



Inteligencia Artificial e Ingeniería del Conocimiento

- ✎ I. A. Clásica
- ✎ “Nuevos” enfoques de la I. A.
 - ✓ Agentes Inteligentes
 - ✓ Aprendizaje
 - ✓ Introducción
 - Aprendizaje inductivo
 - Aprendizaje de árboles de decisión
 - ✓ Planteamiento conectivista. Redes Neuronales
 - Arquitecturas
 - Algoritmos
 - ✓ Aplicaciones
 - ✓ Lógicas multivaluadas
 - ✓ Computación evolutiva

Intel. Artif e Ing. del Conocimiento

Aprendizaje. R. Neuronales

1

©Vidal Moreno Rodilla. Dpto Inf. y Autom. USAL



Introducción I

- ✎ Representación de la información
 - ✓ Fundamentos
 - ✓ Lógica de predicados
 - ✓ Redes semánticas
 - ✓ Marcos/Guiones
 - ✓ Herramientas
 - ✓ Lenguajes de programación (LISP, PROLOG)
 - Programación Orientada a Objetos
 - ✓ Shells (MYCIN, NEXPERT, JESS)
- ✎ Búsqueda de solución
 - ✓ Fundamentos
 - ✓ Algoritmos de búsqueda
 - ✓ Utilización de heurísticas

Intel. Artif e Ing. del Conocimiento

Aprendizaje. R. Neuronales

2

©Vidal Moreno Rodilla. Dpto Inf. y Autom. USAL



Introducción II

- ✎ Comportamientos inteligentes revisados
 - ✓ Procesamiento simbólico
 - ✓ Adquirir
 - ¿Procesamiento de las percepciones?
 - ✓ Memorizar
 - Organización de la información (Reglas, Redes semánticas)
 - ✓ Recuperación de la información
 - Métodos de inferencia
 - ✓ Resolver problemas
 - ✓ Problemas de búsqueda
 - Backtracking, Exploración de grafos
 - ✓ Utilización de heurística
 - ✓ Razonamiento automático
 - ✓ Sistemas expertos



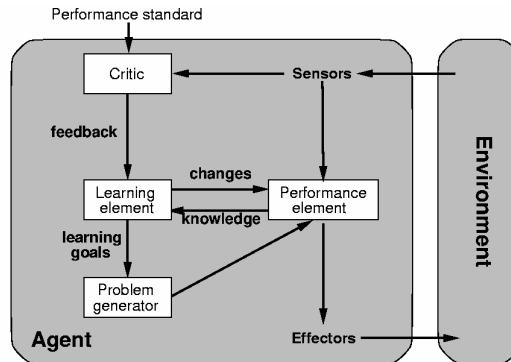
Introducción III

- ✎ Comportamientos Inteligentes
 - ✓ Percepción
 - ✓ Sensores (Cámaras, Microfonos,)
 - ✓ Aplicaciones
 - Visión artificial
 - Preprocesamiento
 - Análisis
 - Interpretación (Clasificación de patrones)
 - Lenguaje natural
 - Reconocimiento del lenguaje (ViaVoice)
 - Analizador del lenguaje (Traductores??)
 - ✓ APRENDIZAJE (EXTRACCION DE LA INFORMACION)
 - ✓ Generación de KB, Formulación de problemas, ... (realizada por experto o Ingeniero del Conocimiento)
 - ✓ ¿APRENDIZAJE AUTOMATICO?



Introducción IV

- ✎ Elemento aprendiz
 - ✓ Determina la evolución que ha de tener el agente
- ✎ Elemento de comportamiento
 - ✓ Selecciona la actuación del agente
- ✎ Crítico
 - ✓ Define si el agente se comporta de forma adecuada de acuerdo con la información exterior
- ✎ Generador de problemas
 - ✓ Sugiere acciones que lleven a la adquisición de más conocimiento



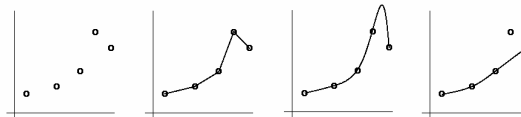
Introducción (y V)

- ✎ Agente de aprendizaje
 - ✓ Elemento de comportamiento. Componentes
 - ✓ Mapeo directo de condiciones y acciones
 - ✓ Información para extraer propiedades relevantes de la secuencia de percepciones
 - ✓ Información de cómo evoluciona el mundo
 - ✓ Información sobre los resultados de las posibles acciones
 - ✓ Estados deseables
 - ✓ Objetivos que describen las clases de estados cuyo cumplimiento maximiza la utilidad del agente
 - ✓ Se trabaja en un modo de aprendizaje supervisado
 - ✓ Se conocen parejas de entradas y salidas deseadas
 - ✓ En aprendizaje no-supervisado el agente no dispone de las salidas deseadas



Aprendizaje inductivo I

- ✎ Dado un conjunto de pares ejemplo
 - ✓ (entradas,salida): $(x, f(x))$
 - ✓ La tarea de la **inferencia de inducción pura (INDUCCION)** es generar una función que aproxime a $f(x)$
 - ✓ $h \rightarrow$ Hipótesis
 - ✓ La tarea es realizada por el Elemento de Comportamiento
 - Ejemplo:
 - Predicción de series temporales
 - Lógica (Método de árbol de decisión)



Aprendizaje inductivo (y II)

- ✎ Estructura de agente reflejo simple con aprendizaje

```

global ejemplos ← {}
function ELEMENTO-COMPORTAMIENTO-REFLEJO (percepción)
return acción

if (percepción,a) in ejemplos then return a
else
  h ← INDUCIR(ejemplos)
  return h(percepción)

procedure ELEMENTO-APRENDIZAJE-REFLEJO(percepción,acción)

input:  percepción; percepción de realimentación
        Acción; acción de realimentación

ejemplos ← ejemplos ∪ {(percepción, acción)}
  
```



Aprendizaje de árboles de decisión I

Árbol de decisión

- ✓ Descripción
 - ✓ Entrada: Situación descrita por una serie de propiedades
 - ✓ Salida: Lógica (Si/No)
- ✓ Representan funciones booleanas
- ✓ Tienen un predicado global (Objetivo)
- ✓ Ejemplo
 - ✓ Objetivo: ¿WillWait?
 - ✓ Descripción mediante predicados
 - Alternate, Bar, Fri/Sat, Hungry, Patrons, Price, Raining, Reservation, Type, WaitEstimate
 - ✓ Ejemplo de regla para aprender

$$(\forall r) Patrons(r, Full) \wedge WaitEstimate(r, 0-10) \wedge Hungry(r, N) \Rightarrow WillWait(r)$$



Aprendizaje de árboles de decisión II

Expresividad de los árboles

- ✓ Cualquier función booleana se puede expresar mediante un árbol de decisión
- ✓ Problemas
 - ✓ Función-mayoría (majority-function) si más de la mitad de las entradas son verdaderas

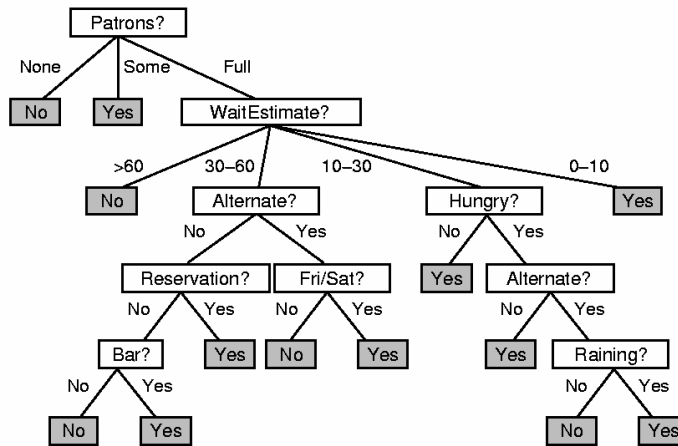
Objetivo

- ✓ Construir un árbol de decisión a partir de un conjunto de pares (entrada/salida)
 - ✓ CONJUNTO DE ENTRENAMIENTO
- ✓ Se dice que el predicado objetivo es un CLASIFICADOR
 - ✓ Se tratará de definir el tamaño del conjunto mínimo que permite realizar el aprendizaje



Aprendizaje de árboles de decisión III

Ejemplo: (Arbol de decisión)



Intel. Artif e Ing. del Conocimiento

Aprendizaje. R. Neuronales

11

©Vidal Moreno Rodilla. Dpto Inf. y Autom. USAL



Aprendizaje de árboles de decisión IV

Ejemplo: (Conjunto de entrenamiento)

Example	Attributes										Goal
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
X ₁	Yes	No	No	Yes	Some	\$\$\$	No	Yes	French	0-10	Yes
X ₂	Yes	No	No	Yes	Full	\$	No	No	Thai	30-60	No
X ₃	No	Yes	No	No	Some	\$	No	No	Burger	0-10	Yes
X ₄	Yes	No	Yes	Yes	Full	\$	No	No	Thai	10-30	Yes
X ₅	Yes	No	Yes	No	Full	\$\$\$	No	Yes	French	>60	No
X ₆	No	Yes	No	Yes	Some	\$\$	Yes	Yes	Italian	0-10	Yes
X ₇	No	Yes	No	No	None	\$	Yes	No	Burger	0-10	No
X ₈	No	No	No	Yes	Some	\$\$	Yes	Yes	Thai	0-10	Yes
X ₉	No	Yes	Yes	No	Full	\$	Yes	No	Burger	>60	No
X ₁₀	Yes	Yes	Yes	Yes	Full	\$\$\$	No	Yes	Italian	10-30	No
X ₁₁	No	No	No	No	None	\$	No	No	Thai	0-10	No
X ₁₂	Yes	Yes	Yes	Yes	Full	\$	No	No	Burger	30-60	Yes

Intel. Artif e Ing. del Conocimiento

Aprendizaje. R. Neuronales

12

©Vidal Moreno Rodilla. Dpto Inf. y Autom. USAL



Aprendizaje de árboles de decisión V

Algoritmo

```

function APRENDIZAJE-ARBOL-
DECISION(ejemplos, atributos, defecto) returns arbol-decision

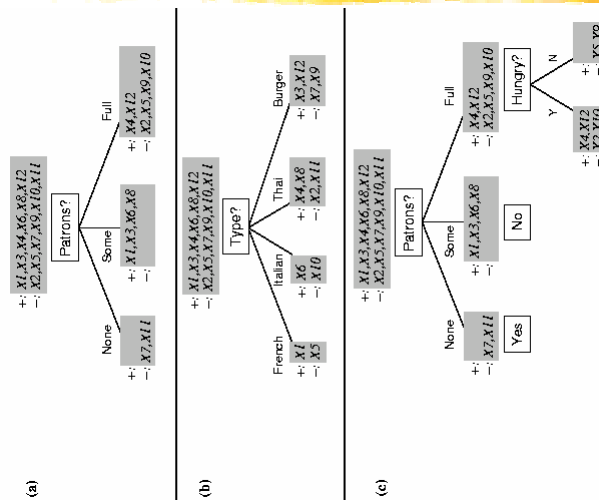
  inputs:      ejemplos; Conjunto de ejemplos
                atributos; Conjunto de atributos
                defecto; El valor por defecto para el predicado

  if ejemplos está vacío then return defecto
  else if todos los ejemplos tienen la misma clasificación then return la
clasificación
  else if atributos está vacía then return VALOR-MAYORIA(ejemplos)
  else
    best ← ELEGIR-ATRIBUTO(atributos, ejemplos)
    arbol ← Nuevo árbol de decisión con raíz best
    for each valor vi de best do
      ejemplosi ← {elementos de ejemplos con valor best vi}
      subarbol ← APRENDIZAJE-ARBOL-DECISION(ejemplosi, atributos - best, VALOR-MAYORIA(ejemplosi))
      añadir una rama a arbol-decision con etiqueta vi y subarbol
    end
  return arbol-decision

```



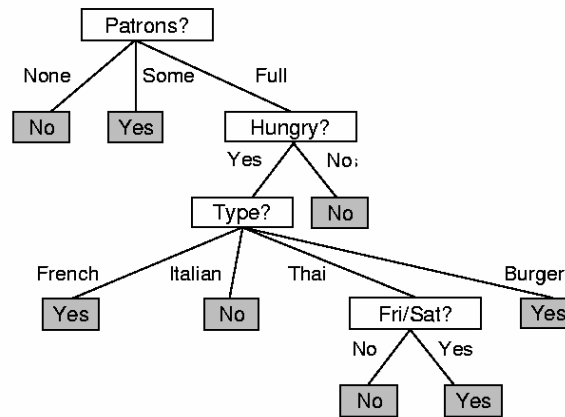
Aprendizaje de árboles de decisión VI





Aprendizaje de árboles de decisión VII

Arbol de decisión obtenido



Intel. Artif e Ing. del Conocimiento

Aprendizaje. R. Neuronales

15

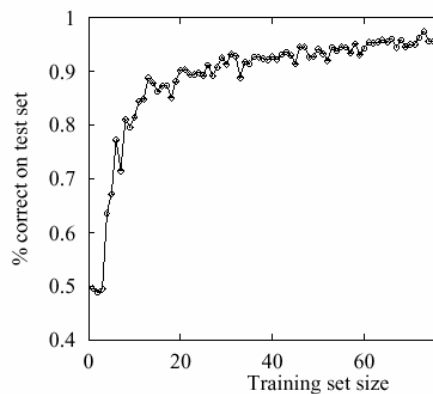
©Vidal Moreno Rodilla. Dpto Inf. y Autom. USAL



Aprendizaje de árboles de decisión VIII

Medida de prestaciones

- ✓ Considerar un conjunto de ejemplos grande
- ✓ Dividirlo en dos conjuntos de entrenamiento y prueba
- ✓ Utilizarlo para generar una hipótesis
- ✓ Medir sobre el conjunto de test el porcentaje de aciertos
- ✓ Repetir los pasos hasta alcanzar una medida correcta



Intel. Artif e Ing. del Conocimiento

Aprendizaje. R. Neuronales

16

©Vidal Moreno Rodilla. Dpto Inf. y Autom. USAL



Aprendizaje de árboles de decisión IX

- ✎ Aplicaciones
 - ✓ 1986. BP desarrolla un Sistema Experto para diseñar los sistemas de separación de Gas-Oil
 - ✓ Depende de un número elevado de atributos
 - ✓ 2500 reglas (Mayor Sistema Experto del mundo)
 - ✓ 10 personas/año
 - ✓ 100 personas/día (Utilizando la BD sobre diseños existentes)
- ✎ Es válido para hipótesis expresan en términos de lógica de predicados
 - ✓ Función con una definición lógica
- ✎ Las redes neuronales permiten representar cualquier tipo de función



Aprendizaje de árboles de decisión X

- ✎ Selección de atributos
 - ✓ Teoría de la información

$$I(P(v_1), \dots, P(v_n)) = \sum_{i=1}^n -P(v_i) \log_2 P(v_i)$$

$$I\left(\frac{1}{2}, \frac{1}{2}\right) = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} = 1 \text{ bit}$$
 - ✓ Información Atributo
 - ✓ n favorables, p no favorables

$$I\left(\frac{p}{p+n}, \frac{n}{p+n}\right)$$
 - Información antes de probar
 - ✓ Un atributo con v valores

$$\text{Restante}(A) = \sum_{i=1}^v \frac{p_i + n_i}{p+n} I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right)$$
 - ✓ Se selecciona

$$\text{Ganancia}(A) = I\left(\frac{p}{p+n}, \frac{n}{n+p}\right) - \text{Restante}(A)$$
- Seleccionado → $\text{Ganancia}(\text{Patrons}) = 0.54 \text{ bits}$
 $\text{Ganancia}(\text{Type}) = 0 \text{ bits}$