades y limitaciones de ID3

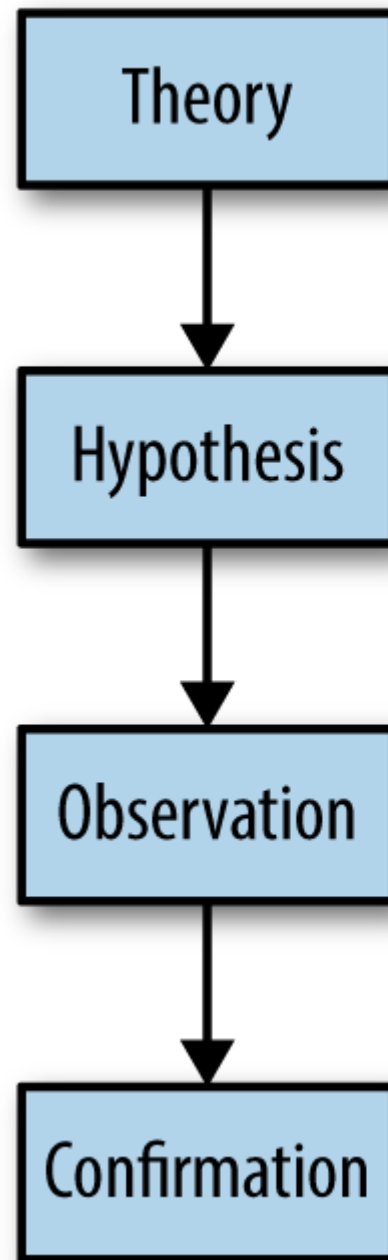
Usa todas las instancias en cada paso para hacer decisiones basadas en estadísticas

- Diferente al perceptrón por ejemplo (épocas)
- Ventaja: el resultado de la búsqueda es menos sensible a errores en instancias individuales
- Se puede modificar el criterio de terminación para aceptar  $h$ 's que no se ajustan **perfectamente** a los datos ( $\mathcal{D}$  con ruido)

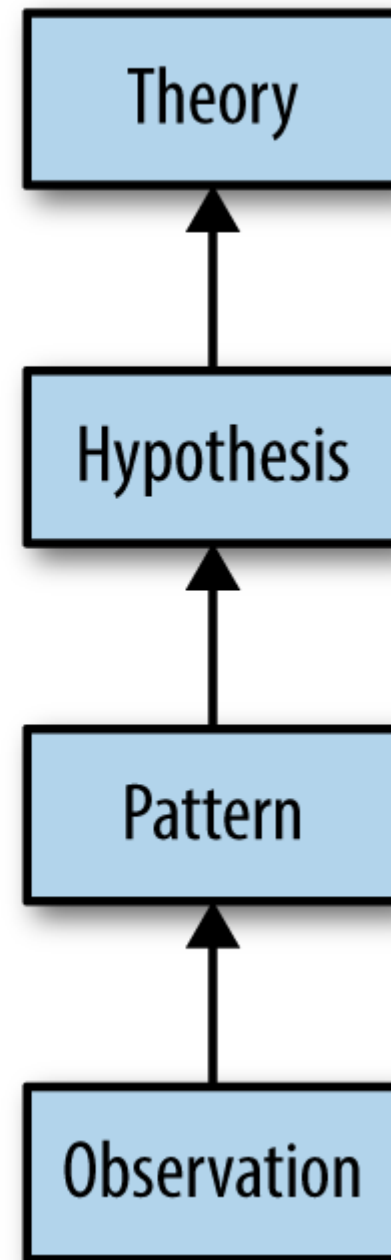


# Bias inductivo

## Deductive Reasoning



## Inductive Reasoning



# Generally speaking: Deduction vs Induction

- Deductive reasoning works from the more general to the more specific. Sometimes this is informally called a "top-down" approach. We might begin with thinking up a theory about our topic of interest. We then narrow that down into more specific hypotheses that we can test. This ultimately leads us to be able to test the hypotheses with specific data -- a confirmation (or not) of our original theories.
- Inductive reasoning works the other way, moving from specific observations to broader **generalizations** and theories. Informally, we sometimes call this a "bottom up" approach. In inductive reasoning, we begin with specific observations (the data) and measures, begin to **detect patterns and regularities**, formulate some tentative hypotheses that we can explore, and finally end up developing a general model.

# Bias inductivo

- Conjunto de suposiciones que, junto con  $\mathcal{D}$ , deductivamente justifica la clasificación asignada por el algoritmo de aprendizaje para instancias futuras
- Dado un  $\mathcal{D}$  típicamente algunos árboles consistentes con  $\mathcal{D}$
- El bias inductivo significa describir la base en la que se escoge uno u otro árbol consistente

# Bias inductivo

## Bias inductivo en ID3

- Escoge el primer árbol aceptable que se ajusta a los datos
  - 1) Selecciona árboles cortos en vez de grandes
  - 2) Selecciona árboles que tienen los atributos con mayor ganancia de información cerca a la raíz**
- **Occam's razor (ley de parsimonia)**: prefiera la hipótesis más simple (menos suposiciones) que se ajuste a los datos

# Cuestiones importantes en árboles de decisión

# Cuestiones importantes

## Cuestiones prácticas

- Determinar cuán profundo debe ser el árbol
- Trabajar con atributos continuos
- Escoger una medida de selección de atributos adecuada
- Trabajar con datos de entrenamiento que contengan valores vacíos en los atributos
- Atributos con diferentes costos
- Mejorar la eficiencia computacional

# Cuestiones importantes

## Evitar el **overfitting** de los datos

- ID3 inserta una rama tan profundamente como sea necesario para clasificar perfectamente los ejemplos de entrenamiento
- Parece lo razonable pero podría dar dificultades cuando hay ruido en  $\mathcal{D}$  o cuando  $|\mathcal{D}|$  es demasiado pequeño para reproducir una muestra representativa de  $f$



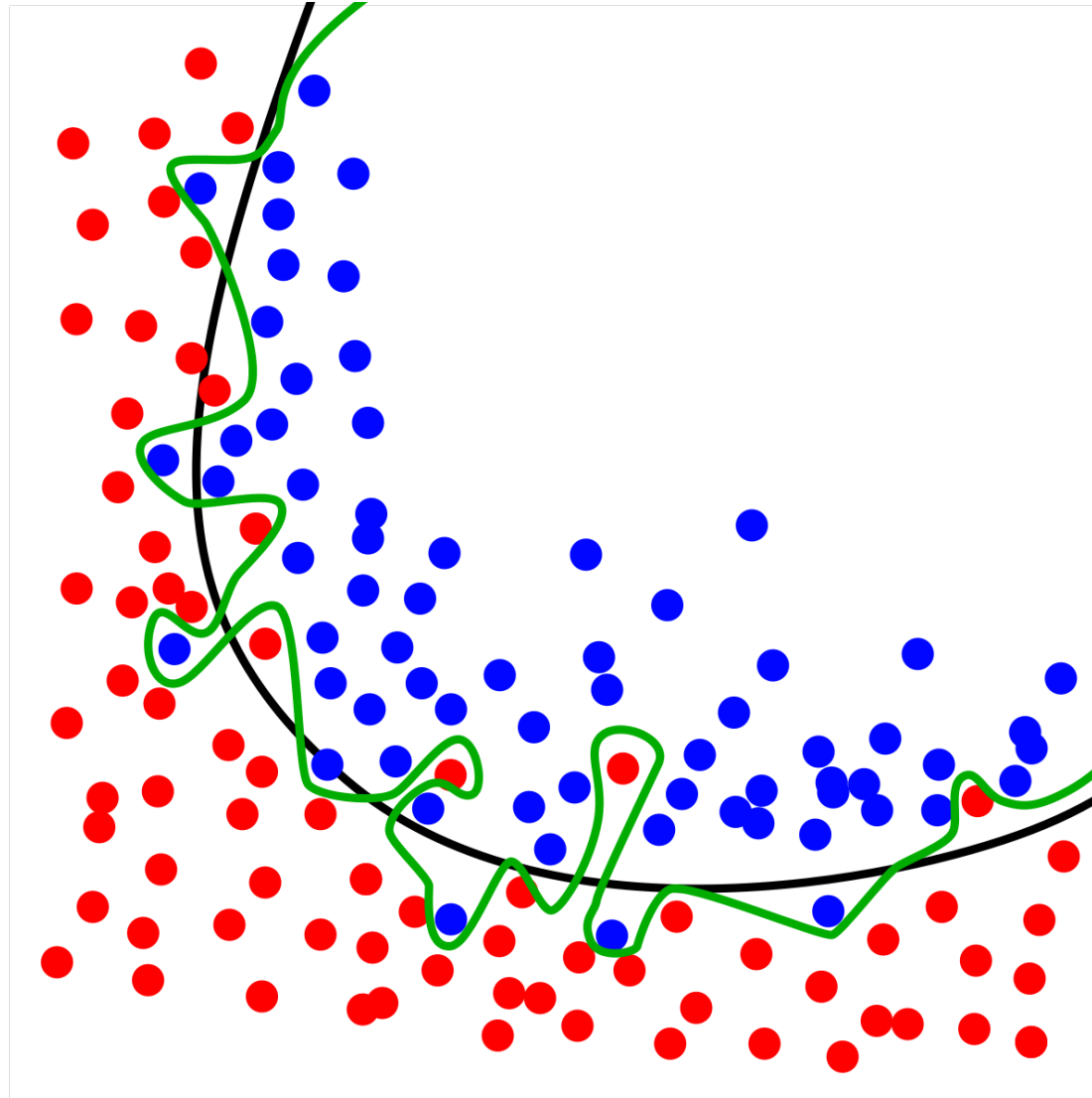
# Cuestiones importantes

## Evitar el overfitting de los datos

- Overfitting

Dado un hypothesis set  $\mathcal{H}$ , una hipótesis  $h \in \mathcal{H}$  se dice que sobre ajusta (overfit) los **datos de entrenamiento** si existe una hipótesis alternativa  $h' \in \mathcal{H}$ , tal que  $h$  tiene un error menor que  $h'$  en los ejemplos de entrenamiento, pero  $h'$  tiene un error menor que  $h$  en la distribución completa de instancias (más allá de  $\mathcal{D}$ )

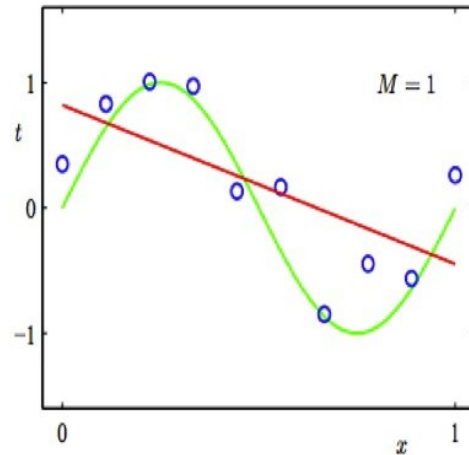
# Overfitting



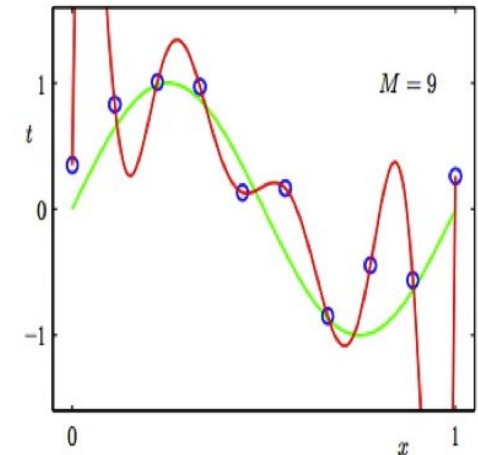
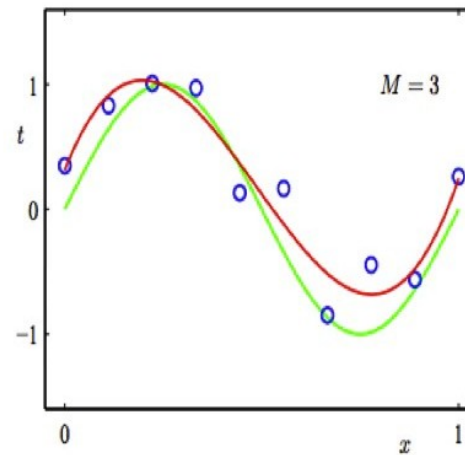
Src: <https://en.wikipedia.org/wiki/Overfitting>

# Overfitting

Regression:

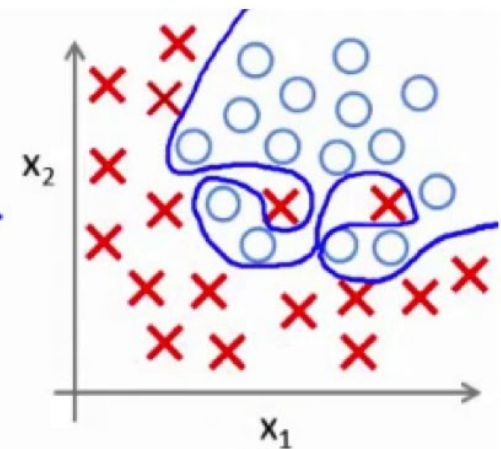
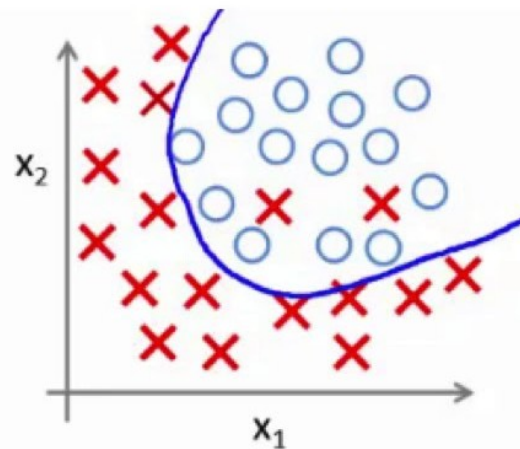
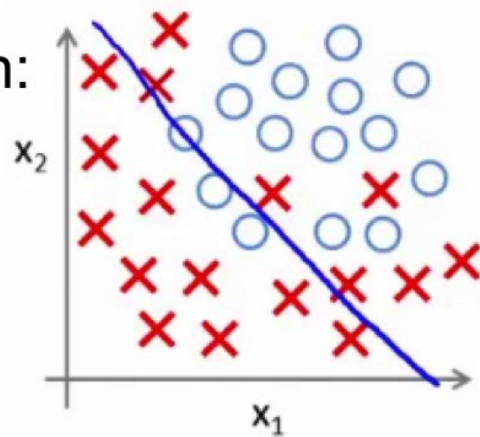


predictor too inflexible:  
cannot capture pattern



predictor too flexible:  
fits noise in the data

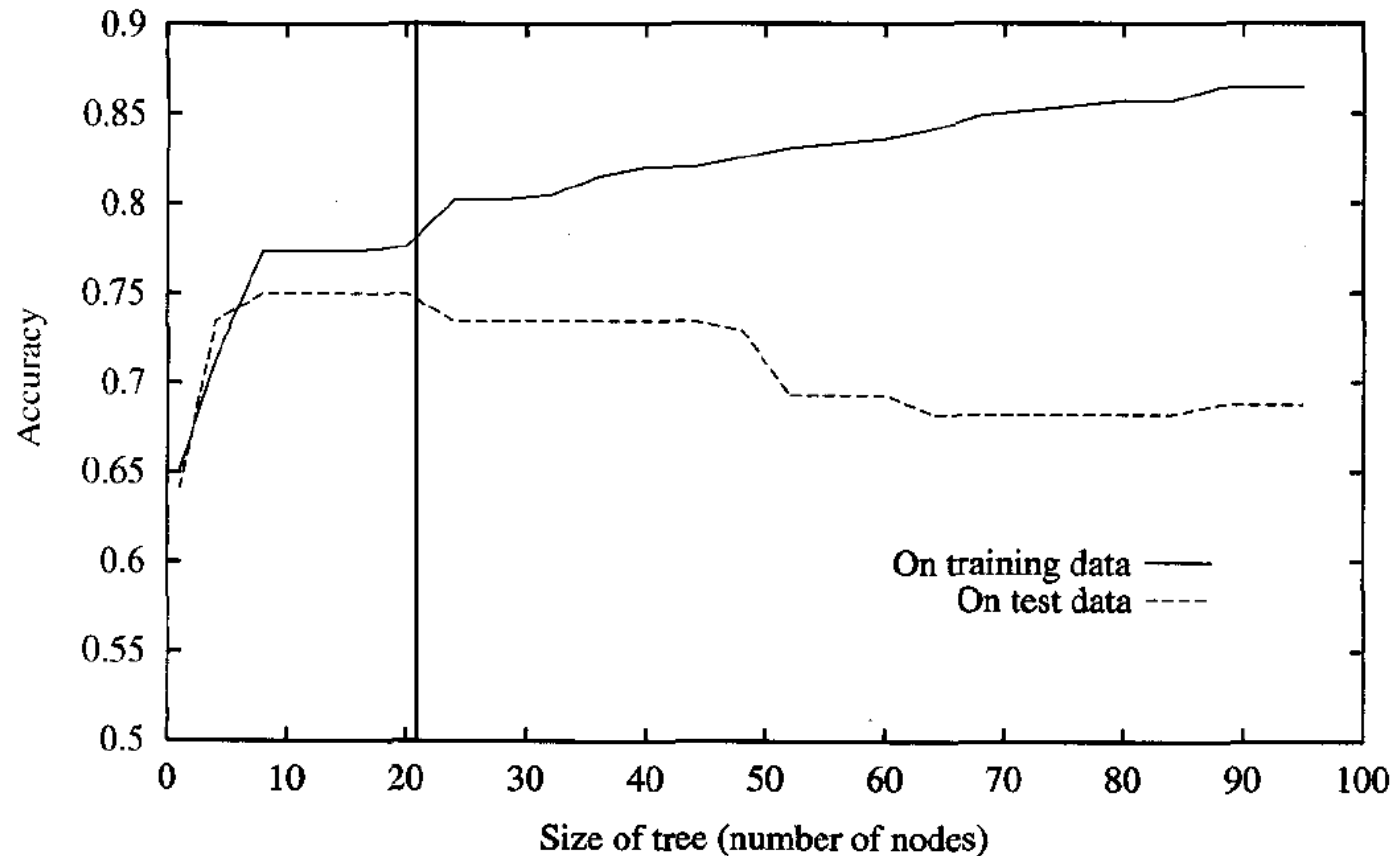
Classification:



Copyright © 2014 Victor Lavrenko

Src: <https://youtu.be/dBLZg-RqoLg>

# Overfitting



**FIGURE 3.6**

Overfitting in decision tree learning. As ID3 adds new nodes to grow the decision tree, the accuracy of the tree measured over the training examples increases monotonically. However, when measured over a set of test examples independent of the training examples, accuracy first increases, then decreases.

# Cuestiones importantes

## Evitar el overfitting de los datos

- Enfoques (divididos en dos grupos)
  - 1) Los que detienen el algoritmo antes de que se clasifiquen **perfectamente** los datos de entrenamiento
    - Difícil estimar cuándo parar al hacer crecer el árbol
  - 2) Los que permiten el overfitting y luego podan el árbol
    - Más exitosos en la práctica

# Cuestiones importantes

## Evitar el overfitting de los datos

- Sin importar qué enfoque, la pregunta es  
**¿Qué criterio usar para determinar el tamaño final correcto del árbol?**

# Cuestiones importantes

## Evitar el overfitting de los datos

- Enfoques
  - 1) Usar un set de ejemplos, distinto a los ejemplos de entrenamiento, para evaluar la utilidad de podar los nodos del árbol (**evaluation set**)
  - 2) Usar todos los datos pero aplicar un test estadístico para estimar si expandiendo o podando un nodo se producirá una mejora más allá de  $\mathcal{D}$
  - 3) Usar una **heurística** que indique la complejidad de codificar los ejemplos de entrenamiento y el árbol (Minimum Description Length principle)

# Cuestiones importantes

## Evitar el overfitting de los datos

- Enfoques

1) Usar un set de ejemplos, distinto a los ejemplos de entrenamiento, para evaluar la utilidad de podar los nodos del árbol

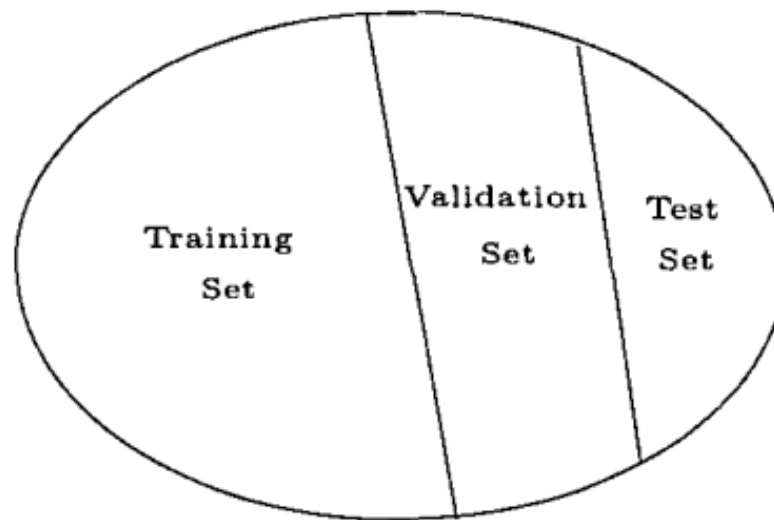
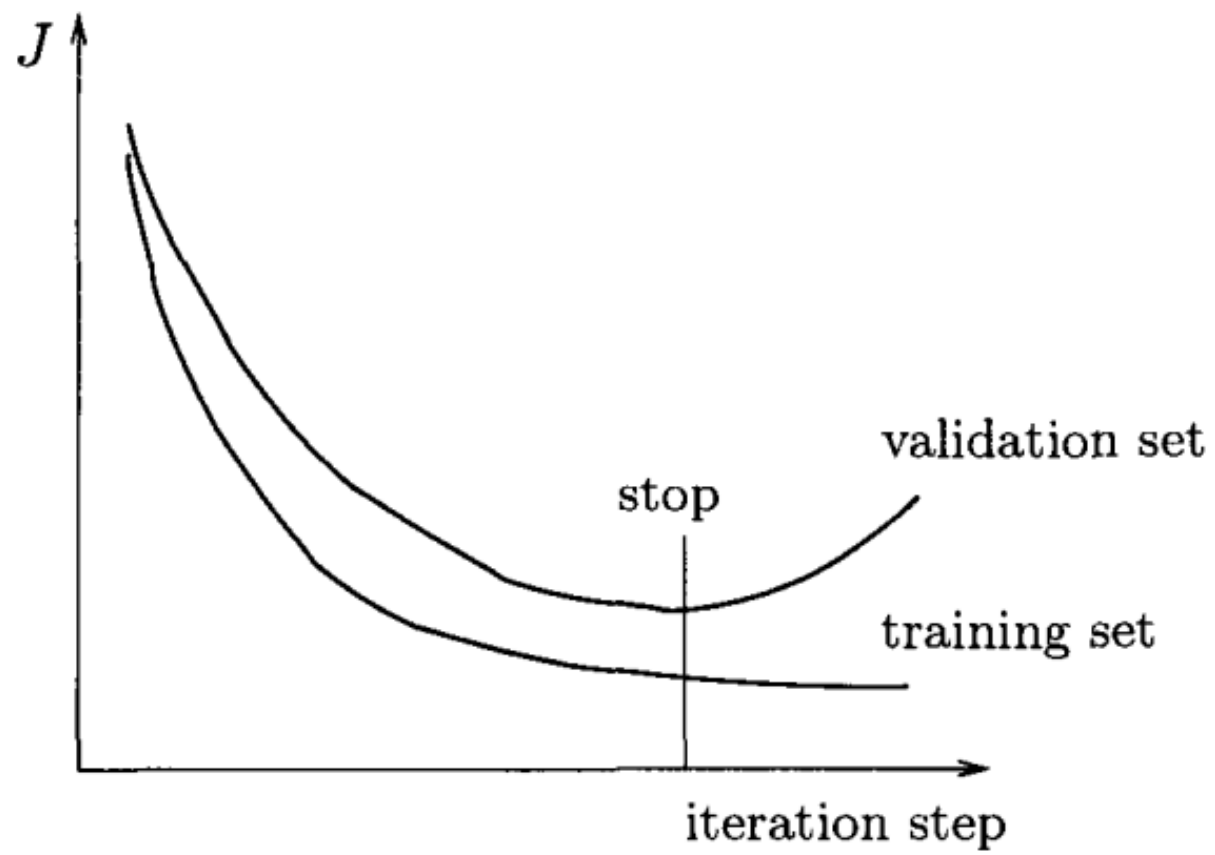
- El más común
- Referido como enfoque training y validation set
- El validation set debe ser lo suficientemente grande para proveer una muestra de instancias estadísticamente significativa
- Heurística común:  $\frac{1}{3}$  validation,  $\frac{2}{3}$  training



# Cuestiones importantes

## Evitar el overfitting de los datos

- Generalmente se divide los datos ( $\mathcal{D}$ ) en **training set**, **validation set** y **test set**
- El validation set se usa para decidir cuándo parar el entrenamiento = cuando el error es mínimo en el set de validación
- Los datos del test set se dejan totalmente aislados en el proceso de entrenamiento y validación
- Se pretende incrementar la **generalización**



Suykens et al. (2002)

# Cuestiones importantes

## Incorporar atributos de valores continuos

- Inicialmente ID3 con valores discretos
- Definir dinámicamente valores discretos que dividen los valores de los atributos continuos en un conjunto discreto de intervalos

---

<i>Temperature:</i>	40	48	60	72	80	90
<i>PlayTennis:</i>	No	No	Yes	Yes	Yes	No

---

- $Temperature > 54$ ,  $Temperature > 85$
- Information gain

# Cuestiones importantes

## Medidas alternativas para seleccionar atributos

- Information gain

Sesgo que favorece a atributos con muchos valores por encima de aquellos con pocos valores

## Alternativas

- Gain ratio
- Gini index
- Error de clasificación
- Etc.

# Cuestiones importantes

## Ejemplos de entrenamiento con atributos sin valor

### Lo más simple

- Asignar el valor que es más común entre los ejemplos de entrenamiento en el nodo  $n$
- O asignar el valor más común entre los ejemplos en el nodo  $n$  que tienen la clasificación  $c(\mathbf{x})$

### Procedimiento más complejo

- Asignar una probabilidad a cada posible valor de  $A$  en vez de asignar el valor más común a  $A(\mathbf{x})$
- C4.5

# Cuestiones importantes

## Atributos con diferentes costos

- Algunas alternativas que modificar la función de ganancia de información (reducción de entropía)

$$\frac{Gain^2(S, A)}{Cost(A)}$$

$$\frac{2^{Gain(S, A)} - 1}{(Cost(A) + 1)^w}$$

# Ejercicio

- Revisar el subcapítulo *Decision tree learning* (pág 80-92) del libro de Raschka (2016)
- Ponga especial atención en los otros índices de pureza/impureza de información que se pueden usar
- Reproduzca todos los ejemplos que se presentan
- Entienda muy bien cómo trabajar con ADD y random forests en `sklearn`

# Conceptos y términos importantes



# Conceptos y términos importantes

- Los ADD proveen un método práctico para aprender conceptos
- Funciones de valores discretos
- ID3 busca en el  $\mathcal{H}$  completo lo que evita que se consideren sets restringidos donde no podría estar  $f$
- El bias inductivo de ID3 incluye una preferencia por árboles pequeños
- Overfitting
- Extensiones de ID3 que incluyen post-prunning, valores reales, atributos sin valor, aprendizaje online, costos para los atributos, otras medidas de ganancia de información, etc

# Referencias

- Mitchell, T.M., 1997. **Machine Learning**, McGraw-Hill series in computer science. McGraw-Hill, New York.
- Suykens, Johan A. K., Tony Van Gestel, Jos De Brabanter, Bart De Moor, and Joos Vandewalle, eds. **Least Squares Support Vector Machines**. River Edge, NJ: World Scientific, 2002.
- Raschka, S., 2016. **Python machine learning**, Community experience distilled. Packt Publishing, Birmingham Mumbai.

# Preguntas

