

## □ Tema 5: Aprendizaje automático

- Introducción
- Tipos de aprendizaje
  - Según el grado de realimentación
  - Según el paradigma utilizado
  - Según lo que se aprende  
(o según el problema que se resuelve)

- Aspectos fundamentales de la Inteligencia Artificial
  - Representación del conocimiento
  - Métodos de búsqueda
  - Aprendizaje

**No se puede hablar de inteligencia sin aprendizaje**

# Aprendizaje Automático (*machine learning*)

- ❑ Aprendizaje
  - ❑ “El acto, proceso o experiencia de adquirir conocimiento o aptitudes”
  - ❑ ... y poder hacer algún tipo de inferencia o acción con ello
- ❑ El aprendizaje es la clave de la inteligencia
- ❑ El aprendizaje está relacionado con el conocimiento
  - ❑ “El proceso mediante el cual un ente adquiere conocimiento”
  - ❑ Este conocimiento puede ser suministrado
    - ❑ por otro ente denominado “profesor” o
    - ❑ o adquirirse por el propio ente “automáticamente”
- ❑ El aprendizaje automático es un área muy activa porque...
  - ❑ Hay gran cantidad de información en formato electrónico
  - ❑ La mejora constante de las técnicas y los algoritmos
  - ❑ La potencia de procesamiento cada vez es más barata

## Aprendizaje de... conocimiento

- ❑ Sistema Inteligente
  - ❑ ¿qué conocimiento? ¿cómo adquirirlo?
- ❑ Ingeniería de conocimiento – expertos humanos
  - ❑ El cuello de botella de la adquisición de conocimiento
  - ❑ Mucho tiempo y mala gestión de errores
  - ❑ ¡¡¡ A veces ni siquiera está disponible !!!
  - ❑ Ejemplo: todos somos expertos en reconocimiento de imágenes
    - ❑ Podemos ver una foto y reconocer gente envejecida, con poca luz, bronceada, girada,....
  - ❑ ¿Podemos escribir un programa que lo haga? ¿Y ayudar a explicitar el conocimiento que utilizamos para realizar este proceso?
- ❑ Hay muchos casos en los que las personas aprenden de ejemplos

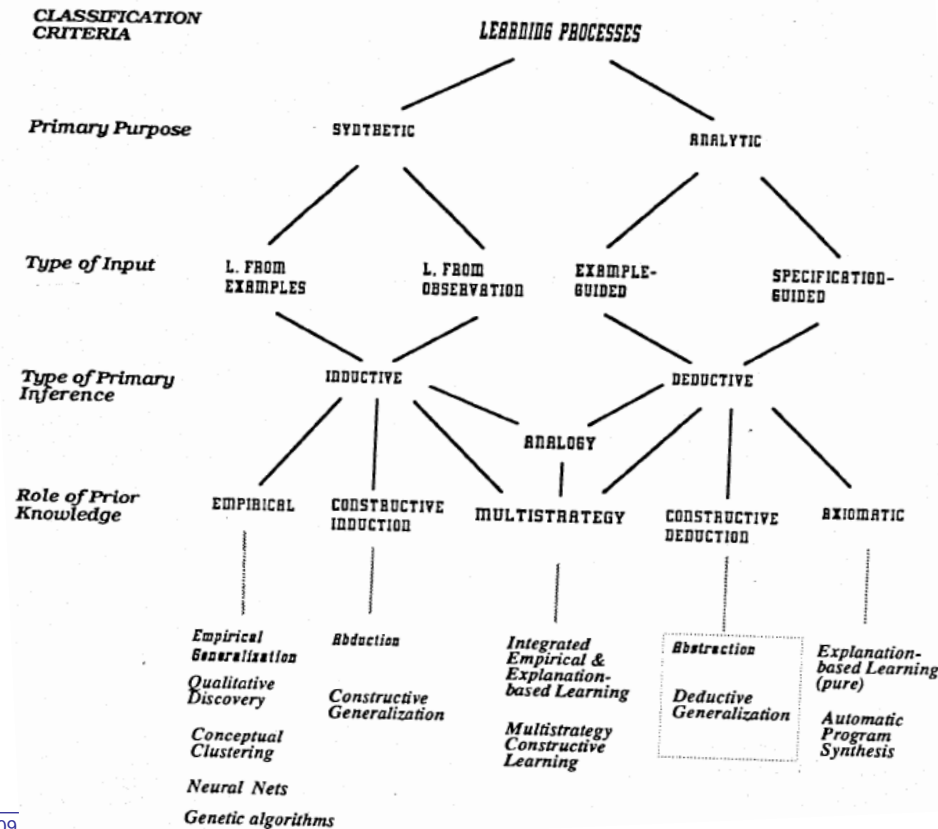
## ¿Qué entendemos por aprendizaje?

- ☐ “Cambios en el sistema que son adaptativos: permiten llevar a cabo la misma tarea de un modo más eficiente y eficaz”  
(Herb Simon, 1983)
  - ☐ El comportamiento en el desarrollo de una determinada tarea mejora con la **experiencia**
    - ☐ Refinamiento de habilidades
    - ☐ Adquisición de conocimiento
- ☐ El aprendizaje es, por sí mismo, un proceso de **resolución de problemas**
- ☐ La idea de poder aprender de la propia experiencia en la resolución de problemas nos lleva a esperar obtener mejores soluciones en un futuro
  - ☐ En dominios grandes, puede representar la diferencia entre la resolución rápida de un problema y la imposibilidad de resolverlo

## ¿Qué entendemos por aprendizaje?

- ☐ *“La capacidad de aprendizaje permite realizar nuevas tareas que previamente no podían realizarse, o bien realizar mejor (más rápidamente, con mayor exactitud, ...) las que ya se realizaban, como resultado de los cambios producidos en el sistema al resolver problemas anteriores”*
- ☐ *“La captura y/o transformación de conocimiento en un formato utilizable para mejorar el rendimiento”*
- ☐ El aprendizaje no puede añadirse a posteriori a un programa
- ☐ Aprender significa varias cosas. Entre otras:
  - ☐ Memorizar
  - ☐ Observar y explorar
  - ☐ Practicar para desarrollar las capacidades
  - ☐ Organizar el nuevo conocimiento de una forma general y efectiva

# Clasificaciones de Aprendizaje



# Clasificaciones de Aprendizaje

- ❑ Según el grado de realimentación
  - ❑ Cómo se observa, explora y memoriza
  - ❑ Supervisado, no supervisado (por descubrimiento) y por refuerzo
- ❑ Según el paradigma utilizado
  - ❑ Aprendizaje inductivo: aprende descripción (con muchos ejemplos “+” y “-”)
  - ❑ Aprendizaje analítico/deductivo: mejor resolver problemas (1ejemplo+teoría)
  - ❑ Algoritmos genéticos: crea generación de soluciones y evalúa su calidad
  - ❑ Método conexionista-red neuronal: reconocimiento de patrones (letras, voz)
- ❑ Según lo que se aprende (usan los paradigmas anteriores)
  - ❑ Aprendizaje de resolución de problemas
    - ❑ Memorístico o rutinario ; Por ajuste de parámetros ;
    - ❑ Macro-operadores ; Macro-reglas y meta-reglas ;
    - ❑ Por analogía ; Razonamiento basado en casos (CBR)
  - ❑ Aprendizaje de conceptos
    - ❑ Inductivo (Winston,Mitchell, Quinlan ID3). Analítico (explicaciones EBL)

## Tipos de aprendizaje: según el grado de realimentación

### ☐ Aprendizaje supervisado

- ☐ Los ejemplos de entrenamiento son comprobables...Ya sea porque
  - ☐ "el profesor" le suministre el valor real al sistema
  - ☐ o porque pueda percibirlo (en entornos observables)
- ☐ Redes neuronales supervisadas y los demás que veremos (no clustering)

### ☐ Aprendizaje no supervisado o por descubrimiento

- ☐ No hay "profesor" para verificar los ejemplos
- ☐ Objetivo: descubrir patrones en el conjunto de entrenamiento que...
- ☐ Permitan agrupar y diferenciar unas observaciones (ejemplos) de otras:
  - ☐ Formación de taxonomías y clustering conceptual
  - ☐ Descubrimiento de leyes cualitativas (AM) de leyes cuantitativas (BACON)

### ☐ Aprendizaje por refuerzo

- ☐ El sistema recibe algún tipo de recompensa (positiva o negativa)
- ☐ Cada vez que produce una respuesta,
- ☐ Ajustando su comportamiento en función de dicha recompensa

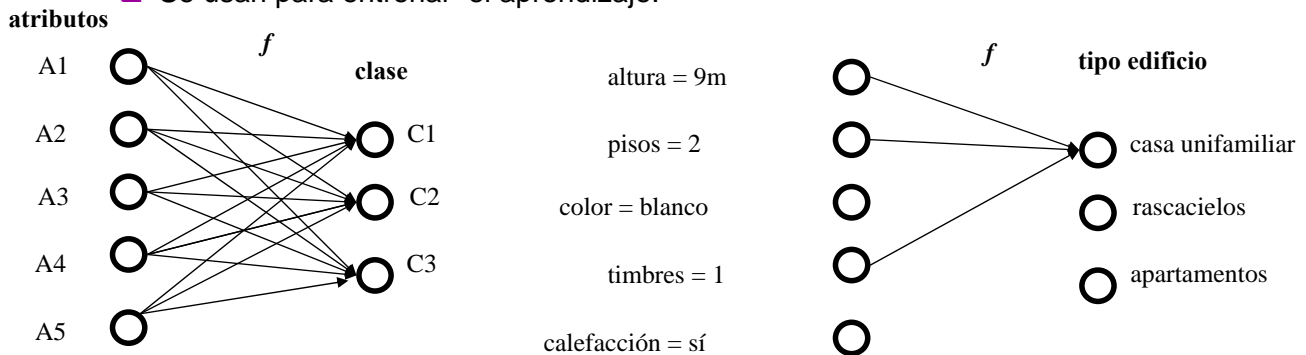
## ---Aprendizaje inductivo ---

### ☐ Clasificaciones del aprendizaje:

- ☐ Grado de realimentación: no supervisado
- ☐ Paradigma: inductivo
- ☐ Qué aprende: descripción de Conceptos (y también reglas)

# Aprendizaje inductivo: Definiciones previas

- ❑ Estrategia de aprendizaje: algoritmo que genera la descripción concepto/clase
- ❑ Ejemplos de entrenamiento
  - ❑ Describen situaciones reales mediante un conjunto de atributos
  - ❑ Se usan para entrenar el aprendizaje.



- ❑ Ejemplos positivos “+” (ejemplos que sí son de ese concepto o clase) y negativos “-” (que no son del concepto, contraejemplos)
- ❑ Clasificar los ejemplos: asignarle etiqueta de + o -
- ❑ Ruido: si se ha dado una etiqueta + a un ejemplo negativo o viceversa
- ❑ Hipotesis: conjeturas posibles sin probar (solución: la descripción adecuada).

# Aprendizaje inductivo: *aprendizaje a partir de ejemplos*

- ❑ Objetivo: construir la descripción de un concepto (conjunto condiciones)
  - ❑ Así establecer rasgos comunes de una serie de ejemplos
  - ❑ En la que encajen todos los ejemplos “+” y ningún ejemplo “-”
  - ❑ Que permita decidir si un ejemplo sin etiquetar es “+” ó “-”
  - ❑ Y así diferenciarlo de otro concepto : Clasifica en Categorías
- ❑ Estrategia: Proceso de búsqueda en el espacio de hipótesis
- ❑ Un Problema: Dada una lista de ejemplos (previamente clasificadas)
  - ❑ pares ordenados de números: {[ (3,4), E+ ], [ (4,5), E- ], [ (9,5), E- ], [ (1,6), E+ ]}
- ❑ Una solución (concepto obtenido):
  - ❑ "par ordenado de números en el que el primero es menor que el segundo" (una hipótesis de todas las posibles)

## Aprendizaje inductivo : Dificultades

- ☐ Disponemos de un número suficiente de ejemplos?
- ☐ Problemas de ruido (ejemplos mal clasificados)
- ☐ Ejemplos malos/sesgados/poco representativos
- ☐ Información irrelevante → dificulta la inducción

- ☐ Ejercicio:

- ☐ Definir un vehículo: bici, helicóptero, barco,...

## --- Sistema de aprendizaje de Winston ---

- ☐ Es inductivo supervisado
- ☐ Construir definiciones de conceptos del dominio “mundo de bloques”
  - ☐ Recibe **incrementalmente ejemplos** y contraejemplos de un concepto
  - ☐ Genera incrementalmente una **descripción** estructural de dicho **concepto**
  - ☐ Esta descripción sirve para clasificar correctamente ejemplos desconocidos
- ☐ Se emplea en tareas de reconocimiento:
  - ☐ de caracteres, de piezas defectuosas, etc.
- ☐ Es **sensible al orden** de los ejemplos
- ☐ Realiza un recorrido primero en profundidad
  - ☐ en el espacio de posibles descripciones de conceptos

## Aprendizaje de Winston: Análisis situación

- ❑ Se empieza por analizar la situación, determinando
  - ❑ El problema: crear una descripción textual basada en reglas  
Distinguir lo que es un “arco” de otra cosa
  - ❑ Elementos relevantes: objetos, relaciones o propiedades  
Objetos: bloques B, C, D  
Relaciones: vertical, horizontal, separados, es\_un, tiene\_parte
  - ❑ Determinar la representación formal de la descripción textual  
apoyado\_en(a1 a2) no\_unión(a2 a3)  
Si hecho1(a1 a4) & ... hecho8(a2) Entonces es\_un(a1, arco)
  - ❑ Construir ejemplos “+” y “-” (50% cada) en el lenguaje formal
    - ❑ con la ayuda de expertos del dominio (el “profesor”)
- ❑ Algoritmo: Encontrar una descripción que sea un patrón (regla)
  - ❑ Donde encajen los ejemplos adecuados (+) y no los contraejemplos (-)
  - ❑ Mediante un proceso de refinamiento (generaliza o especifica la regla)

## Aprendizaje de Winston: Algoritmo

Pasos:

1. Crear una descripción estructural (regla) usando el primer ejemplo +
2. Generalizar la regla para que incluya todos los ejemplos +
3. Especificar la regla para que excluya todos los ejemplos -

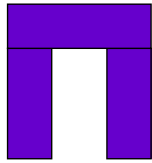
Operaciones con los ejemplos (para los dos últimos pasos)

- ❑ Variabilización: Añadir variables (generalizar)
- ❑ Jerarquizar las clases o conceptos (generalizar)
- ❑ Debilitar añadiendo disyunciones de condiciones (generalizar)
- ❑ Fortalecer añadiendo conjunciones de condiciones (especificar)
- ❑ Cuantificar para todos los ejemplos (generalizar). Así se termina.



## Aprendizaje de Winston : Ejemplo en texto

### Ejemplos recibidos

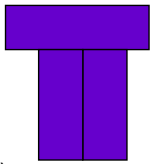


( + )

### Descripción estructural generada

**Dos bloques verticales  
y un bloque horizontal**

Se empieza analizando una instancia conocida del concepto que quiere describirse (*arco, en este caso*)



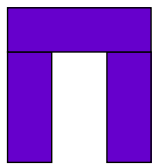
( - )

**Dos bloques verticales *separados*  
y un bloque horizontal**

Cuando un ejemplo (–) encaja es porque la descripción es demasiado general → *especializar*

## Aprendizaje de Winston : Ejemplo en texto

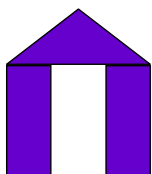
### Ejemplos recibidos



( + )

### Descripción estructural generada

**Dos bloques verticales separados  
y un bloque horizontal**



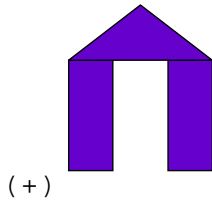
( + )

**Dos bloques verticales separados  
y un *objeto* horizontal**

Cuando un ejemplo (+) no encaja es porque la descripción es demasiado específica → *generalizar*

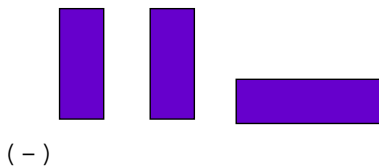
## Aprendizaje de Winston : Ejemplo en texto

### Ejemplos recibidos



### Descripción estructural generada

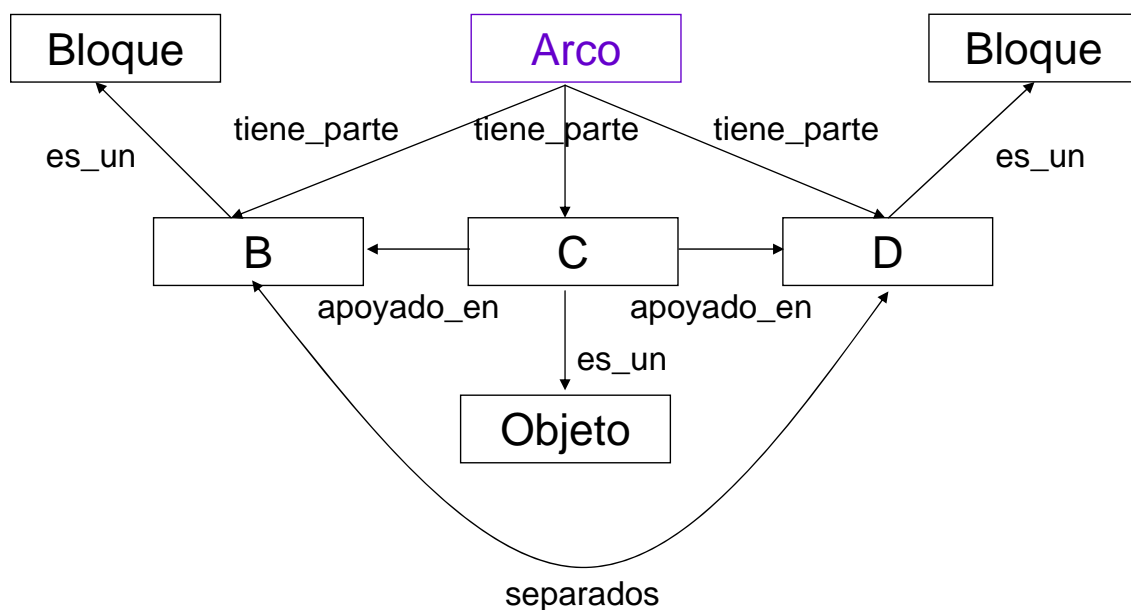
Dos bloques verticales separados  
y un objeto horizontal



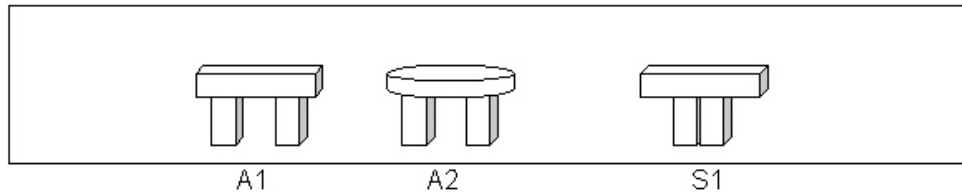
Dos bloques verticales separados  
y un objeto horizontal  
*apoyado en ellos*

## Aprendizaje de Winston : Ejemplo con redes

- Las descripciones se representan con redes semánticas
- Facilitan el proceso de generalización y particularización



## Aprendizaje de Winston : Ejemplo formalizado



- ❑ **Paso 1:** análisis de la instancia A1 (ejemplo positivo de arco)  
`tiene_parte(A1, B1) & tiene_parte(A1, C1) & tiene_parte(A1, D1) & apoyado_en(C1, B1) & apoyado_en(C1, D1) & es_un(B1, ladrillo) & es_un(C1, ladrillo) & es_un(D1, ladrillo) & no_union(B1, D1)`
- ❑ **Paso 2:** análisis de la instancia A2 (ejemplo positivo de arco)  
`tiene_parte(A2, B2) & tiene_parte(A2, C2) & tiene_parte(A2, D2) & apoyado_en(C2, B2) & apoyado_en(C2, D2) & es_un(B2, ladrillo) & es_un(C2, cilindro) & es_un(D2, ladrillo) & no_union(B2, D2)`
- ❑ Luego, la descripción se **generaliza** para incluir a ambas instancias  
`tiene_parte(A, B) & tiene_parte(A, C) & tiene_parte(A, D) & apoyado_en(C, B) & apoyado_en(C, D) & es_un(B, ladrillo) & es_un(C, objeto) & es_un(D, ladrillo) & no_union(B, D)`  
donde tanto ladrillo (bloque) como cilindro son instancias de una clase superior objeto

## Aprendizaje de Winston : Ejemplo formalizado

- ❑ **Paso 3:** se considera la instancia S1 (semejante, ejemplo negativo)  
`tiene_parte(S1, T1) & tiene_parte(S1, U1) & tiene_parte(S1, V1) & apoyado_en(U1, T1) & apoyado_en(U1, V1) & es_un(T1, ladrillo) & es_un(U1, ladrillo) & es_un(V1, ladrillo)`
- ❑ La definición del concepto debe **restringirse** para excluir el semejante
  - ❑ La definición debe excluir de manera explícita al semejante fortaleciendo la relación con “no\_debe\_union”
- ❑ La descripción queda de la siguiente manera:  
`tiene_parte(A, B) & tiene_parte(A, C) & tiene_parte(A, D) & apoyado_en(C, B) & apoyado_en(C, D) & es_un(B, ladrillo) & es_un(C, objeto) & es_un(D, ladrillo) & no_debe_union(B, D)`

## --- Espacio de versiones de Mitchell -----

- ☐ Es inductivo y supervisado
- ☐ Mismo objetivo:
  - ☐ producir una **descripción** de un **concepto** que sea consistente con los **ejemplos** positivos y negativos de entrenamiento
- ☐ No se ve afectado por el orden en que se presentan los ejemplos
- ☐ Se mantiene un **conjunto de descripciones posibles** hasta el final
  - ☐ Existen descripciones más generales que otras (**orden parcial**)
  - ☐ La jerarquía completa forma el **espacio de descripciones de conceptos**

## Espacio de versiones de Mitchell

- ☐ A medida que se procesan los ejemplos de entrenamiento se va refinando la noción de dónde se encuentra el concepto destino
  - ☐ **G**: conjunto de descripciones más **generales** que son consistentes con los ejemplos
  - ☐ **E**: conjunto de descripciones más **específicas** que son consistentes con los ejemplos
- ☐ La parte del espacio de descripciones de conceptos comprendida entre las fronteras G y E se denomina **espacio de versiones**

## Espacio de versiones de Mitchell: algoritmo

**G** := hipótesis vacía (*variables*)

**E** := primer ejemplo positivo

**mientras** queden ejemplos **y** no parar

- ☐ Coger siguiente ejemplo SE
- ☐ Si SE es positivo => eliminar de G cualquier descripción que no sea consistente con SE. **Generalizar E lo imprescindible para que sea consistente con SE**
- ☐ Si SE es negativo => eliminar de E descripciones consistentes con SE. **Especializar G lo imprescindible para que no sea consistente con SE**
- ☐ Si E y G son unitarios e iguales => imprimir contenido y parar
- ☐ Si E y G son unitarios e incompatibles => error de inconsistencia y parar

## Espacio de versiones de Mitchell: ejemplo 1

- ☐ Aprender el perfil característico de los pacientes con una determinada enfermedad
  - ☐ Cada paciente se describe con 5 características:
    - ☐ Edad (<20, 20-50, >50)
    - ☐ Lugar de residencia (ciudad, campo, montaña, costa)
    - ☐ Actividad física (baja, media, alta)
    - ☐ Consumo de tabaco (nulo, bajo, medio, alto)
    - ☐ Tipo de trabajo (sedentario, físico, paro)
  - ☐ Ejemplos de entrenamiento
    - 1. (20-50, ciudad, baja, nulo, sedentario)    +
    - 2. (>50, campo, baja, medio, físico)        -
    - 3. (>50, ciudad, baja, alto, sedentario)    +
    - 4. (20-50, ciudad, media, bajo, sedentario) -
    - 5. (20-50, ciudad, baja, medio, paro)       +

## Espacio de versiones de Mitchell: ejemplo 1

### ❑ Proceso de aprendizaje

- ❑ Ejemplo 1: (20-50, ciudad, baja, nulo, sedentario) +
- ❑  $G = \{(X1, X2, X3, X4, X5)\}$  *hipótesis vacía*
  - ❑  $E = \{(20-50, ciudad, baja, nulo, sedentario)\}$  *1º ej. positivo*
- ❑ Ejemplo 2: (>50, campo, baja, medio, físico) -
- ❑  $G = \{(20-50, X2, X3, X4, X5), (X1, ciudad, X3, X4, X5), (X1, X2, X3, nulo, X5), (X1, X2, X3, X4, sedentario)\}$   
*especializar lo imprescindible para que no sea consistente*
  - ❑  $E = \{(20-50, ciudad, baja, nulo, sedentario)\}$  *nada a eliminar*
- ❑ Ejemplo 3: (>50, ciudad, baja, alto, sedentario) +
- ❑  $G = \{(X1, ciudad, X3, X4, X5), (X1, X2, X3, X4, sedentario)\}$   
*eliminar lo no consistente*
  - ❑  $E = \{(X1, ciudad, baja, X4, sedentario)\}$  *generalizar*

## Espacio de versiones de Mitchell: ejemplo 1

### ❑ Proceso de aprendizaje (continuación)

- ❑ Ejemplo 3: (>50, ciudad, baja, alto, sedentario) + (repetimos por claridad)
- ❑  $G = \{(X1, ciudad, X3, X4, X5), (X1, X2, X3, X4, sedentario)\}$
  - ❑  $E = \{(X1, ciudad, baja, X4, sedentario)\}$
- ❑ Ejemplo 4: (20-50, ciudad, media, bajo, sedentario) -
- ❑  $E = \{(X1, ciudad, baja, X4, sedentario)\}$  *nada a eliminar*
  - ❑  $G = \{(X1, ciudad, baja, X4, X5), (X1, X2, baja, X4, sedentario)\}$   
*especializar lo imprescindible para que no sea consistente*
- ❑ Ejemplo 5: (20-50, ciudad, baja, medio, paro) +
- ❑  $G = \{(X1, ciudad, baja, X4, X5)\}$   
*eliminar lo no consistente*
  - ❑  $E = \{(X1, ciudad, baja, X4, X5)\}$  *generalizar*

## Espacio de versiones de Mitchell

- ☐ Sesgo del sistema (elementos necesarios para que pueda aprender):
  - ☐ Conjunción de literales positivos
- ☐ Desarrolla una búsqueda en anchura
  - ☐ Trabajando con un conjunto de descripciones posibles en la franja contenida entre G y E ([espacio de versiones](#))
- ☐ El conjunto E contiene una única descripción
  - ☐ Si tuviera 2, p1 y p2, la intersección de ambas también sería consistente y más específica.
- ☐ Tiene [baja tolerancia al ruido](#):
  - ☐ un ejemplo mal etiquetado puede hacer que no converja

## Espacio de versiones de Mitchell: ejemplo 2

- ☐ Representación del [concepto coche como marco](#):

(origen = x1;	$x1 \in \{\text{Japón, EEUU, UK, Italia,...}\}$
marca = x2;	$x2 \in \{\text{Honda, Toyota, Chrysler, Fiat,...}\}$
color = x3;	$x3 \in \{\text{azul, blanco, amarillo, verde,...}\}$
década = x4;	$x4 \in \{\text{1950, 1960, 1970, 1980, 1990, 2000,...}\}$
tipo = x5)	$x5 \in \{\text{económico, lujo, deportivo,...}\}$
- ☐ Un ejemplo o instancia del concepto [coche](#) en [particular](#):

(origen = Japón; marca = Honda; color = azul;  
década = 1970; tipo = económico)
- ☐ La descripción de conceptos se puede poner en términos de ranuras y valores. Por ejemplo, el concepto [coche económico japonés](#)

(origen = Japón; marca = x2; color = x3;  
década = x4; tipo = económico)

## Espacio de versiones de Mitchell: ejemplo 2

- ❑ Ejemplo: inducir el concepto **coche económico japonés**
  - ❑ Se inicializa  $G = \{(x1, x2, x3, x4, x5)\}$  *hipótesis vacía*
  - ❑ Primer ejemplo positivo
    - ❑ (**origen = Japón; marca = Honda; color = azul; década = 1970; tipo = económico**)
    - ❑  $E = \{(\text{Japón, Honda, azul, 1970, económico})\}$  *1º ej. positivo*
  - ❑ Segundo ejemplo: negativo
    - ❑ (**origen = Japón; marca = Toyota; color = verde; década = 1970; tipo = deportivo**)
    - ❑  $G = \{(x1, \text{Honda}, x3, x4, x5), (x1, x2, \text{azul}, x4, x5), (x1, x2, x3, x4, \text{económico})\}$  *especializar para no consistencia*
  - ❑ Tercer ejemplo: positivo
    - ❑ (**origen = Japón; marca = Toyota; color = azul; década = 1990; tipo = económico**)
    - ❑  $G = \{(x1, x2, \text{azul}, x4, x5), (x1, x2, x3, x4, \text{económico})\}$  *eliminar*
    - ❑  $E = \{(\text{Japón, x2, azul, x4, económico})\}$  *generalizar*

## Espacio de versiones de Mitchell: ejemplo 2

- ❑ Tercer ejemplo: positivo
  - ❑ (**origen = Japón; marca = Toyota; color = azul; década = 1990; tipo = económico**)
  - ❑  $G = \{(x1, x2, \text{azul}, x4, x5), (x1, x2, x3, x4, \text{económico})\}$
  - ❑  $E = \{(\text{Japón, x2, azul, x4, económico})\}$
- ❑ Cuarto ejemplo: negativo
  - ❑ (**origen = EEUU; marca = Chrysler; color = azul; década = 1980; tipo = económico**)
  - ❑  $G = \{(\text{Japón, x2, azul, x4, x5}), (\text{Japón, x2, x3, x4, económico})\}$
- ❑ Quinto ejemplo: positivo
  - ❑ (**origen = Japón; marca = Honda; color = blanco; década = 1980; tipo = económico**)
  - ❑  $G = \{(\text{Japón, x2, x3, x4, económico})\}$  *eliminar*
  - ❑  $E = \{(\text{Japón, x2, x3, x4, económico})\}$  *generalizar*
- ❑ Dado que G y E convergen en un **conjunto unitario** se llegó a la descripción del concepto buscada



## Espacio de versiones : El problema del ruido

- ❑ Problema: un error en el etiquetado de un ejemplo
  - ❑ Puede causar que se pade del espacio de versiones el concepto objetivo
  - ❑ P.E.: si el 3º coche ejemplo se etiqueta como negativo por error,
    - ❑ no se llega al concepto de coche económico japonés
- ❑ Posible solución: mantener varios conjuntos G y E
  - ❑ Un conjunto G es consistente con todas las instancias de entrenamiento, otro lo será con todas menos una, otro con todas menos dos, etc. (ídem para E)
  - ❑ Cuando se presenta una inconsistencia, el algoritmo cambia de conjuntos de entrenamiento
  - ❑ Mantener múltiples espacios de versiones puede resultar muy costoso

## ----- ID3 (Quinlan) -----

- ❑ Inductivo y aprendizaje de conceptos
  - ❑ A partir de un conjunto de ejemplos,
  - ❑ Construye un **árbol de decisión** (no necesariamente binario)
  - ❑ Que le permite clasificar casos nuevos (sin etiquetar)
  - ❑ Determina qué atributo es más importante (raíz)
- ❑ Objetivo: construir el árbol de decisión más simple
  - ❑ Que clasifique bien los ejemplos de entrenamiento
- ❑ Sesgo: descripciones que incluyen conjunciones, disyunciones y negaciones (*formas normales disyuntivas; cjtos. de reglas*)
- ❑ Ventaja: permite aprender varios conceptos simultáneamente (no sólo un concepto y su negación)
- ❑ **No es incremental:** usa la totalidad del conjunto inicial de ejemplos

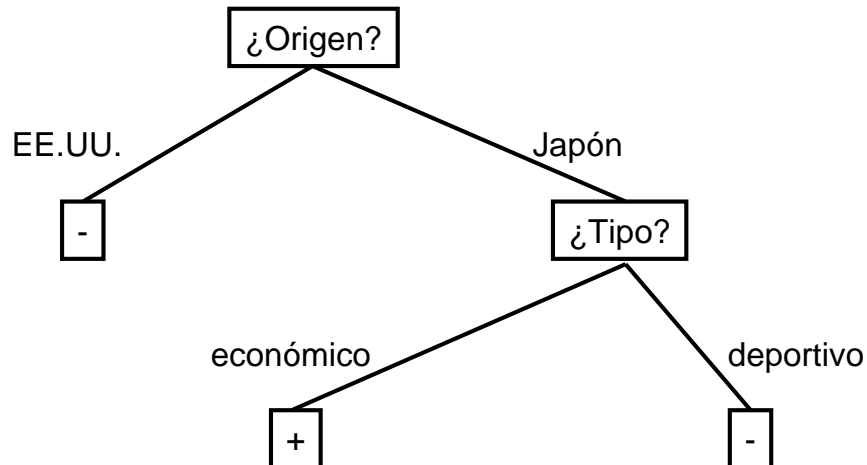
## Árboles de decisión: ejemplo 1

### ❑ Especificación con un árbol de decisión de un concepto

#### ❑ ¿Cuál?

#### ❑ “Coche económico japonés”

❑ Origen, marca, color, década, tipo



#### ❑ Clasificación booleana: ejemplos positivos o negativos

## Árboles de decisión: ejemplo 2

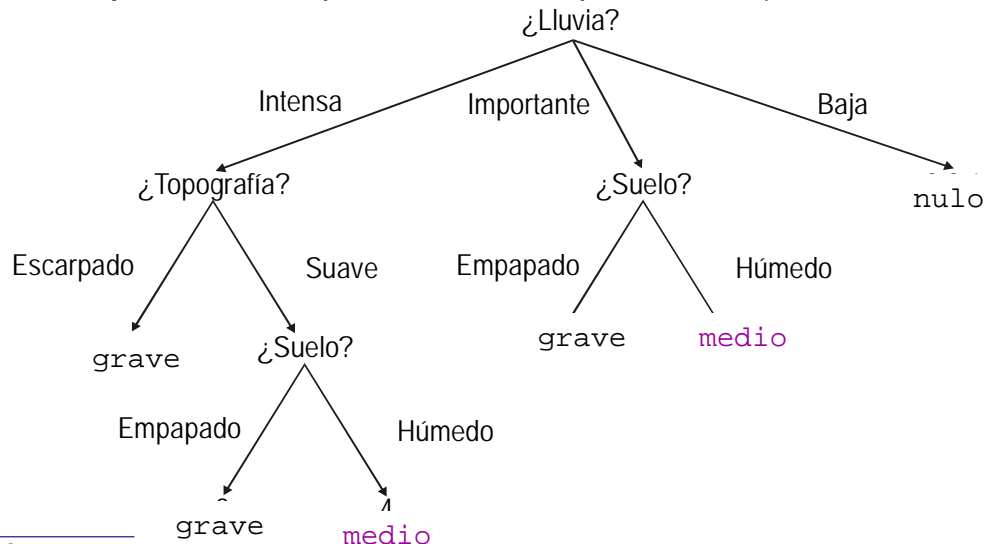
### ❑ Aprender a clasificar diferentes situaciones de riesgo de inundación (en función de los atributos lluvia, suelo y topografía)

#### ❑ Ejemplos de entrenamiento:

Caso	Lluvia	Suelo	Topografía	Problema
1	intensa	empapado	escarpada	grave
2	intensa	empapado	suave	grave
3	intensa	húmedo	escarpada	grave
4	intensa	húmedo	suave	medio
5	importante	empapado	escarpada	grave
6	importante	húmedo	escarpada	medio
7	importante	húmedo	suave	medio
8	baja	empapado	escarpada	nulo
9	baja	húmedo	escarpada	nulo
10	baja	húmedo	suave	nulo

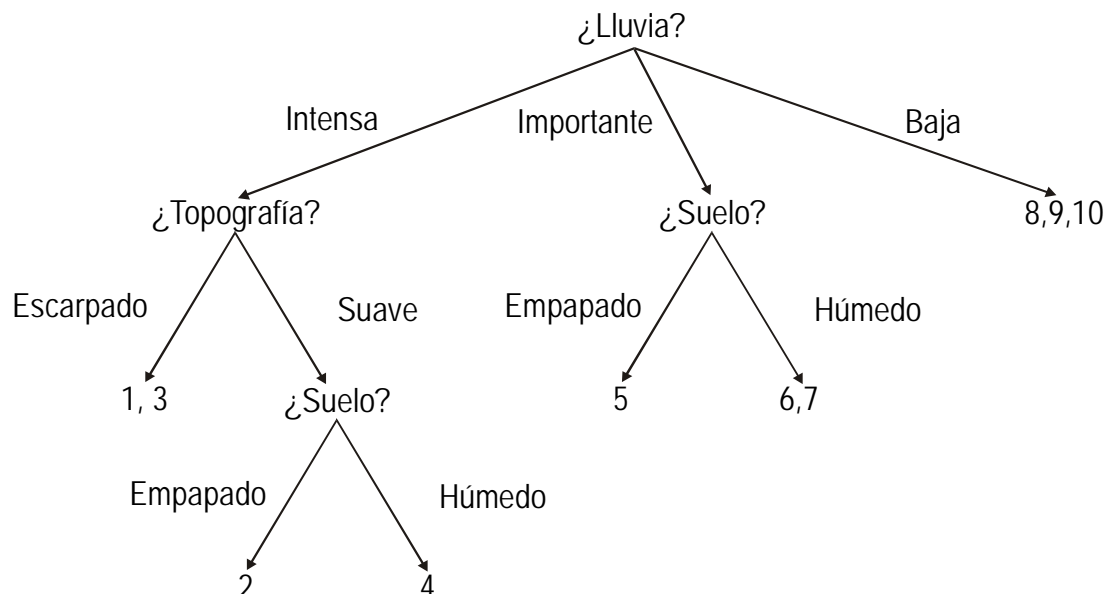
# Árboles de decisión

- Representación natural de los criterios de decisión de las personas
  - Nodos: preguntas (atributos o características)
  - Arcos: posibles respuestas (valores posibles en los ejemplos)
  - Hojas están etiquetadas con la predicción (clase de ese ejemplo)



# Árboles de decisión

Criterio principal en una inundación? : Lluvia ← la raíz



## Construcción de árboles de decisión con ID3

- ❑ Árbol de decisión trivial
  - ❑ Un camino de la raíz a una hoja por cada ejemplo
  - ❑ Memoriza exactamente los ejemplos de entrenamiento, no extrae ningún patrón, luego no sirve para extrapolar a ejemplos nuevos
- ❑ Buscamos el árbol de decisión *más pequeño* que sea consistente con los ejemplos. ¡Problema intratable...!
  - ❑ El orden en el que se eligen los atributos para construir un árbol de decisión afecta al tamaño del árbol (*1º + importante*)
  - ❑ Usando heurísticas, puede encontrarse uno más bien pequeño
- ❑ Solución: ID3 (Quinlan, 83) o su sucesor C4.5 (Quinlan, 93)
  - ❑ Algoritmo de búsqueda heurística
  - ❑ Optimización local: escalada
  - ❑ *Teoría de la información* para estimar el mejor candidato
  - ❑ Se consigue complejidad lineal

## Construcción de árboles de decisión con ID3

- ❑ ID3 construye automáticamente un árbol de decisión dadas diferentes instancias positivas y negativas del concepto destino
- ❑ Algoritmo iterativo, que comienza eligiendo un subconjunto aleatorio de ejemplos de entrenamiento: *ventana*
- ❑ Construye un árbol que clasifica todos los ejemplos de la ventana, y prueba con ejemplos fuera de la ventana
  - ❑ Si los ejemplos son correctamente clasificados, el algoritmo finaliza
  - ❑ Si no, se añaden a la ventana más ejemplos y el proceso se repite
- ❑ Los árboles se construyen creando nodos a partir de los *atributos* que proporcionan más información: *más discriminantes*
- ❑ Cuando se llega a atributos que dividen perfectamente las instancias de entrenamiento en subconjuntos cuyos miembros participan con una etiqueta común (positivo o negativo) la ramificación ha terminado y los nodos hoja están etiquetados

- ❑ Datos sobre los que opera el algoritmo:
  - ❑ Conjunto de ejemplos de entrenamiento
    - ❑  $E$
  - ❑ Conjunto de las clases a las que pueden pertenecer los ejemplos
    - ❑  $Cl_1, Cl_2, \dots, Cl_N$
  - ❑ Conjunto de atributos definidos sobre los ejemplos
    - ❑  $A, B, C, \dots$
  - ❑ Conjuntos con los valores posibles para cada atributo
    - ❑  $A_1, \dots, A_K, B_1, \dots, B_L, C_1, \dots, C_M, \dots$

- ❑ El árbol se construye de arriba a abajo, trabajando por niveles
- ❑ En cada iteración del algoritmo se pretende:
  - ❑ Obtener el atributo en base al cual ramificar el nodo problema
  - ❑ Se seleccionará aquél que mejor discrimine entre el conjunto de ejemplos
    - ❑ Heurística para obtener árboles pequeños (en profundidad)
    - ❑ El atributo más discriminante será aquél que conduzca a un estado con menor **entropía**
- ❑ La entropía (Shannon, 1948) mide la ausencia de homogeneidad de un conjunto de ejemplos con respecto a su clase
  - ❑ Es una medida estándar del desorden (*utilizada en física y en la teoría de la información: 0 es homogeneidad total*)
- ❑ Ganancia de información = diferencia entre la entropía del conjunto original y la de los subconjuntos obtenidos

- ❑ La entropía se define utilizando resultados de la **teoría de la información**
- ❑ La **entropía inicial** de un nodo  $X$  (antes de clasificar los ejemplos que contiene en base a alguno de los atributos) es:

$$E(X) = - \sum_{j=1}^N P_X(Cl_j) \cdot \log_2 P_X(Cl_j)$$

donde

$$\log_2 x = \begin{cases} 0 & \text{si } x = 0 \\ \log_2 x & \text{en otro caso} \end{cases}$$

y la probabilidad de una clase  $Cl_j$  en el nodo  $X$  es

$$P_X(Cl_j) = \frac{|\text{nº de ejemplos correspondientes a } Cl_j \text{ en } X|}{|\text{nº total de ejemplos en } X|}$$

para  $j \in [1..N]$

- ❑ La **entropía final** del nodo  $X$  al ramificar utilizando el atributo  $A$ , es igual a la suma de las entropías de los nodos resultantes de fijar el valor del atributo multiplicadas por la probabilidad de cada valor

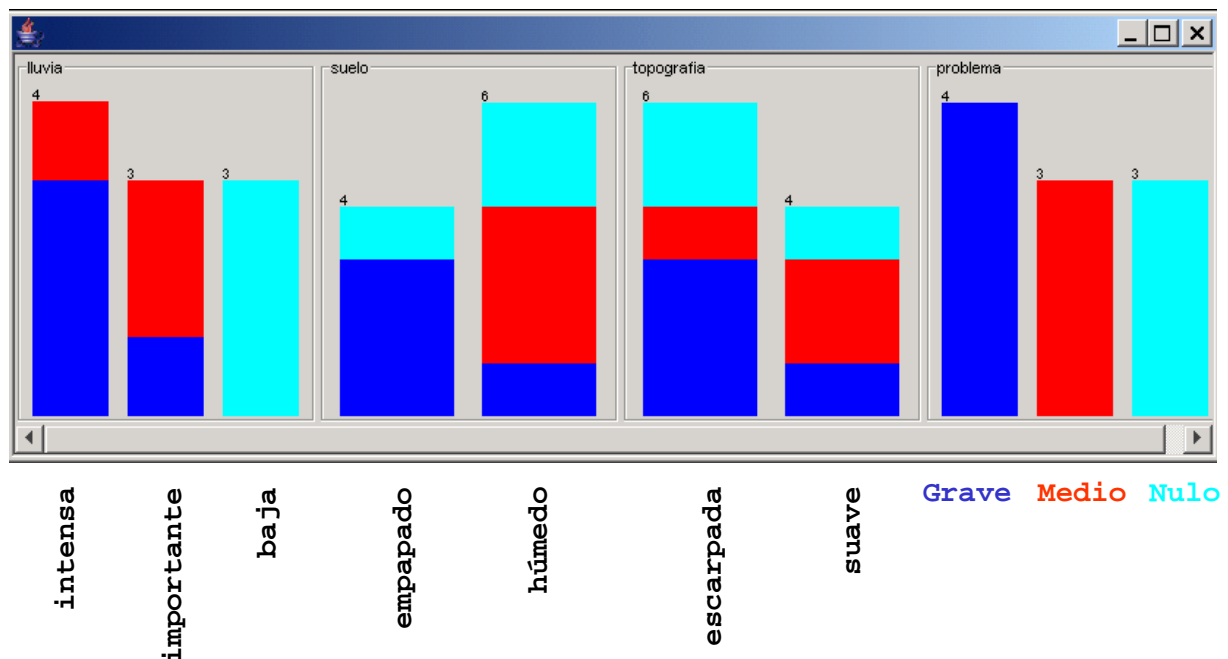
$$E_A(X) = \sum_{i=1}^k P_X(A_i) \cdot E(A_i)$$

donde la probabilidad  $P_X(A_i)$  es

$$P_X(A_i) = \frac{|\text{nº de ejemplos en } X \text{ con atributo } A = A_i|}{|\text{nº total de ejemplos en } X|}$$

- ❑ Para cada atributo se calcula la disminución de entropía causada por su utilización
  - ❑ Disminución de entropía $_A(X) = E(X) - E_A(X)$
  - ❑ Disminución de entropía $_B(X) = E(X) - E_B(X)$
  - ❑ Disminución de entropía $_C(X) = E(X) - E_C(X) \dots$
- ❑ En cada nodo, se selecciona aquel atributo que mayor disminución de entropía provoca
- ❑ Aplicado al ejemplo
  - ❑ Hay 3 clases
    - ❑  $Cl_1$  problema grave,  $Cl_2$  problema medio,  $Cl_3$  problema nulo
  - ❑ Y 3 atributos
    - ❑ A lluvia, B suelo, C topografía

## ID3: ejemplo



## ID3: ejemplo

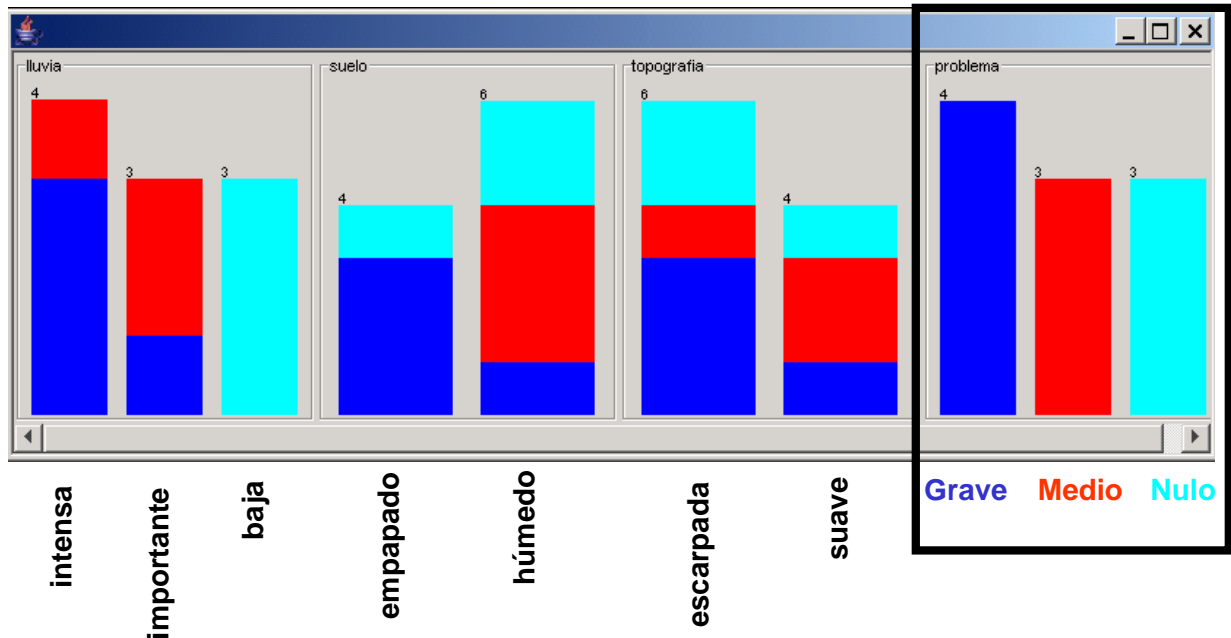
### Entropía inicial en la raíz del árbol: (del problema global)

$$P(\text{grave}) = 0,4$$

$$P(\text{medio}) = 0,3$$

$$P(\text{nulo}) = 0,3$$

$$E(\text{raíz}) = -0,4 \log_2 0,4 - 0,3 \log_2 0,3 - 0,3 \log_2 0,3 = 1,571$$



## ID3: ejemplo

### Entropía final clasificando según lluvia (A):

$A_1$ : lluvia intensa,

$A_2$ : lluvia importante,

$A_3$ : lluvia baja

$$E(A_1) = -0,75 \log_2 0,75 - 0,25 \log_2 0,25 = 0,811$$

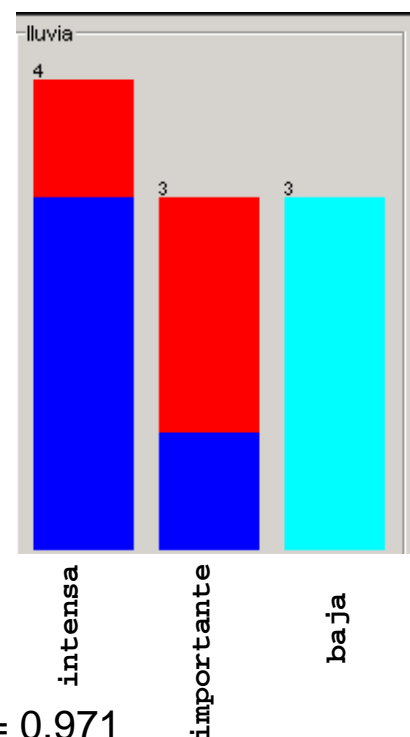
$$E(A_2) = 0,918$$

$$E(A_3) = 0$$

$$E_A(\text{raíz}) = 0,4 \cdot 0,811 + 0,3 \cdot 0,918 = 0,6$$

Probabilidad de intensa

entropía



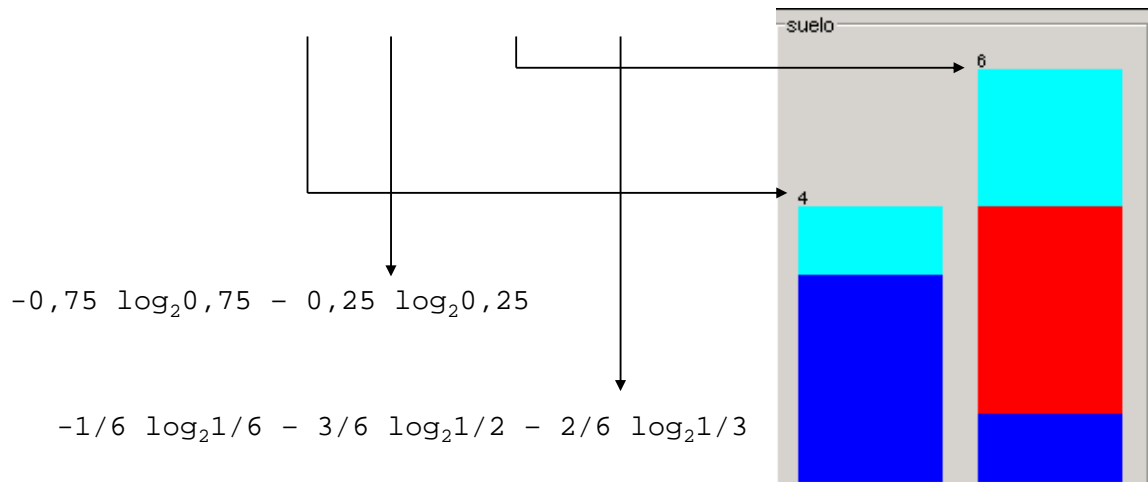
$$\text{Disminución de entropía}_A(\text{raíz}) = 1,571 - 0,60 = 0,971$$



## ID3: ejemplo

- Entropía final clasificando según suelo(B):

$$E_B(\text{raíz}) = 0,4 \cdot 0,811 + 0,6 \cdot 1,459 = 1,20$$



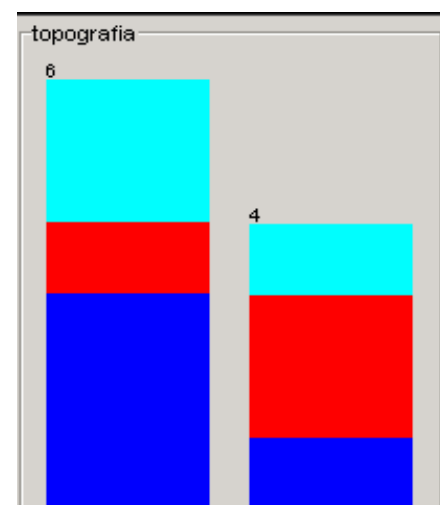
$$\text{Disminución de entropía}_B(\text{raíz}) = 1,571 - 1,20 = 0,371$$

## ID3: ejemplo

- Entropía final clasificando según topografía (C):

$$E_C(\text{raíz}) = 0,6 \cdot 1,459 + 0,4 \cdot 1,50 = 1,475$$

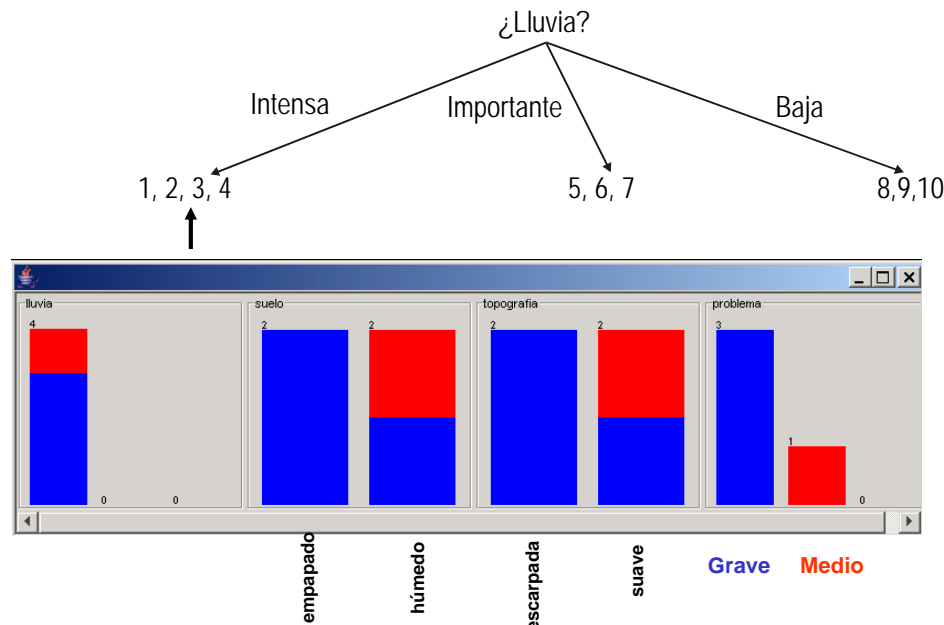
Diagram illustrating the entropy calculation for the 'topografía' attribute. The decision tree splits on 'topografía' into two branches: '4' and '6'. The '4' branch leads to a node with entropy calculation:  $-1/4 \log_2 1/4 - 2/4 \log_2 2/4 - 1/4 \log_2 1/4$ . The '6' branch leads to a node with entropy calculation:  $-1/6 \log_2 1/6 - 3/6 \log_2 1/2 - 2/6 \log_2 1/3$ .



$$\text{Disminución de entropía}_C(\text{raíz}) = 1,571 - 1,475 = 0,096$$

- La mayor disminución de entropía se consigue con el atributo A y por ello éste es el seleccionado para el primer nivel del árbol

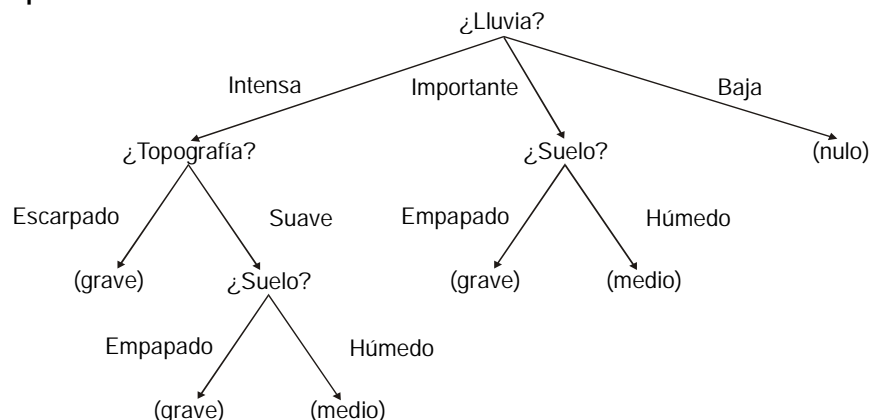
## ID3: ejemplo



- En la siguiente iteración se vuelve a aplicar el algoritmo sobre cada uno de los tres nuevos nodos, considerando en cada uno el subconjunto de ejemplos obtenido y habiendo eliminado el atributo lluvia del conjunto de atributos

## ID3

- Terminación:
  - La expansión de un nodo se detiene cuando todos sus ejemplos pertenecen a la misma clase ( $\equiv$  entropía nula)
  - El proceso se detiene cuando no se puede seguir expandiendo ningún nodo
  - A las hojas se les asigna la clase a la que pertenecen todos sus ejemplos



## ID3: refinamientos

- ❑ Cuando es muy grande, del conjunto de ejemplos disponibles se escoge una **ventana** para construir el árbol (*incremental*)
  - ❑ Si el resto de ejemplos es clasificado correctamente, se detiene
  - ❑ Si no, se añaden elementos a la ventana y se repite el proceso
  - ❑ Análisis empíricos demuestran que esta estrategia es más eficiente que considerar todos los ejemplos desde el principio (*no claro*)
- ❑ **Ruido** en los datos: para mejorar la tolerancia
  - ❑ Sólo se ramifica un nodo cuando la disminución de entropía está por encima de un determinado umbral. A cada nodo terminal se le asigna la clase de la mayoría de sus ejemplos
- ❑ Ha evolucionado dando lugar al algoritmo **C4.5** (Quinlan 93)
  - ❑ Soluciona un pequeño problema de ID3: tiene una cierta tendencia a favorecer la elección de atributos con muchos valores posibles, lo que redundaría en una peor generalización de las observaciones
  - ❑ Última versión: C5.0 comercial (*RuleQuest Research, de Quinlan*)

## ----- Aprendizaje por Descubrimiento -----

- ❑ Clasificaciones del aprendizaje:
  - ❑ Grado de realimentación: no supervisado
  - ❑ Paradigma: inductivo
  - ❑ Qué aprende: agrupa en clases
- ❑ Aprendizajes NO supervisados (sin “profesor”)
  - ❑ Formación de taxonomías y clustering conceptual    ← veremos
  - ❑ Descubrimiento de leyes cualitativas
  - ❑ Descubrimiento de leyes cuantitativas
- ❑ Aprendizaje por descubrimiento:
  - ❑ Transformar o reorganizar las experiencias
  - ❑ De manera que se puedan extraer nuevas conclusiones

# Formación de taxonomías

- ❑ Escenario (situación):
  - ❑ Conjunto de objetos / observaciones reales
    - ❑ No hay nadie que aporte información sobre a qué clase pertenecen
  - ❑ Descripción del objeto:
    - ❑ N pares <atributo, valor> (espacio N dimensiones)
  - ❑ Similitud entre objetos: distancia entre los puntos que los representan
- ❑ Objetivos:
  - ❑ Agrupar ejemplos con características similares para formar categorías
  - ❑ Crear árbol de clasificación con grupos a diferentes alturas de abstracción
  - ❑ maximizar la similitud “intra-cluster”
  - ❑ minimizar la similitud entre “clusters”

## Formación de taxonomías : Algoritmo

- ❑ FASE 1: Crear Matriz de Distancias o su opuesta de Similitudes
- ❑ FASE 2: Agrupación de Individuos (técnica jerárquica)
  - ❑ Cada objeto es un cluster
  - ❑ Usando M. Distancias:
    - ❑ Ordenar por parecidos todos los individuos
    - ❑ Agrupar de dos en dos los objetos más parecidos
    - ❑ Crear un objeto (cluser) nuevo que sustituya a los dos
      - ❑ Sus coordenadas son la media de las de ambos
  - ❑ Repetir FASE 2 hasta que solo un cluster con todos los individuos
    - ❑ Forman un Dendograma , es el Arbol de Clasificación
  - ❑ Formar clases a diferentes niveles
    - ❑ Se corta el Dendograma a diferentes alturas
- ❑ Algunos ejemplos
  - ❑ P. 700 Inteligencia Artificial (J.T.Palma, R. Martín)
  - ❑ <http://anthropologynet.files.wordpress.com/2008/02/population-dendogram.jpg>

## --Paradigma: Aprendizaje analítico/deductivo--

- ☐ Se aplica habitualmente a **resolución de problemas**
- ☐ Parten de muy pocos ejemplos (normalmente 1) junto con una teoría del dominio
- ☐ El ejemplo se utiliza para guiar las cadenas deductivas que deben seguirse para resolver nuevos problemas o para formular reglas de control de búsqueda que posibiliten una aplicación más eficiente del conocimiento del dominio
- ☐ El ejemplo del que se parte suele ser una traza de la resolución del problema junto con anotaciones sobre la justificación de las decisiones adoptadas. Este conocimiento debe ser generalizado de acuerdo con la teoría del dominio
- ☐ No intentan ampliar lo que sabe hacer el sistema sino mejorar su eficiencia

## ---Aprendizaje basado en explicaciones (EBL)---

- ☐ Clasificaciones del aprendizaje:
  - ☐ Realimentación: supervisado
  - ☐ Paradigma: analítico / deductivo (usa teoría del dominio)
  - ☐ Qué aprende: definición estructural de un concepto
    - ☐ A partir de una descripción funcional y un ejemplo

## ❑ Entrada:

- ❑ C: **Concepto objetivo** (a aprender) descrito funcionalmente
- ❑ E: Ejemplo positivo (de C)
- ❑ TD: Teoría del dominio (axiomas y reglas de inferencia)
  - ❑ Permite probar que E es un ejemplar de C usando TD
- ❑ CO: Criterio de operacionalidad : debe cumplirlo cualquier
  - ❑ Nueva descripción estructural más genérica

## ❑ Salida: descripción estructural más genérica

# Ejemplo EBL

**Concepto objetivo:** TAZA

**Definición funcional:**

$\text{RECIPIENTE\_ABIERTO}(x) \cap \text{ESTABLE}(x) \cap \text{ALZABLE}(x) \leftrightarrow \text{TAZA}(x)$

**Ejemplo de entrenamiento:**

TIENE\_PARTE(OBJ1, CONCAVIDAD1)

COLOR(OBJ1, ROJO)

CONCAVIDAD(CONCAVIDAD1)

ORIENTADA\_HACIA\_ARRIBA(CONCAVIDAD1)

TIENE\_DUEÑO(OBJ1, SAM)

TIENE\_PARTE(OBJ1, FONDO1)

FONDO(FONDO1)

PLANO(FONDO1)

LIGERO(OBJ1)

TIENE\_PARTE(OBJ1, ASA1)

ASA(ASA1)

LONGITUD(ASA1, 5)

## Ejemplo EBL

### Teoría del dominio:

**TIENE\_PARTE(OBJ, F)  $\cap$  FONDO(F)  $\cap$  PLANO(F)  $\rightarrow$  ESTABLE(OBJ)**

**TIENE\_PARTE(OBJ, C)  $\cap$  CONCAVIDAD(C)  $\cap$   
ORIENTADA\_HACIA\_ARRIBA(C)  $\rightarrow$  RECIPIENTE\_ABIERTO(OBJ)**

**TIENE\_PARTE(OBJ, A)  $\cap$  ASA(A)  $\cap$  LIGERO(OBJ)  $\rightarrow$  ALZABLE(OBJ)**

### Criterio operacional:

**El concepto tiene que definirse en términos de los predicados usados en el ejemplo.**

## Aprendizaje EBL

### ☐ Pasos:

- ☐ Crea explicación: porqué un ejemplo satisface una definición funcional
- ☐ Crea una descripción estructural con la explicación

### ☐ El proceso de aprendizaje tiene 2 pasos:

- ☐ Se usa la teoría del dominio para construir una explicación (demostración) de que el ejemplo de entrenamiento es un ejemplo positivo del concepto objetivo. Los nodos terminales del árbol de explicación tienen que ser operativos
- ☐ Transformar los nodos terminales en un conjunto de condiciones suficientes para que la demostración siga siendo válida (generalizar la explicación de acuerdo con la teoría del dominio)

## Ejemplo EBL

### Resultado del proceso de aprendizaje:

$\text{TIENE\_PARTE}(X, Y) \cap \text{CONCAVIDAD}(Y) \cap$   
 $\text{ORIENTADA\_HACIA\_ARRIBA}(Y) \cap$   
 $\text{TIENE\_PARTE}(X, Z) \cap \text{FONDO}(Z) \cap \text{PLANO}(Z) \cap$   
 $\text{TIENE\_PARTE}(X, W) \cap \text{ASA}(W) \cap \text{LIGERO}(X)$

## Métodos analíticos: Ventajas e Inconvenientes

### ☐ Ventajas:

- ☐ Proporcionan justificación lógica de la descripción del concepto que han obtenido
- ☐ Sólo necesitan un ejemplo positivo
- ☐ Permiten disyunciones y conjunciones
- ☐ Tolerancia al ruido

### ☐ Inconvenientes:

- ☐ Requieren considerable conocimiento del dominio  
(→ menor aplicabilidad)
- ☐ Sustituyen la búsqueda en el espacio de descripciones de conceptos por búsqueda en el espacio de explicaciones)



# ----- Paradigma: Algoritmos genéticos -----

**Tipos de Aprendizaje: paradigmas**

**(recordatorio)**

- ☐ **Aprendizaje inductivo**
- ☐ **Aprendizaje analítico o deductivo**
- ☐ **Algoritmos genéticos**
- ☐ **Métodos conexionistas (enfoque subsimbólico)**

☐ **Clasificaciones del aprendizaje:**

- ☐ **Realimentación: supervisado**
- ☐ **Paradigma: Algoritmos genéticos**
- ☐ **Qué aprende: encuentra solución a un Problema**
  - ☐ **Crea generaciones de soluciones y las evalúa**
  - ☐ **Se queda con la mejor**