

# Curso de Inteligencia Artificial

## **Unidad VI: Aprendizaje**

Jose Aguilar

CEMISID, Facultad de Ingeniería

Universidad de los Andes

Mérida, Venezuela

aguilar@ula.ve

# Aprendizaje

Por mas formalismos de representación y razonamiento,  
nunca se abarcará **todo el conocimiento posible**

- Comportamiento inteligente: capacidad de adaptarse o aprender de experiencias

Un sistema **aprende** cuando es *capaz de* experimentar *modificaciones estructurales y/o funcionales*, de acuerdo con la *experiencia*, en vista a conseguir *mayor eficacia* en su interacción con el medio

G. Pajare, M. Santos

# Aprendizaje

- Se utiliza para :
  - Memorización de algo
  - Aprendizaje de hechos con la observación o la exploración
  - Desarrollo de habilidades cognitivas a través de la práctica
  - Reorganización del conocimiento existente
  - Incorporación de nuevo conocimiento

El aprendizaje denota *cambios adaptativos* en el sistema en el sentido que permiten al sistema *hacer la(s) tarea(s)* más *eficiente y eficazmente* la próxima vez.

H. Simon

# Aprendizaje

- Esencial para **ambientes desconocidos**
- Es un método de **construcción de sistemas**
- **Modifica los mecanismos de decisión** de los agentes para mejorar su comportamiento
- Única forma para que **agente adquiera por si solo lo que requiere**
- **Base de la autonomía de un agente**
- Forma automática de **adquisición del conocimiento**

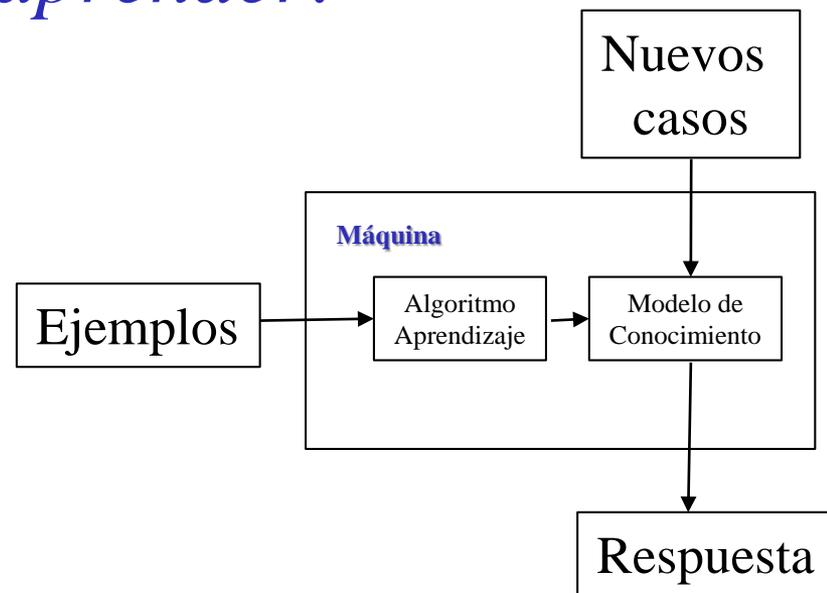
# ¿Qué se deriva del Aprendizaje?

- **Incrementa el conocimiento** que se posea
  - Adquisición del conocimiento
  - Capacidad de memorizar
  - Adaptación al entorno
- **Mejora la exactitud de la solución**
  - El objetivo es la síntesis (definición) de una estructura nueva
  - Supone realizar una inferencia (inductiva, abductiva)
- **Mejora el tiempo de calculo de una solución**
  - El objetivo es el análisis (explicación) de una observación
  - Supone realizar una inferencia deductiva

# Aprendizaje automático o aprendizaje de máquinas (del inglés, "Machine Learning")

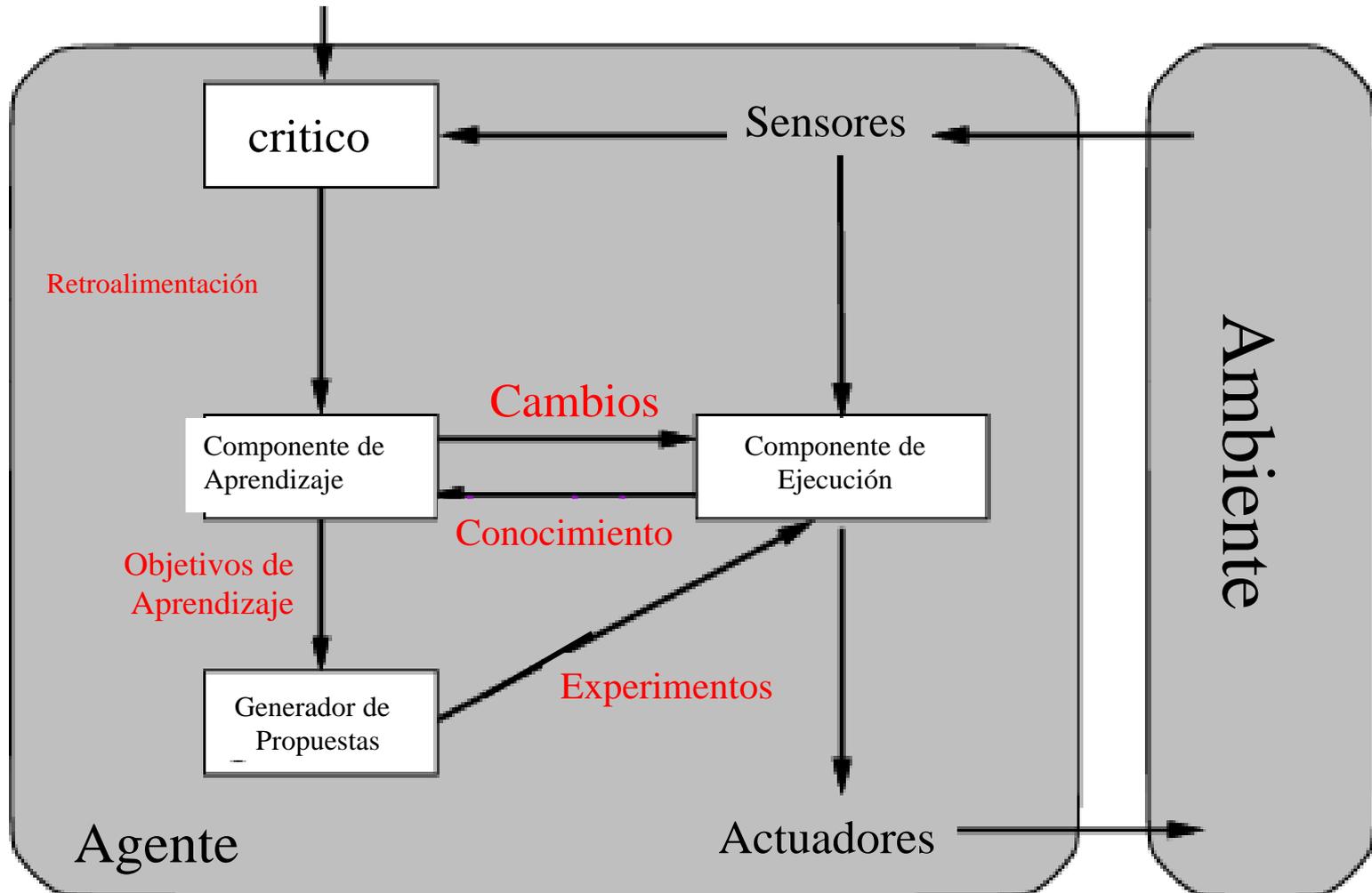
Rama de la inteligencia artificial que desarrolla técnicas que permitan a las computadoras *aprender*.

- Programas capaces de generalizar comportamientos a partir de una información no estructurada suministrada en forma de ejemplos.
- Proceso de generación del conocimiento.



# Agente con aprendizaje

Medida de Rendimiento



# Un *componente de ejecución* puede incluir

- Un mapeo de las condiciones actuales a acciones
- Un medio para inferir propiedades relevantes del mundo según lo que se percibe
- Usa
  - Información de cómo evoluciona el mundo
  - Información sobre resultados de las posibles acciones
  - Información indicando lo deseable de cada estado
  - Información de que tan deseable es una acción en un estado en particular
- Tiene objetivos a optimizar

# *Componente de aprendizaje*

- Cuales **componentes** del componente de ejecución deben ser aprendidos
- Cual es la **representación** usada para esos elementos
- Cual es el **mecanismo** de retro-alimentación disponible
  - Aprendizaje Supervisado
  - Aprendizaje No supervisado => agrupamiento
  - Aprendizaje Reforzado
- Que **conocimiento-a-priori** esta disponible

# Tipos de APRENDIZAJE

- **APRENDIZAJE ANIMAL**
- **APRENDIZAJE AUTOMÁTICO**
  - **Deductivo**
    - » **Generalización.**
  - **Inductivo**
    - » **Adquisición de componentes (supervisado).**
    - » **Por observación (no supervisado).**
  - **Por analogía**

# Aprendizaje animal

- Es también difícil enunciar una definición, así que se definen una serie de características:
  - **Habitación:** El sistema aprende a no prestar atención de ciertas entradas y nos habituamos a ello y no lo consideramos importante
  - **Aprendizaje asociativo:** Aprenden asociando elementos que están asociados (perro, comida y campanilla).
  - **Impronta:** Durante un periodo de tiempo de su vida todo lo que el ser ve lo sigue como a una madre.
  - **Imitación:** Consiste en imitar el comportamiento de otro ser.

# Aprendizaje

Sentencia clásica:  $P \vee BC \Rightarrow C$

- P: puede ser premisas, sentencias
- BC: es base de conocimiento o información previa
- C: puede ser consecuente, observación, datos de entrada
- Aprendizaje tiene dos elementos:
  - Objetivo de Aprendizaje (O)
  - Información inicial (I)
- Inferencia inductiva en el aprendizaje:
  - $O=P$
  - $I=C, BC$

**Descubro desde lo que observo**
- Inferencia deductiva en el aprendizaje :
  - $I=P, BC$
  - $O=C$

**Construyo desde experiencia y conocimiento previo**

# Aprendizaje Inductivo

*consiste* en inducir información de una concepto a partir de un conjunto de cosas en concreto.

- No requiere información previa del dominio.

Ejemplo: ese gato tiene 4 patas -> todos los gatos tienen 4 patas.

- Si veo un caso supongo que todos los casos son iguales hasta que encuentre una contradicción que me obligue a remodelar las informaciones.
- Se realiza una estructura del dominio a base de inducciones.

# Tipos de Aprendizaje

## Aprendizaje inductivo

Basados en observaciones o ejemplos

- Ejemplo:
  - El pedazo de pan número 1 me alimento cuando yo lo comí.
  - El pedazo de pan número 2 alimentaba cuando yo lo comí.
  - El pedazo de pan número 3 alimentaba cuando yo lo comí.
  - El pedazo de pan número 100 alimentaba cuando yo lo comí.
- Por lo tanto,  
**todos los pedazos de pan me alimentan cuando los como (P)**

# Tipos de aprendizaje inductivo

Hay fundamentalmente dos tipos:

- *Adquisición de conceptos* (aprendizaje inductivo supervisado).
  - Contamos con una serie de elementos ya clasificados y sabemos a que clase pertenece cada elemento.
  - Nuestro objetivo es obtener un descriptor apropiado de cada clase.
- *Por observación (no supervisado)*.
  - Tenemos todo el dominio y debemos agrupar los elementos de nuestro sistema.

# Aprendizaje Deductivo

Mediante procesos de inferencia deductiva aplicando a hechos reglas conocidas.

- Necesita conocimientos previos.
- Ejemplo: hay colillas -> han fumado.
- Componentes:
  - » Resolución de problemas.
  - » Análisis de la traza.
  - » Filtrado.
  - » Generalización.
  - » Construir nueva información.
  - » Incorporar.

# Tipos de Aprendizaje

## Aprendizaje deductivo

Basado en explicaciones, hipótesis, premisas

– Ejemplo:

- dada una regla sintáctica en un lenguaje de una oración  $(O=S+V) \Rightarrow$  **Premisa**,
- y conociendo otras reglas (por ejemplo la concordancia de genero, la posibilidad de varios sujetos, etc.)  $\Rightarrow$  **BC**,
- el objetivo es encontrar una mejor definición de la oración (**O(C)**) que tenga más sentido en el contexto dado

## • Otros tipos de Aprendizaje

- Por analogía  $\Rightarrow$  Basado en casos
- Por imitación
- Multiestrategías
- Algoritmos genéticos
- Conexionistas

# Clases de aprendizaje en Agentes

- **Aprendizaje Supervisado**

Dado un conjunto de ejemplos de pares de entrada/salida, encontrar una regla que prediga la salida asociada con una nueva entrada

- **Aprendizaje No supervisado => agrupamiento**

Dado un conjunto de ejemplos, pero sin etiqueta, agruparlos o caracterizarlos naturalmente,

- **Aprendizaje Reforzado**

Un agente interactúa con su ambiente haciendo observaciones y tomando acciones, y es recompensado o penalizado por ellas. El aprende a escoger las acciones que lo recompensen mas

# Aprendizaje Inductivo

Dada una colección de ejemplos de  $f$ , regresa una función  $h$  que lo aproxime

- La mas simple forma: aprender una función desde ejemplos

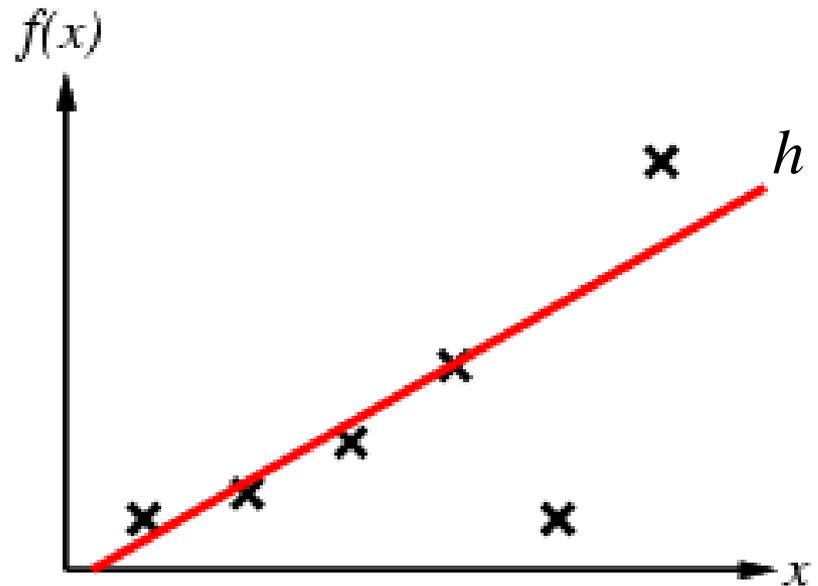
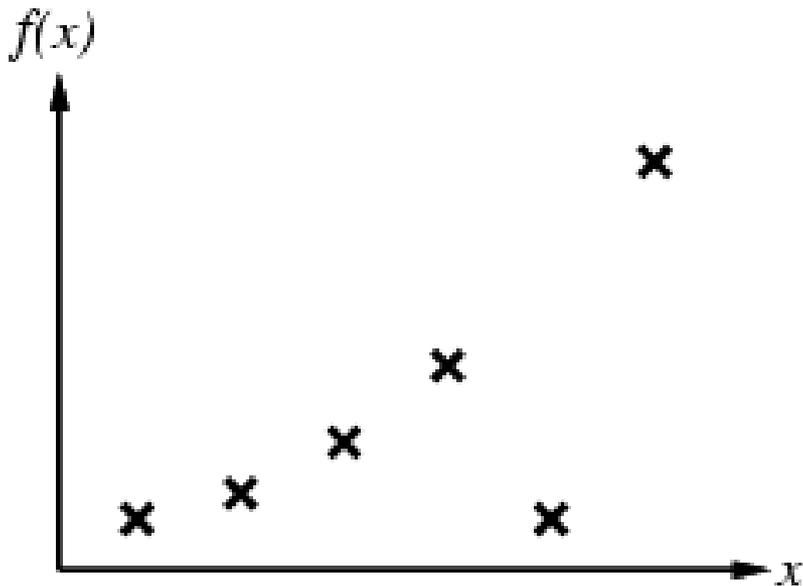
$f$  es la **función objetivo**

Un **ejemplo** es  $(x, f(x))$

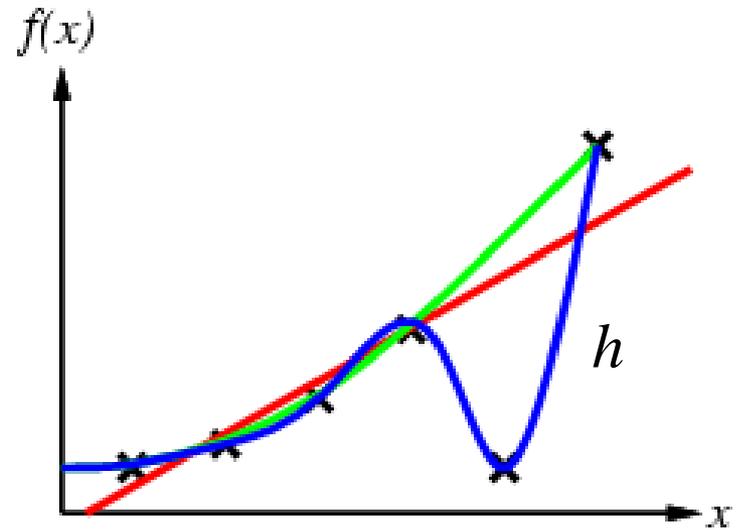
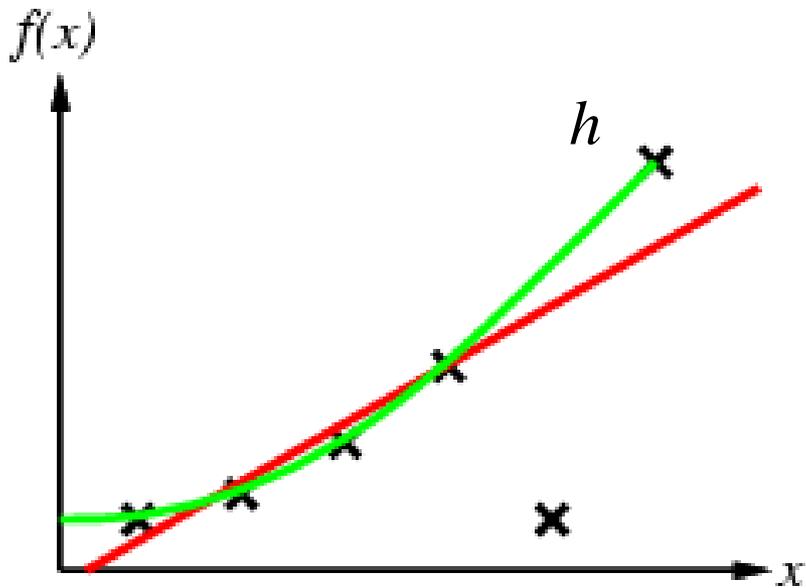
- Problema: encontrar una **función hipotética**  $h$   
Tal que  $h \approx f$   
Dado un **conjunto de** ejemplos
- Hipótesis
  - Ignora conocimiento a prior (en algunos casos)
  - Asume que los ejemplos son dados

# Aprendizaje Inductivo

Construir  $h$  tal que se acerque a  $f$  a través de un conjunto de entrenamientos



# Aprendizaje Inductivo



# Aprendizaje Inductivo

## Agente de Aprendizaje Simple Reflexivo

### Componente de Ejecución:

- Si (percepción, a) en ejemplos entonces
  - regresa a
- de lo contrario
  - $h = \text{induce}(\text{ejemplo})$
  - regresa  $h(\text{percepción})$

Aprendizaje!!

### Componente de Aprendizaje:

- $\text{ejemplo} = \text{ejemplo} + \{(\text{percepción}, \text{acción})\}$
- Estrategia de aprendizaje inductiva

# Árbol de Decisión

**Toma como entrada una situación y da como salida una decisión (por ejemplo: si/no)**

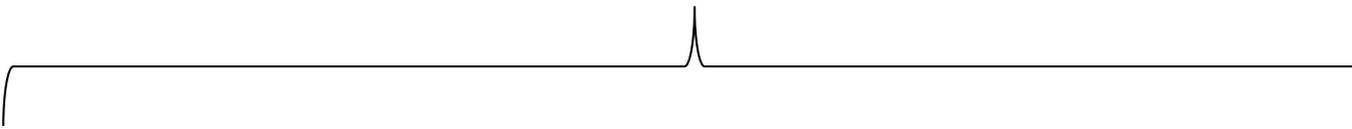
- **Ejemplo: decidir si esperar o no por una mesa en un restaurant basado en los siguientes criterios**
  1. ¿Hay otro restaurant cerca? (Alternativa)
  2. ¿Hay un bar confortable para esperar? (Bar)
  3. ¿Hoy es Viernes o Sábado? (Día)
  4. ¿Hay hambre? (EdoM)
  5. Numero de personas en el restaurant (Patrón: Vacio, Algo, lleno)
  6. ¿Precio? (\$, \$\$, \$\$\$)
  7. ¿Esta lloviendo? (Edo.D)
  8. ¿Se tiene una reservación?
  9. Tipo de restaurant (Francés, Italiano, Japonés, Hamburguesa)
  10. Tiempo de espera estimado (0-10min, 10-30min, 30-60, >60min)

# Árbol de Decisión

## Ejemplos

Criterios

¿Qué aprendo?



Ej	Alt	Bar	Dia	EdM	Patr	Prec	EdD	Tipo	RES	T --->	Espera
X1	S	N	N	S	Alg	\$\$\$	N	Franc	S	0-10	S
X2	S	N	N	S	llen	\$	N	Jap	S	10-15	N
X3	N	S	N	N	Alg	\$	N	Hamb	N	0	S
...											
X12	S	S	S	S	llen	\$	N	Hamb	N	10	S

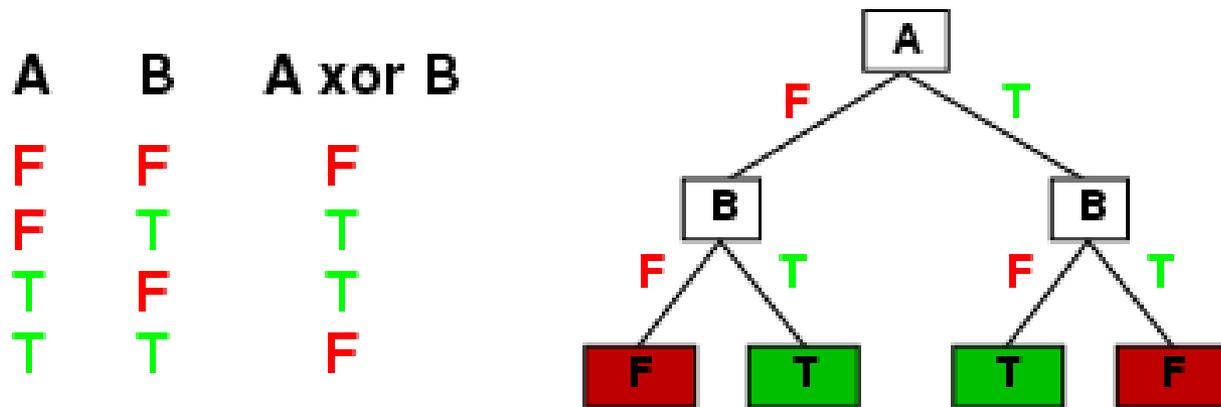
# Tablas de decisión

Forma más simple y más rudimentaria para representar la salida de la máquina de aprendizaje.

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

# Árbol de Decisión

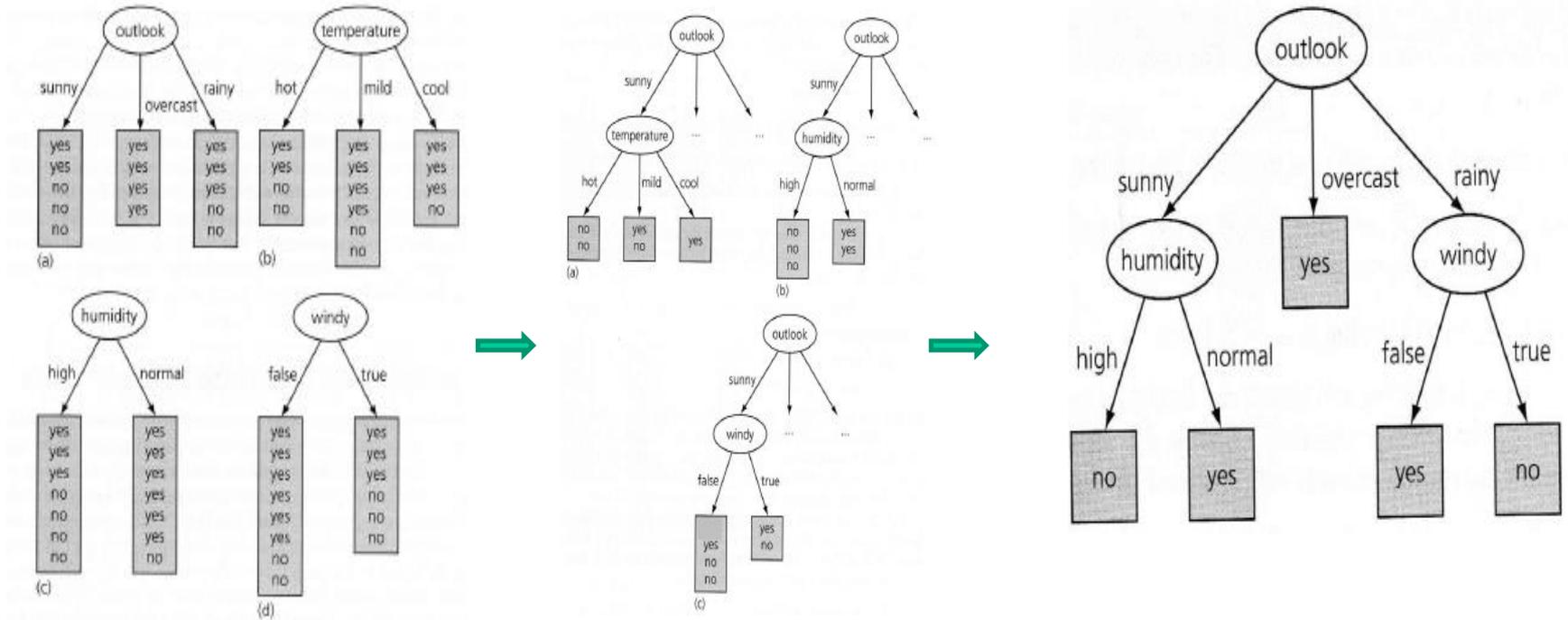
- Puede expresar cualquier función a partir de sus atributos de entrada.
- Un árbol de decisión es consistente para cualquier conjunto de entrenamiento, cuando hay un **camino a una hoja para uno o varios ejemplos**
- Basado en la idea de **tablas de la verdad**:



Es una estrategia de aprendizaje inductivo

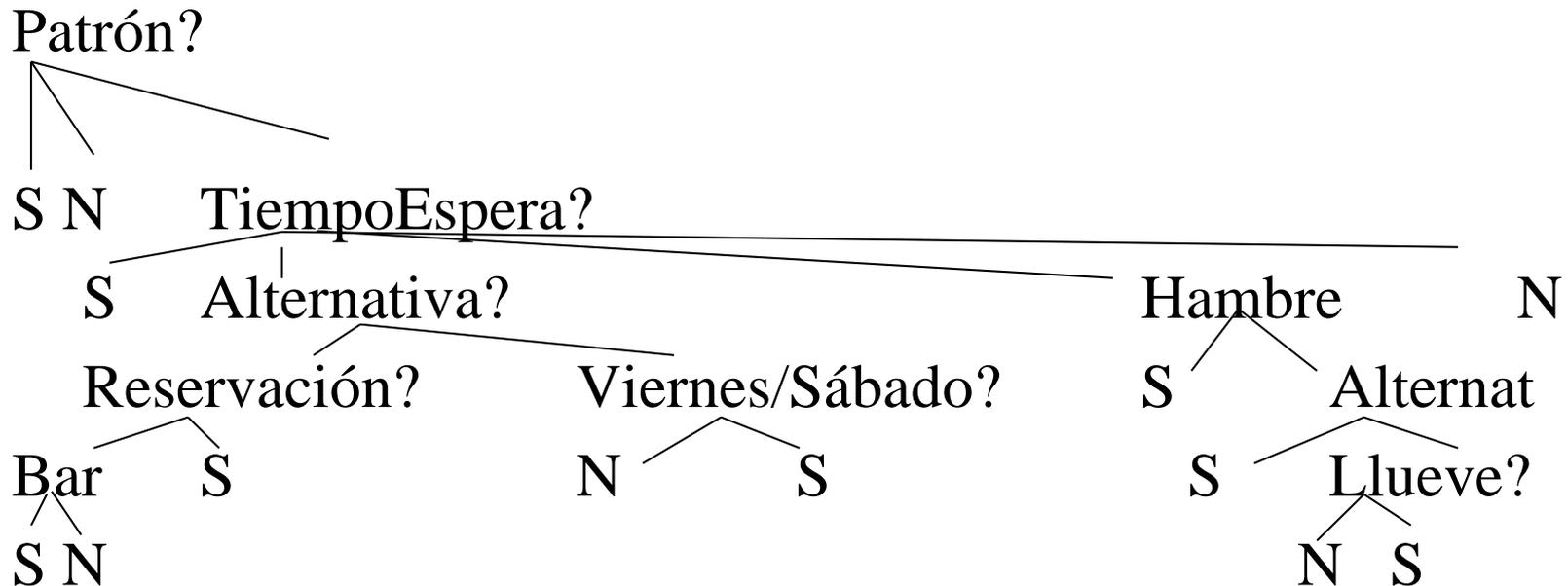
# Arboles de decisión

## Transformaciones



# Árbol de Decisión

Para nuestro ejemplo inicial:



# Árbol de Decisión

- Idea: escoger atributo "**más significativo**" como raíz del (sub)-árbol

## ¿Cómo?

- Si hay + y - ejemplos escoger atributo que mejor los divida (mayor discriminante)
- Si hay particiones con + y -, buscar un 2do atributo para seguir partiendo

## Macroalgoritmo AD(ejemplos, atributos)

Si ejemplos no vacíos entonces

Si ejemplos clasificados entonces

regresar (clasificación)

de lo contrario

mejor: `escoger_atributo(atributos, ejemplos)`

arbol: un nuevo árbol de decisión con *mejor* como raíz

por cada valor  $V_i$  de mejor

Subejemplos: ejemplos con  $mejor = V_i$

Subarbol: `AD(Subejemplos, atributos)`

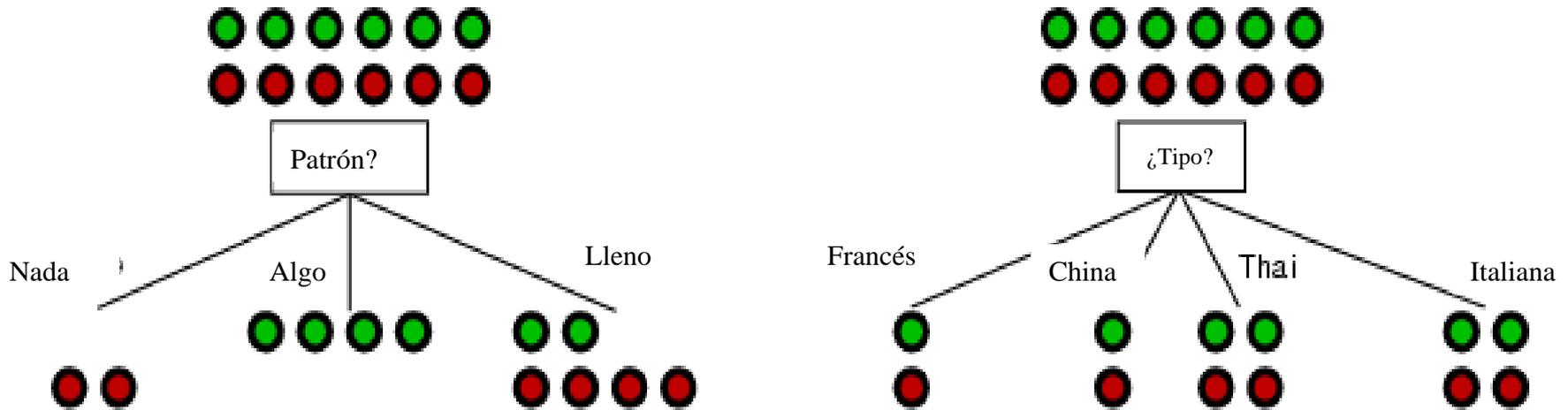
Arbol: `actualizar(nueva rama con etiqueta  $V_i$  y Subarbol)`

Regresa(árbol)

# Escoger un atributo

aprender reglas (clases)

¿*Patrón* es una mejor escogencia que *Tipo*?



Basado en conceptos vinculados a *contenidos de información*, p.ej.:

$$Info(p, n) = -p \log_2(p) - n \log_2(n)$$

Es una medida de la entropía (grado de desorden) de los ejemplos

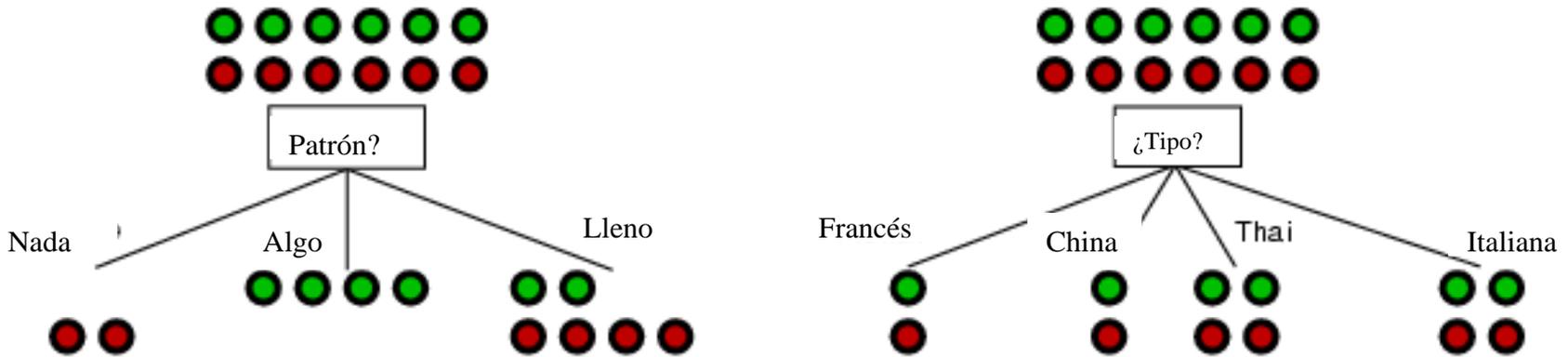
n: numero de ejemplos -

p: numero de ejemplos +

# Escoger un atributo

aprender reglas (clases)

¿*Patrón* es una mejor escogencia que *Tipo*?



Escoger atributo  $A$  con mas grande  $IG$  (ganancia en información)

$$IG(A) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - resto(A)$$

Donde:

$I$  es entropía de los ejemplos: 
$$I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$

y 
$$resto(A) = \sum_{i=1}^v \left| \frac{p_i - n_i}{p+n} \right| I\left(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\right)$$

$v$ : posibles valores de  $A$

$p_i$  y  $n_i$ ? ver siguiente lamina

# Escoger un atributo

## aprender reglas (clases)

¿Quién es  $p_i$ ?  $p_i$  puede ser

$$p_i = \frac{|E_i^+|}{|E_i^+| + |E_i^-|}$$

Donde  $E_i^+$  es el porcentaje de ejemplos clasificados como + por el valor  $v_i$  del atributo A

### Una Formula general para escoger a los atributos:

Como hay que elegir el atributo con mayor información (menor entropía), otra posibilidad es calcular una **función de merito (FM)**

$$FM(A) = \sum_{i=1}^v r_i \inf o(p_i, n_i)$$

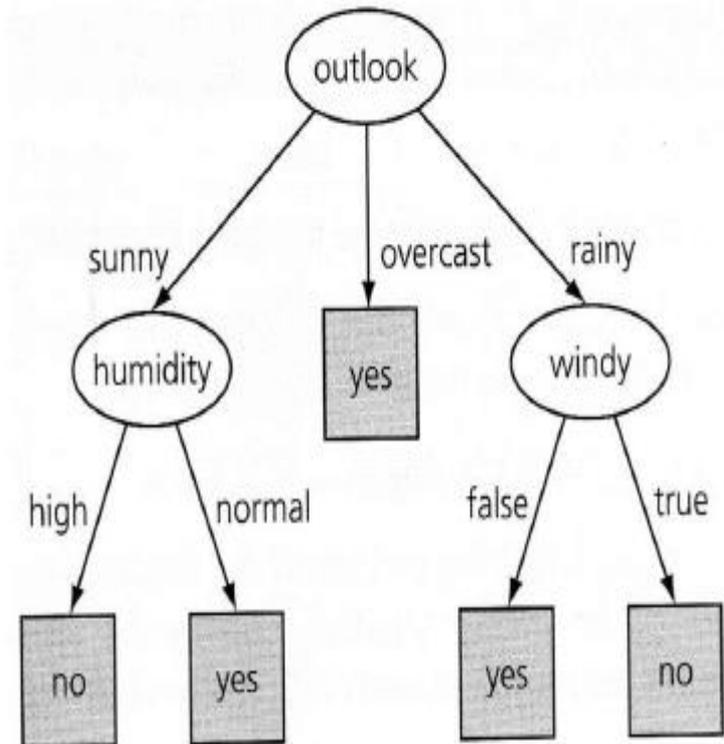
$p_i$  = % ejemplos clasificados como + en la rama i

$$r_i = \left| \frac{p_i - n_i}{p + n} \right|$$

# Construcción de árboles de decisión

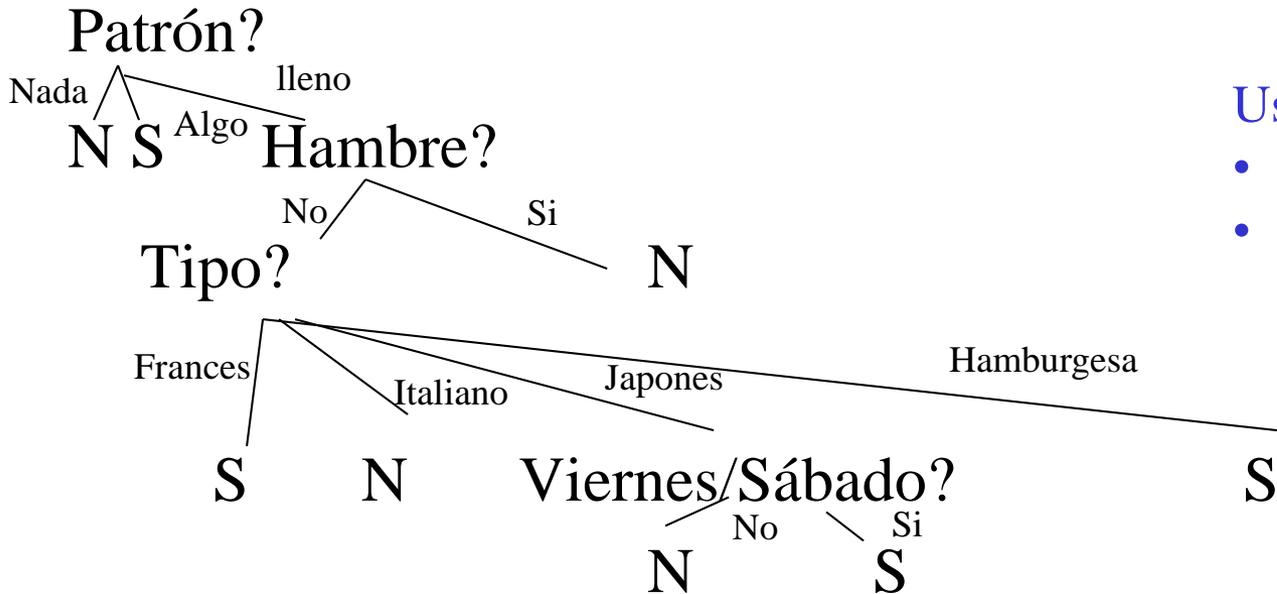
Se completa el árbol completando cada rama hasta cumplir un ciertos compromisos:

- **Número mínimo de hojas.**
- **Cobertura:** Mínimo número (o porcentaje) de casos posibles cubiertos correctamente de la BD.
- **Precisión:** Error de clasificación menor de un umbral puesto. Por ejemplo: precisión del 80%. Significa, que pararemos en esa hoja cuando el número de clases clasificadas correctamente sea mayor o igual al 80%.



# Arbol de Decisión y Lógica de Predicado

$\forall r \text{ espera}(r) \Rightarrow \text{Patrón}(r, \text{ algo}) \text{ O } (\text{Patrón}(r, \text{ full}) \text{ Y } \text{NoHambre}(r) \text{ Y } \text{tipo}(r, \text{ francés})) \text{ O } (\text{Patrón}(r, \text{ full}) \text{ Y } \text{NoHambre}(r) \text{ Y } \text{tipo}(r, \text{ hamburguesa})) \text{ O } (\text{Patrón}(r, \text{ full}) \text{ Y } \text{NoHambre}(r) \text{ Y } \text{tipo}(r, \text{ Japones}) \text{ Y } \text{viernes/Sabado}(r) )$



Uso de operadores:

- Para unir ramas O
- Para seguir una rama Y

# Deducción de reglas rudimentarias: Reglas de clasificación

Relation: weather.symbolic					
No.	1: outlook Nominal	2: temperature Nominal	3: humidity Nominal	4: windy Nominal	5: play Nominal
3	overcast	hot	high	FALSE	yes
7	overcast	cool	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
10	rainy	mild	normal	FALSE	yes
14	rainy	mild	high	TRUE	no
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes

## Evaluando los atributos de los datos

	Attribute	Rules	Errors	Total errors
1	outlook	sunny → no overcast → yes	2/5 0/4	4/14
2	temperature	rainy → yes hot → no* mild → yes cool → yes	2/5 2/4 2/6 1/4	5/14
3	humidity	high → no normal → yes	3/7 1/7	4/14
4	windy	false → yes true → no*	2/8 3/6	5/14

Possible atributo

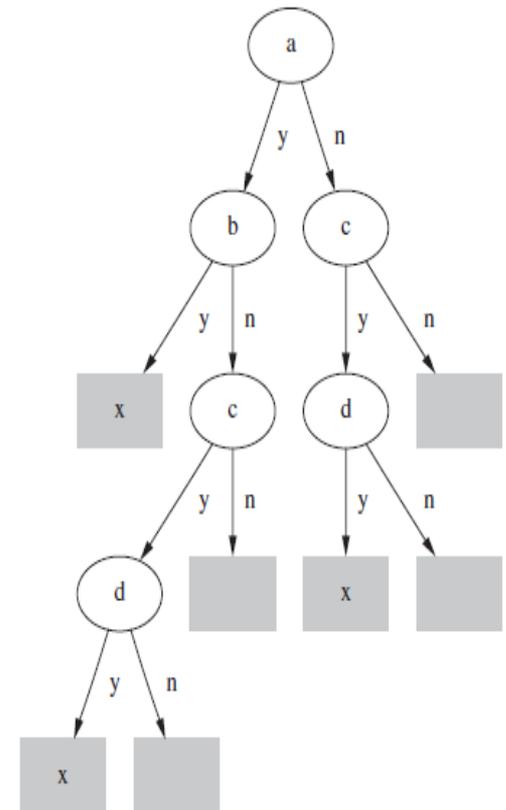
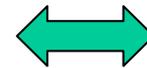
outlook: sunny → no  
overcast → yes  
rainy → yes

# Reglas de clasificación

Las reglas de clasificación son una alternativa popular a los árboles de decisión

Por ejemplo:

```
If outlook = sunny and humidity = high then play = no
If outlook = rainy and windy = true then play = no
If outlook = overcast then play = yes
If humidity = normal then play = yes
If none of the above then play = yes
```





# Modelización estadística

para un día dado

Outlook	Temperature	Humidity	Windy	Play
sunny	cool	high	true	?

Probabilidad de que sea **si**:  $2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.0053$

Probabilidad de que sea **no**:  $3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.0206$

Normalización

$$\text{Probability of yes} = \frac{0.0053}{0.0053 + 0.0206} = 20.5\%$$

$$\text{Probability of no} = \frac{0.0206}{0.0053 + 0.0206} = 79.5\%$$

# Reglas de Asociación

**Se utilizan para agrupar/asociar/descubrir hechos que ocurren en común dentro de un determinado conjunto de datos**

Por ejemplo, en la tabla anterior con la regla:

If temperature = cool then humidity = normal

Otra regla es:

If windy = false and play = no then outlook = sunny  
and humidity = high

# Reglas de Asociación

Similares a las reglas de clasificación

Pueden predecir cualquier atributo, no solo la clase, o predecir combinaciones de atributos.

- La *cobertura* de una regla de asociación es el número de instancias para las cuales ella predice correctamente (*soporte*).
- La *precisión* (*confianza*) es el número de instancias que predice correctamente, expresado como una proporción de todas las instancias a las que se aplica.

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

# Reglas de Asociación

## Items con cobertura mayor o igual a 2

	One-item sets	Two-item sets	Three-item sets	Four-item sets		One-item sets	Two-item sets	Three-item sets	Four-item sets
1	outlook = sunny (5)	outlook = sunny temperature = mild (2)	outlook = sunny temperature = hot humidity = high (2)	outlook = sunny temperature = hot humidity = high play = no (2)	...	...	humidity = normal windy = false (4)	humidity = normal windy = false play = yes (4)	
2	outlook = overcast (4)	outlook = sunny temperature = hot (2)	outlook = sunny temperature = hot play = no (2)	outlook = sunny humidity = high windy = false play = no (2)	38	39	humidity = normal play = yes (6)	humidity = high windy = false play = no (2)	
3	outlook = rainy (5)	outlook = sunny humidity = normal (2)	outlook = sunny humidity = normal play = yes (2)	outlook = overcast temperature = hot windy = false play = yes (2)	40	47	humidity = high windy = true (3)	...	windy = false play = no (2)
4	temperature = cool (4)	outlook = sunny humidity = high (3)	outlook = sunny humidity = high windy = false (2)	outlook = rainy temperature = mild windy = false play = yes (2)					
5	temperature = mild (6)	outlook = sunny windy = true (2)	outlook = sunny humidity = high play = no (3)	outlook = rainy humidity = normal windy = false play = yes (2)					
	...	...	...	...					

# Reglas de Asociación

- **Las reglas se obtienen a partir de valores de las variables**

humidity = normal, windy = false, play = yes

- **Esto nos lleva a las 7 reglas potenciales:**

**If humidity = normal and windy = false → play = yes 4/4**

If humidity = normal and play = yes → windy = false 4/6

If windy = false and play = yes → humidity = normal 4/7

If humidity = normal → windy = false and play = yes 4/6

If windy = false → humidity = normal and play = yes 4/8

If play = yes → humidity = normal and windy = false 4/9

If → humidity=normal and windy=false and play=yes 4/12

# Reglas de Asociación

## Reglas que implican relaciones

Sombreado: parado (standing)

No sombreado: acostado (lying)

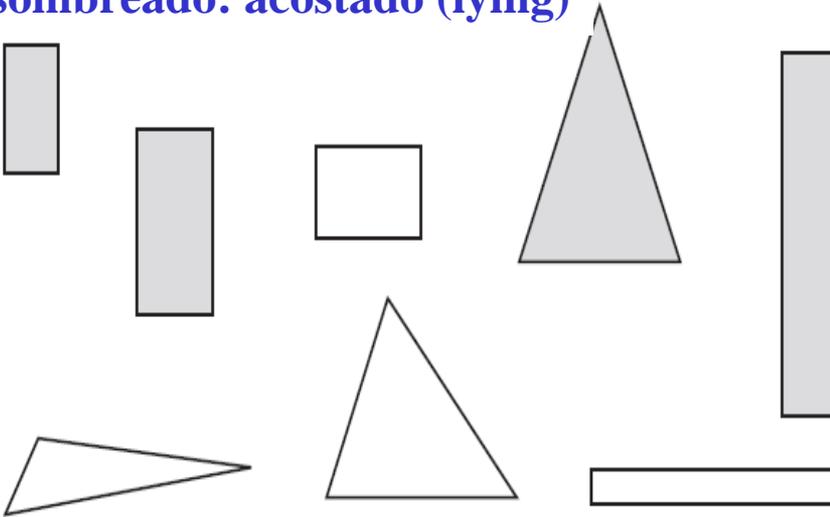


Tabla con datos de entrenamiento

Width	Height	Sides	Class
2	4	4	standing
3	6	4	standing
4	3	4	lying
7	8	3	standing
7	6	3	lying
2	9	4	standing
9	1	4	lying
10	2	3	lying

Reglas

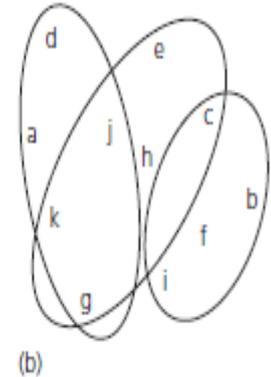
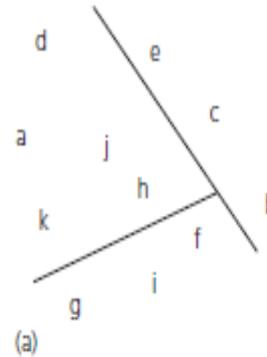


if width  $\geq 3.5$  and height  $< 7.0$  then lying  
if height  $\geq 3.5$  then standing

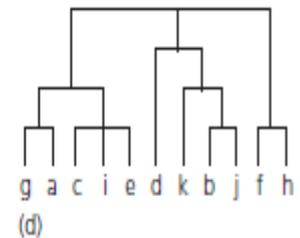
# Clustering

Técnica muy usada en la cual se infiere un árbol de decisión o conjunto de reglas que asigna a cada instancia al grupo al que pertenece

- *Algoritmos aglomerativos* en principio suponen que cada elemento es una clase y luego se van uniendo hasta conseguir el número deseado.
- *Algoritmos divisores* consideran que todos los elementos pertenecen a la misma clase y luego los van dividiendo en subclases.

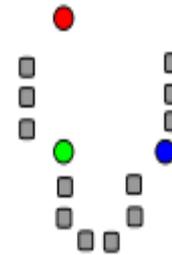


	1	2	3
a	0.4	0.1	0.5
b	0.1	0.8	0.1
c	0.3	0.3	0.4
d	0.1	0.1	0.8
e	0.4	0.2	0.4
f	0.1	0.4	0.5
g	0.7	0.2	0.1
h	0.5	0.4	0.1

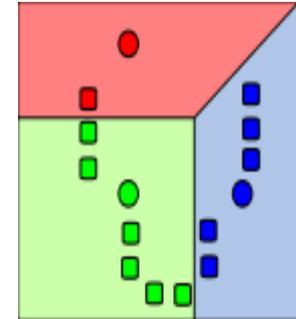


# Clustering

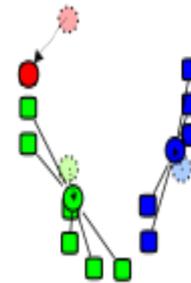
Algoritmo K-medias (K-means): es un método de agrupamiento, que tiene como objetivo la partición de un conjunto (n) en k grupos en el que cada observación pertenece al grupo más cercano a la media.



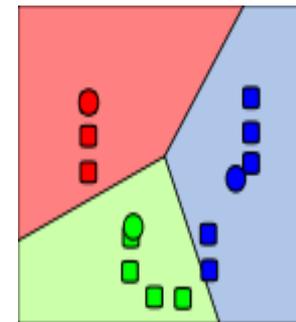
1)  $k$  centroides iniciales generados aleatoriamente (en este caso  $k=3$ )



2)  $k$  grupos son generados asociándole el punto



3) El centroide de cada uno de los  $k$  grupos se recalcula



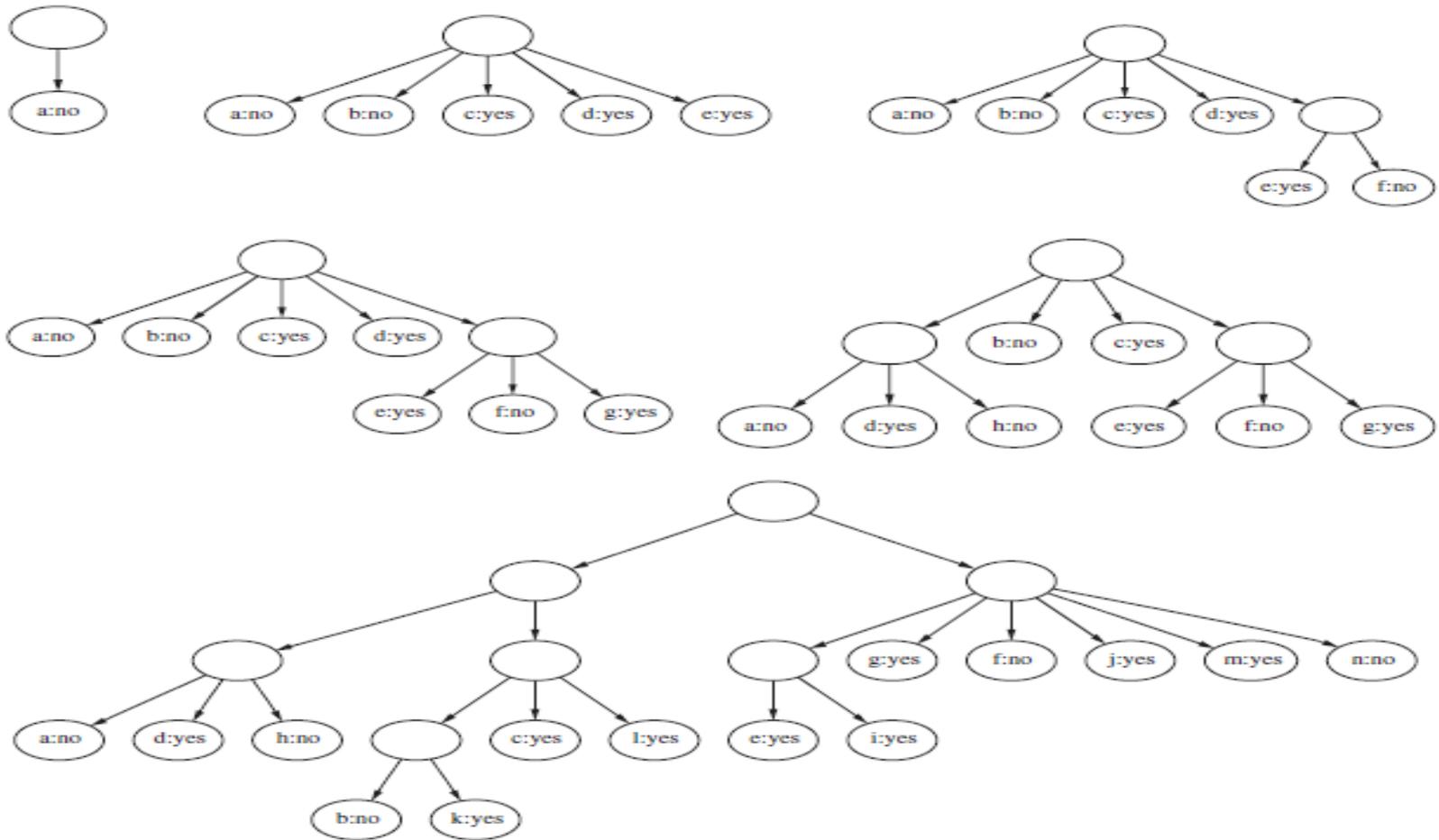
4) Pasos 2 y 3 se repiten hasta que se logre la convergencia.

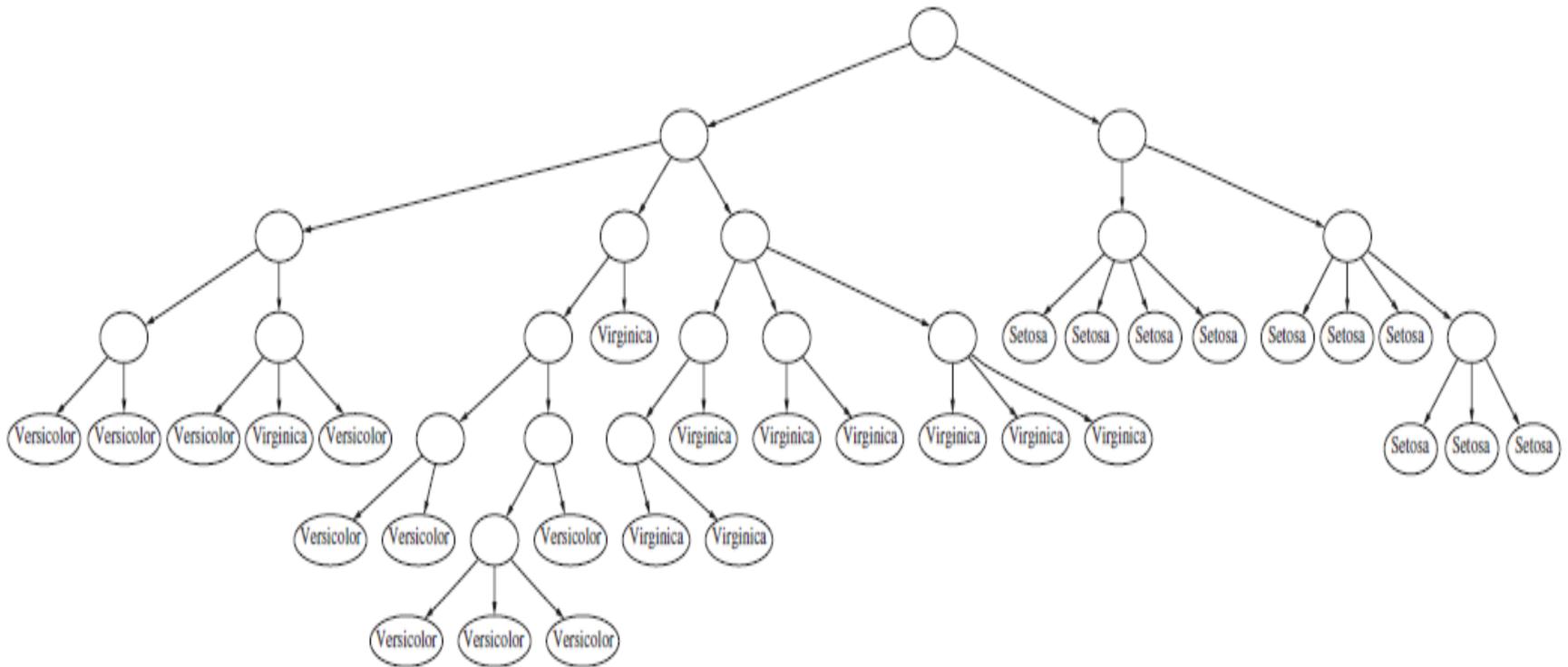
# Clustering Incremental

## Datos de tiempo

ID code	Outlook	Temperature	Humidity	Windy	Play
a	sunny	hot	high	false	no
b	sunny	hot	high	true	no
c	overcast	hot	high	false	yes
d	rainy	mild	high	false	yes
e	rainy	cool	normal	false	yes
f	rainy	cool	normal	true	no
g	overcast	cool	normal	true	yes
h	sunny	mild	high	false	no
i	sunny	cool	normal	false	yes
j	rainy	mild	normal	false	yes
k	sunny	mild	normal	true	yes
l	overcast	mild	high	true	yes
m	overcast	hot	normal	false	yes
n	rainy	mild	high	true	no

# Clustering Incremental





1. *Agrupación conceptual*: basado en la distancia entre los conceptos. Por ejemplo: círculo y triángulo.
2. *Cluster*: a partir de una serie de semillas van creciendo (en forma de estrella)
3. *Teoría de Witt*: Se basa en teorías de la información.
4. *Autoclass*: Teorías de Bayes.

# Utilidad de categoría

Mide la calidad general de una partición de casos en grupos

$$CU(C_1, C_2, \dots, C_k) = \frac{\sum_{\ell} \Pr[C_{\ell}] \sum_i \sum_j (\Pr[a_i = v_{ij} | C_{\ell}]^2 - \Pr[a_i = v_{ij}]^2)}{k}$$

$\Pr[a_i = v_{ij} | C_{\ell}]$  es una mejor estimación de la probabilidad de que el atributo  $a_i$  tiene un valor  $v_{ij}$ , para una instancia en el grupo  $C_{\ell}$

donde  $C_1, C_2, \dots, C_k$  son los  $k$  grupos; la suma exterior es más de estos grupos; las siguientes sumas interiores de los atributos;  $a_i$  es el atributo  $i$ -ésimo, y que se necesita en los valores  $v_{i1}, v_{i2}, \dots$  que son tratados por la suma sobre  $j$ .

# APRENDIZAJE DEDUCTIVO (APRENDIZAJE BASADO EN EXPLICACIONES)

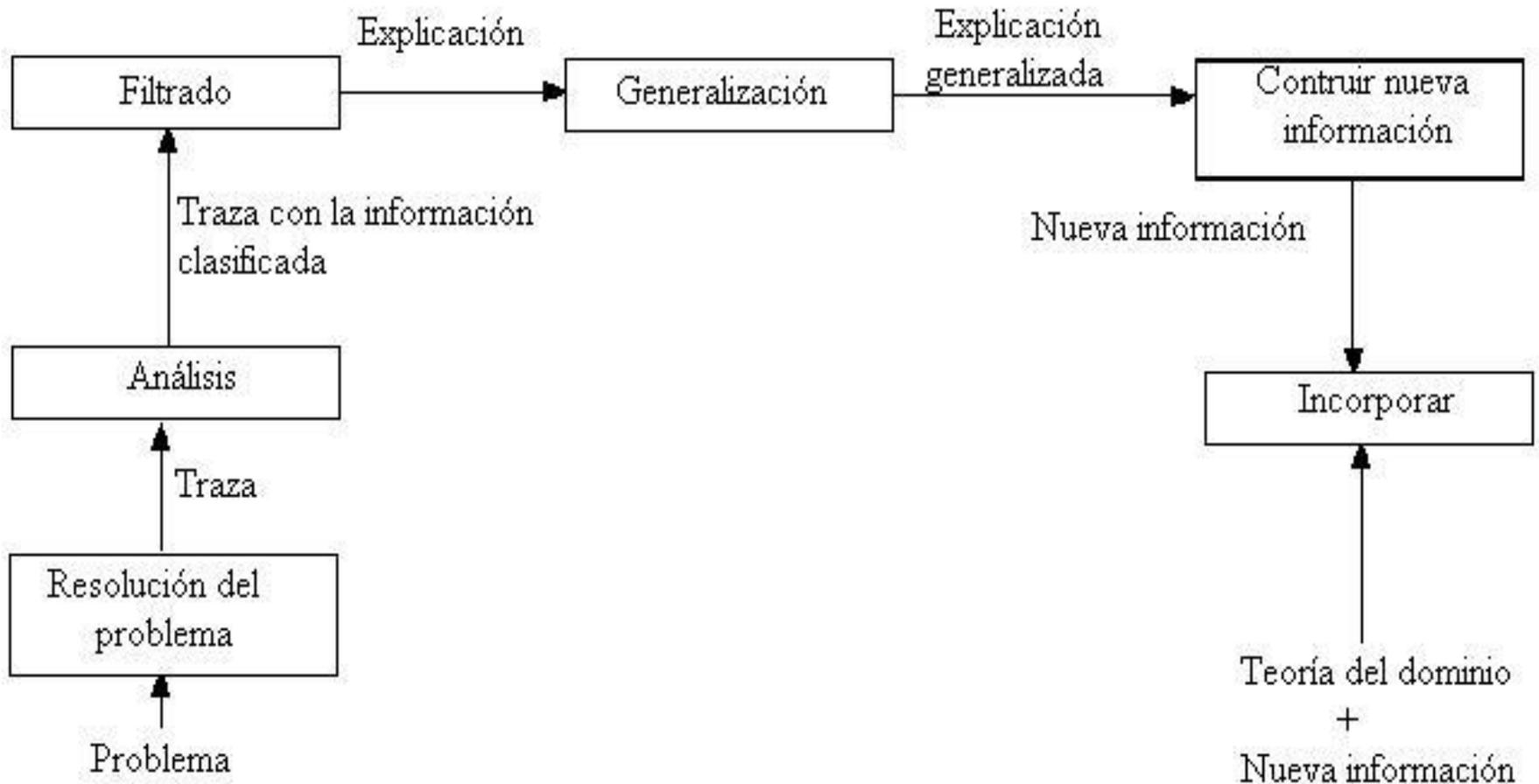
- El aprendizaje inductivo necesita muchos ejemplos para obtener información del dominio.

En el deductivo nos vale con un ejemplo pero necesitamos más conocimiento del dominio.

La deducción permite preservar la veracidad ya que hay conocimiento del dominio que respalda las afirmaciones.

el método deductivo no incorpora nuevo conocimiento, tan solo consigue explicar mejor (concretar) el conocimiento existente

# Componentes APRENDIZAJE DEDUCTIVO



# Componentes APRENDIZAJE DEDUCTIVO

1. ***Resolución del problema:*** utilizar el dominio y el ejemplo para llegar al objetivo.
2. ***Análisis de la traza:*** para conseguir obtener una explicación:
  - A. ***Criterio de relevancia.*** importante para abordar problemas futuros.
  - B. ***Criterio de operatividad:*** reglas que se activan directamente.
3. ***Filtrado:*** Coge lo que es relevante y operativo para obtener una explicación.

# Componentes APRENDIZAJE DEDUCTIVO

4. **Generalización:** Regresionar una fórmula  $f$  a través de una regla  $r$  consiste en determinar las condiciones necesarias y suficientes bajo las cuales puede usarse la regla  $r$  para obtener  $f$ .

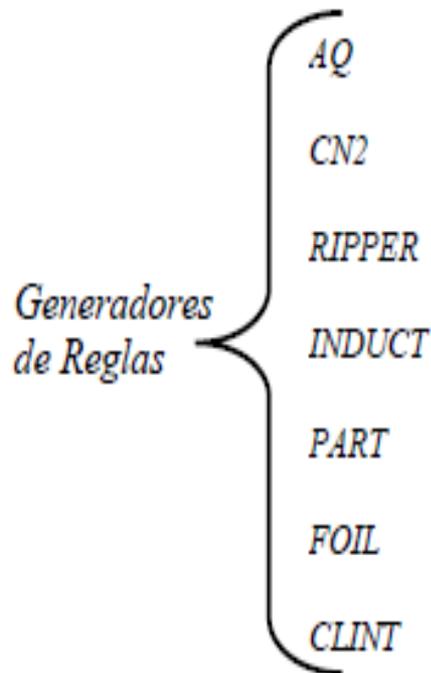
5. **Construir nueva información:** La información que construimos puede ser:

Reglas de dominio que expresaran nuevas definiciones de conceptos.

Reglas de control que se construyen de forma similar.

6. **Incorporar:** esas nuevas reglas hay que incorporarlas a la base de conocimiento.

# Paquetes de Generación de Reglas



- Algunas reglas inducidas pueden derivar de la construcción de un árbol de decisión, siendo primero generado el árbol de decisión y después trasladado a un conjunto de reglas
- Otros algoritmos se basan en el uso de técnicas de aprendizaje con lógica de predicados (ILP, Inductive Logic Programming). (FOIL, FFOIL, CLINT, etc.)

# Algoritmo de Aprendizaje

1. **Coleccionar** una gran cantidad de datos
2. Dividirlo en dos conjuntos disjuntos: de **entrenamiento** y de **prueba**
3. Usar algoritmo de aprendizaje con el conjunto de entrenamiento y **generar hipótesis H**
4. Ejecutar el conjunto de prueba H y **corregirlo**
5. Repetir el paso anterior para diferentes **conjuntos de prueba**, hasta que todos los elementos para diferentes conjuntos de prueba sean correctamente clasificados

# Ejecución de las pruebas

- **Clasifica bien los ejemplos:**

**Ejemplo X1:** Alternativa(X1) Y NoBar(X1) Y NoViern/Sap(X1) Y hambre(X1)...

**Clasificación:** Espera(X1)

- **Falsos resultados como:**

– **Falso Negativo:** Hipótesis dice – y en realidad es +

**Ejemplo:** Espera cuando Restaurante con patrón= full, con un tiempo de espera de 0-10 minutos y cuando agente no tiene hambre

**Pero hipótesis dice:**

$\forall r \text{ Patrón}(r, \text{full}) \text{ Y tiempoespera}(r, 0-10) \text{ Y Hambre}(r, N)$   
 $\Rightarrow \text{NoEspera}(r)$

# Pruebas de las Hipótesis

En el caso de las pruebas habrán:

- Hipótesis consistentes con los ejemplos
- Hipótesis con problemas con los ejemplos
  - Falso positivo: Hipótesis dice + y en realidad es -
    - Escoger **especialización** de H consistente con ejemplos
  - Falso negativo: Hipótesis dice - y en realidad es +
    - Escoger **generalización** de H consistente con ejemplos
  - Si no se consigue consistencia con especialización/generalización entonces falla

# Pruebas de las Hipótesis

- X1 es positivo Alt(X1) es verdad
  - H1:  $\forall x \text{ Espera}(x) \Leftrightarrow \text{Alt}(x)$
- X2 es negativo H1 predice positivo  $\Rightarrow$  **especial. H1**
  - H2:  $\forall x \text{ Espera}(x) \Leftrightarrow \text{Alt}(x) \text{ Y Patron}(x, \text{algo})$
- X3 es positivo H2 predice negativo  $\Rightarrow$  **generaliz. H2**
  - H3:  $\forall x \text{ Esper}(x) \Leftrightarrow \text{Alt}(x) \text{ Y patron}(x, \text{algo})$

## Algoritmo

V : conjunto de hipótesis

Para cada ejemplo (e)

Si V no esta vacía entonces

$V = \{h \in V : h \text{ es consistente con } e\}$

o  $V = \{h \in V : h \text{ especializada (agrega cond.) o generalizada (quita cond)}\}$

regresar V

# Podado de un Árbol

¿Cómo decidir si desea reemplazar un nodo interno con una hoja?

Imaginemos que la verdadera probabilidad de error en el nodo es  $q$ , y que las  $N$  instancias son generados por un proceso de Bernoulli con parámetro  $q$ , de la que  $E$  son los errores. **El intervalo de confianza** viene dado por:

$$\Pr\left[\frac{f - q}{\sqrt{q(1-q)/N}} > z\right] = c,$$

donde  $N$  es el número de muestras,  $f = E / N$  es el porcentaje de error observado, y  $q$  es la tasa de error. Al igual que antes, esto conduce a un **límite superior de confianza** para  $q$ .

Ahora usamos ese límite superior de confianza como una estimación (pesimista) **para la tasa de error  $e$  en el nodo:**

$$e = \frac{f + \frac{z^2}{2N} + z \sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}}}{1 + \frac{z^2}{N}}.$$

# Podado de un Árbol

**None:**  $E = 2$ ,  $N = 6$ , y por lo que  $f = 0,33'$ .  $e = 0,47$ . tasa de error de formación es del 33%, se utilizará la estimación pesimista del 47%.

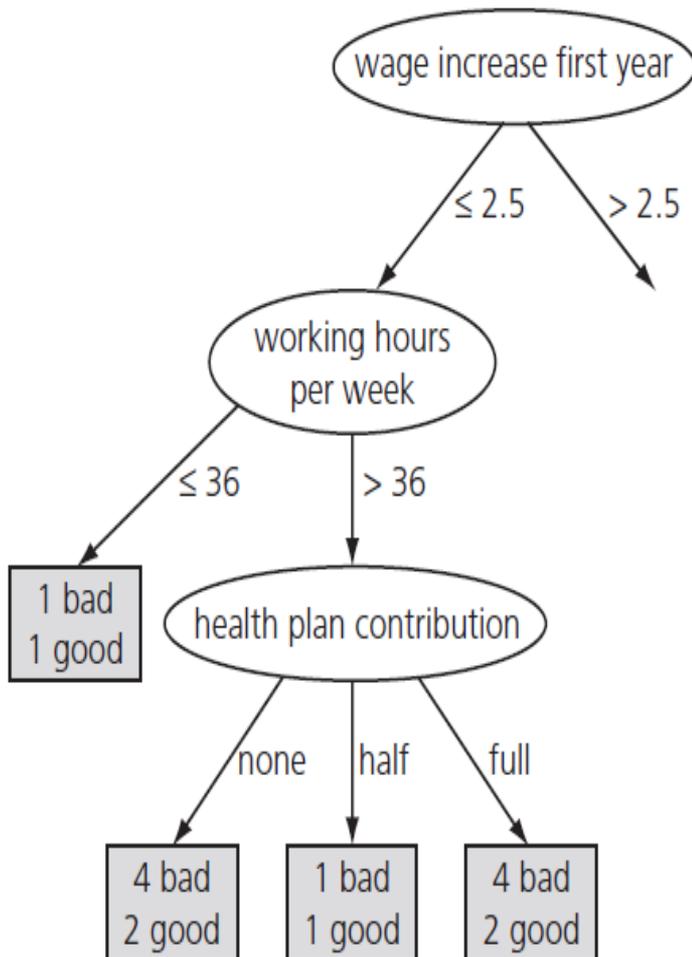
**Half:**  $E = 1$ ,  $N = 2$ ,  $e = 0.72$ .

**Full:** Tiene el mismo valor de  $e$  como el primero.

**El siguiente paso es combinar las estimaciones de error para estos tres hojas** en la relación entre el número de ejemplos que se refieren, 6: 2: 6, lo que conduce a una estimación de error combinado de 0,51.

**Health plan contribution:**  $f = 5/14$ .  $e = 0.46$ . Debido a que este es menor que el error de estimación combinada de los tres niños, **ellos no se podan**.

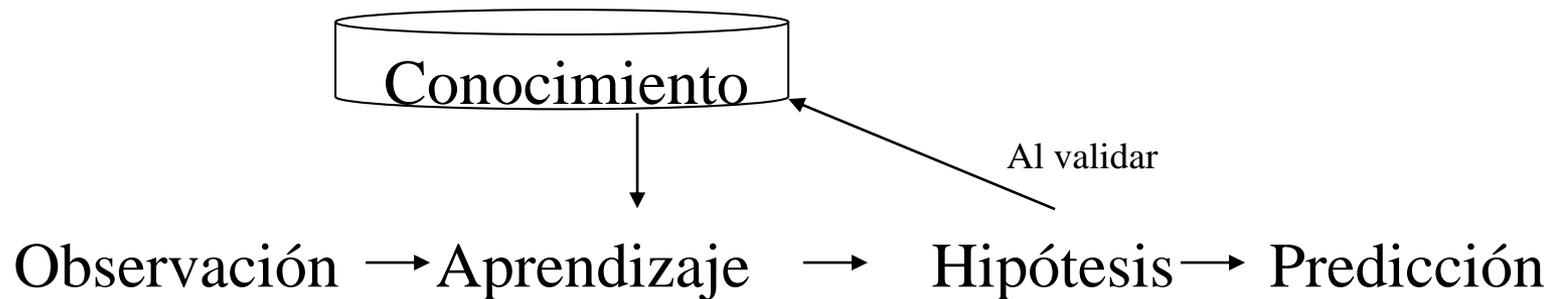
**Working hours per week:** La estimación de error para la primera, con  $E = 1$  y  $N = 2$ , es  $e = 0,72$ , y para el segundo es  $e = 0,46$ . La combinación de estos, 2 : 14, conduce a un valor que es mayor que la estimación del error para el nodo de horas de trabajo, por lo que **el subárbol se poda y se sustituye por un nodo hoja**.



# Aprendizaje y Conocimiento

- Conocimiento a Priori
- Algoritmos de Aprendizaje:

**Añadir conocimiento al inicial**



# Representación basada en instancias

**La forma más simple de aprendizaje es la memorización simple, o el aprendizaje memorístico.**

- Los casos iniciales se almacenan en memoria, de esta forma cuando llega un nuevo caso con valor desconocido se intenta relacionar éste con el caso almacenado que más se parezca.
- Si no se parece a ninguno, la almacena como una nueva clase.
- Se denominan técnicas perezosas (lazy) o retardadas, pues el aprendizaje se hace de forma progresiva,
- No se realiza un trabajo previo de generalización con los datos de entrenamiento (técnicas anticipativas o voraces).



# Aprendizaje y Conocimiento

- **Aprendizaje basado en Información Relevante:**
  - Usa conocimiento a priori para identificar atributos relevantes.
  - Deduce generalizaciones desde ejemplos simples

## Conocimiento a priori:

$\forall x,y,n,l$  Nacionalidad(x,n) Y Nacionalidad(y,n) y Lengua(x,l)  $\Rightarrow$  Lengua(y,l)

## Observación

Nacionalidad(Jose,Vzla) y Lengua(Jose,Español)

## Conclusión

$\forall x$ , Nacionalidad(x,Vzla)  $\Rightarrow$  Lengua(x,Español)

**Nacionalidad determina la lengua!!**

# Aprendizaje y Conocimiento

- **Aprendizaje basado en casos:** Similitud a procesos judiciales y médicos
  1. Extraer de la BC caso más similar al que se plantea nuevamente
  2. Cabe la posibilidad de aprender nuevos casos o añadir elementos a los existentes
- BC:

	Caso 1 resfriado	caso 2: gripe
Edad	20	60
Fiebre	38	39
Fuma	N	S
Congestión nasal	S	N
Dolor de garganta	S	N

**Ante la presencia de un nuevo caso, se trata de seleccionar el más parecido, o añadir un nuevo síntoma no recogido en la BC**

# Aprendizaje y Conocimiento

- **Aprendizaje por analogía: Emular capacidad humana de recordar soluciones de problemas previos cuando aparecen problemas parecidos**
- BC tiene dichas soluciones
- Ejemplo:
  - Se estudia IA  $c/2$ días, SO  $c/\text{día}$  BD  $c/3$ días
  - Cuando se tardará en estudiar IA+SO en un mismo día después de haber estudiado las 3 juntas?
  - Procedimiento:
    - BC= mcm de un grupo de números
    - $M_0 = \text{mcm}(\text{IA}, \text{SO})$  2
    - $M_1 = \text{mcm}(\text{IA}, \text{SO}, \text{BD})$  6
    - $M = M_0 + M_1$  8
- Nuevo problema: tres aviones salen de un aeropuerto cada 3, 5 y 9 días respectivamente. Cuanto tiempo tardaran en salir juntos el 1er y 2do avión después de haber coincido saliendo los 3 juntos?

# Modelo unificado de analogía

KEDAR-COBELL (1988)

1. **Recuperación:** Partiendo del problema objetivo (dominio objetivo) buscamos un problema base en el dominio base que tenga similitudes significativas.
2. **Elaboración:** Define conocimiento del problema base para poder aplicarlo en el problema objetivo.
3. **Mapeo:** Trasladamos el conocimiento desde el problema base hasta el objetivo.
4. **Justificación:** Comprobamos si lo que sabemos es aplicable y funciona en la función objetivo.
5. **Aprendizaje:** Consiste en almacenar el conocimiento obtenido aprendido, en el conocimiento objetivo

**Problema base:** situación conocida de ante mano.

**Dominio base:** lo que se conoce de él.

**Problema objetivo:** aplicamos lo conocido a la nueva situación.

**Dominio objetivo:** conocimiento obtenido de aplicar el problema objetivo junto con la situación conocida.