

# Inteligencia Artificial

## Tema 8. Aprendizaje

José Manuel Fuertes García  
jmf@ujaen.es

Departamento de Informática  
Universidad de Jaén



18 de mayo de 2017

# Objetivos

- ▶ Conocer cómo se puede construir un agente que pueda mejorar su comportamiento estudiando sus propias experiencias.
- ▶ Conocer el aprendizaje inductivo a partir de observaciones.

# Índice

Aprendizaje de observaciones

Aprendizaje inductivo

Aprendizaje de árboles de decisión

Redes Neuronales

Otros Métodos de Aprendizaje

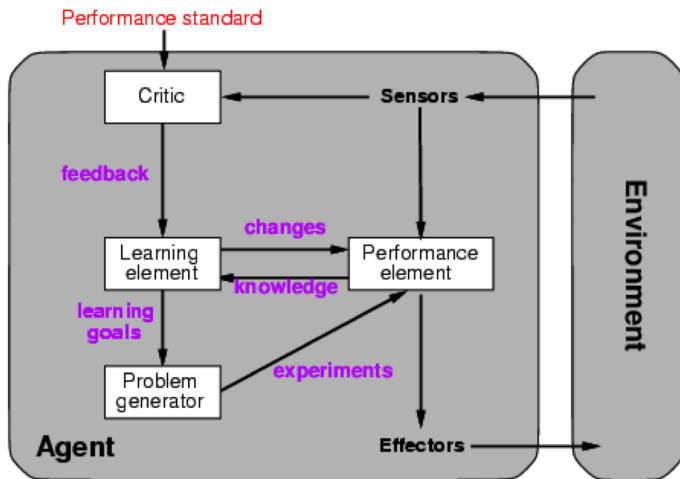
Medida del rendimiento del aprendizaje

# Aprendizaje

**Idea general:** El aprendizaje consiste en utilizar las percepciones para mejorar la habilidad del agente para actuar en el futuro.

- ▶ El aprendizaje es esencial para entornos desconocidos, especialmente cuando al diseñador le falta conocimiento.
- ▶ El aprendizaje es útil como un método de construcción del sistema, es decir, trabaja sobre la idea de exponer al agente a la realidad más que intentar especificar todo lo posible.
- ▶ El aprendizaje modifica los mecanismos de decisión del agente para mejorar el rendimiento.

# Agente de aprendizaje



# Agente de aprendizaje

Un agente de aprendizaje se puede diseñar con:

- ▶ **elemento de acción**: decide qué acciones llevar a cabo, y
- ▶ **elemento de aprendizaje**: modifica el elemento de acción para tomar mejores decisiones.

El diseño de un elemento de aprendizaje viene determinado por:

- ▶ qué *componentes* del elemento de acción tienen que aprenderse
- ▶ qué *realimentación* está disponible para aprenderlos
- ▶ qué tipo de *elemento de rendimiento* se utiliza
- ▶ qué *tipo de representación* se usa para los componentes.

El tipo de realimentación determina el tipo de problema de aprendizaje.

# Tipos de aprendizaje

- ▶ **Supervisado:** Aprender una función a partir de ejemplos de sus entradas y salidas.
  - ▶ Requiere respuestas correctas para cada instancia.
- ▶ **No supervisado:** Aprender a partir de patrones de entradas para los que no se especifican las salidas.
- ▶ **Por refuerzo:** Aprender a partir de recompensas o penalizaciones ocasionales.
  - ▶ Es el más general.

# Índice

Aprendizaje de observaciones

**Aprendizaje inductivo**

Aprendizaje de árboles de decisión

Redes Neuronales

Otros Métodos de Aprendizaje

Medida del rendimiento del aprendizaje



# Aprendizaje inductivo

En su forma más sencilla consiste en aprender una función a partir de ejemplos:

- ▶  $f$  es la función objetivo,
- ▶ un ejemplo es un par  $x, f(x)$ ,
- ▶ el problema de **aprendizaje inductivo** es encontrar una *hipótesis*  $h$  tal que  $h \approx f$  dado un conjunto de ejemplos de entrenamiento.

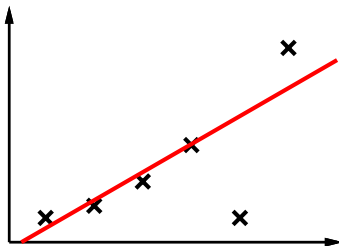
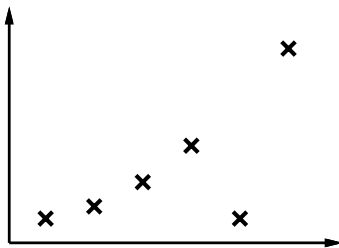
Este es un modelo muy simplificado de aprendizaje:

- ▶ ignora conocimiento a priori
- ▶ asume un entorno determinístico y observable
- ▶ asume que disponemos de un conjunto de ejemplos
- ▶ asume que el agente quiere aprender  $f$

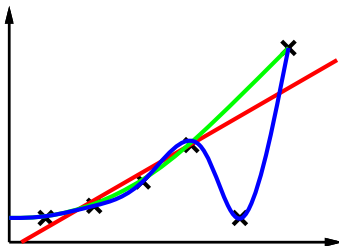
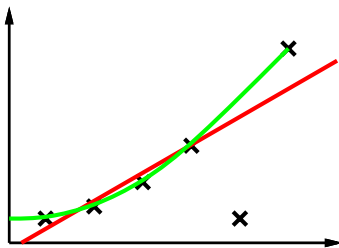
# Aprendizaje inductivo

- ▶ A la función  $h$  se le denomina *hipótesis*.
- ▶ Una buena hipótesis estará bien generalizada si puede predecir ejemplos que no se conocen.
  - ▶ Éste es el problema de inducción fundamental.
- ▶ Ej. de algoritmo de aprendizaje inductivo:
  - ▶ Construir o ajustar  $h$  para que coincida con  $f$  en los ejemplos de entrenamiento.
  - ▶ Se elige el espacio de hipótesis  $H$ .
- ▶ La hipótesis es consistente si verifica todos los datos.

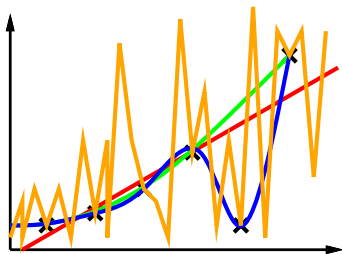
## Ejemplo aprendizaje inductivo



## Ejemplo aprendizaje inductivo



## Ejemplo aprendizaje inductivo



*Elección entre hipótesis consistentes:* A través del principio de la navaja de Ockham (maximizar una combinación entre consistencia y simplicidad).

# Índice

Aprendizaje de observaciones

Aprendizaje inductivo

Aprendizaje de árboles de decisión

Redes Neuronales

Otros Métodos de Aprendizaje

Medida del rendimiento del aprendizaje

# Definición de árboles de decisión

- ▶ Constituyen un tipo de aprendizaje supervisado.
- ▶ Se trata de un método basado en modelos que representa el conocimiento aprendido mediante estructuras que generalizan de forma explícita casos de entrenamiento:
  - ▶ De esa forma se pueden aplicar directamente a la clasificación de otros casos no vistos.
  - ▶ Permiten que el conocimiento aprendido y soportado sea inteligible por el usuario.
- ▶ Un árbol de decisión es una representación gráfica de un procedimiento para clasificar o evaluar un concepto.
- ▶ Intuitivamente un árbol de decisión es una colección de condiciones organizadas jerárquicamente.

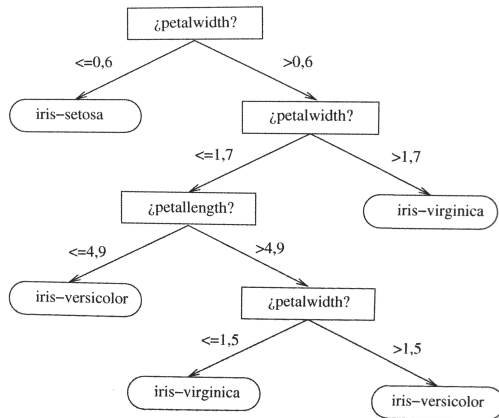
# Definición de árboles de decisión

- ▶ Toma como entrada un objeto o una situación descrita como un conjunto de atributos y devuelve una decisión.
  - ▶ Valor de salida discreto  $\rightarrow$  clasificación.
  - ▶ Valor de salida continuo  $\rightarrow$  regresión.
- ▶ Estructura de un árbol de decisión:
  - ▶ Cada hoja es una categoría de la variable de salida (o un valor continuo o función en regresión).
  - ▶ Cada nodo de decisión especifica una prueba simple a realizar.
  - ▶ Los descendientes de cada nodo son los posibles resultados de la prueba.



## Ejemplo de árbol de decisión

Problema: Clasificación de plantas *iris* (*iris setosa*, *iris versicolor* o *iris virginica*) en función de las características del ancho y largo de su pétalo y/o sépalo.



## Ejemplo de árbol de decisión

Problema: Clasificación de plantas *iris* (*iris setosa*, *iris versicolor* o *iris virginica*) en función de las características del ancho y largo de su pétalo y/o sépalo.

- El anterior árbol es directamente expresable como un conjunto de reglas del tipo SI ENTONCES SI NO donde las condiciones son la conjunción de tests sobre los atributos y las hojas son los consecuentes.

```

SI petalwidth <= 0,6 ENTONCES Iris – setosa
SI NO // petalwidth > 0,6
| SI petalwidth <= 1,7
| | SI petallength <= 4,9 ENTONCES Iris – versicolor
| | SI NO // petallength > 4,9
| | | SI petalwidth <= 1,5 ENTONCES Iris – virginica
| | | SI NO Iris – versicolor // petalwidth > 1,5
| SI NO Iris – virginica // petalwidth > 1,7
  
```

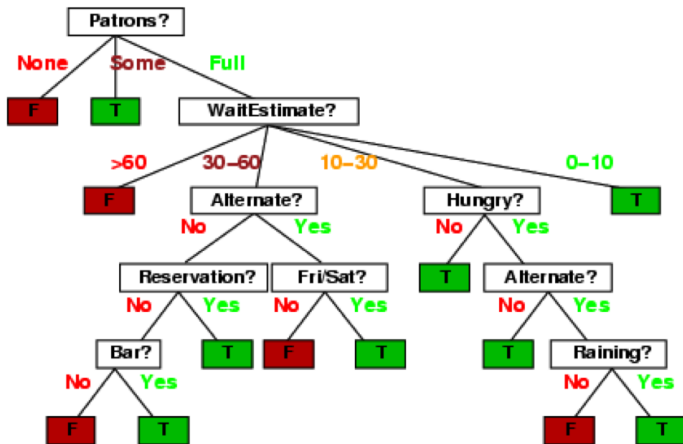
## Ejemplo de árbol de decisión

Problema: decidir si esperar o no en un restaurante por una mesa, en base a los siguientes atributos:

1. Alternativa: Hay un restaurante alternativo cerca?.
2. Bar: Hay una zona de bar confortable para esperar?.
3. Vier/Sab: verdad si es viernes o sábado.
4. Hambriento: si estamos hambrientos.
5. Clientes: número de clientes en el restaurante (ninguno, algunos, lleno).
6. Precio: Rango de precios del restaurante.
7. Lloviendo: Está lloviendo?.
8. Reserva: Se ha hecho una reserva?.
9. Clase de restaurante (francés, italiano, thailandés, hamburguesería).
10. Tiempo de espera estimado (0-10, 10-30, 30-60, >60).

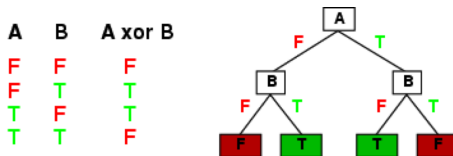
# Árboles de decisión

Es una representación posible para hipótesis.



# Expresividad de árboles de decisión

- ▶ Pueden expresar cualquier función de los atributos de entrada.
- ▶ Trivialmente, hay un árbol de decisión consistente para todo conjunto de entrenamiento con un camino hacia un nodo hoja para todo ejemplo (salvo en funciones  $f$  no determinísticas en  $x$ ) pero probablemente no generalice nuevos ejemplos.



- ▶ Se prefieren árboles de decisión más compactos.

# Aprendizaje de árboles de decisión

- ▶ Objetivo: Encontrar un árbol pequeño consistente con los ejemplos de entrenamiento.
- ▶ Idea: Elegir (recursivamente) el atributo más significativo como raíz del subárbol.
  - ▶ Esta aproximación aplica una estrategia *divide y vencerás* en la que inicialmente se tienen todos los ejemplos, conjunto de datos de entrenamiento, y a cada paso se selecciona el mejor atributo como test del nodo.
  - ▶ Se distribuyen los ejemplos entre los distintos nodos descendientes y se vuelve a proceder como en el nodo padre.

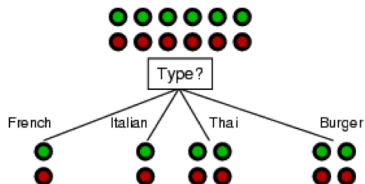
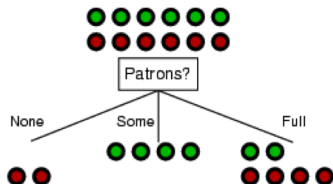
```

function DTL(examples, attributes, default) returns a decision tree
  if examples is empty then return default
  else if all examples have the same classification then return the classification
  else if attributes is empty then return MODE(examples)
  else
    best ← CHOOSE-ATTRIBUTE(attributes, examples)
    tree ← a new decision tree with root test best
    for each value  $v_i$  of best do
      examplesi ← {elements of examples with best =  $v_i$ }
      subtree ← DTL(examplesi, attributes - best, MODE(examples))
      add a branch to tree with label  $v_i$  and subtree subtree
    return tree

```

# Aprendizaje de árboles de decisión

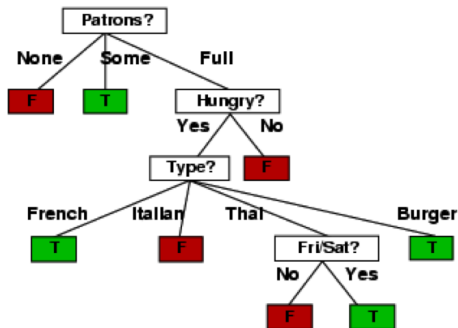
- Una buena variable divide los ejemplos en subconjuntos que idealmente son todos positivos o todos negativos.



- Cientes (*patrons*) es la mejor elección.

# Aprendizaje de árboles de decisión

- ▶ Árbol de decisión aprendido a partir de 12 ejemplos



- ▶ Con mayor número de ejemplos se aprendería un árbol diferente (una hipótesis más compleja).



# Índice

Aprendizaje de observaciones

Aprendizaje inductivo

Aprendizaje de árboles de decisión

**Redes Neuronales**

Otros Métodos de Aprendizaje

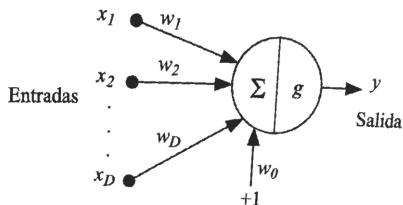
Medida del rendimiento del aprendizaje

# ¿Qué es una Red Neuronal?

- ▶ Una Red Neuronal Artificial (RNA) es un paradigma de procesamiento de información inspirado en la estructura neuronal del cerebro.
- ▶ Las RNA están compuestas por un cierto número de elementos de procesamiento o *neuronas* que trabajan al unísono para resolver un problema específico.
- ▶ Las RNA actuales están basadas en un modelo matemático de neurona propuesto por McCulloch y Pitts en 1943.
  - ▶ En dicho modelo cada neurona recibe un conjunto de entradas  $x_1, x_2, \dots, x_D$  y devuelve una única salida  $y$ .

## Estructura de una Red Neuronal

- ▶ Dentro de una Red Neuronal Artificial existen numerosas conexiones entre las distintas neuronas que la forman.
- ▶ Simulan las conexiones interneuronales del cerebro y pueden establecerse con mayor o menor intensidad.
- ▶ En el caso de las RNA esta intensidad se establece a partir de los *pesos sinápticos*, de tal forma que cada entrada  $x_i$  de una neurona está afectada por un peso  $w_i$ .



# RNA: Tipos de Problemas

Existen dos tipos básicos de problemas que una RNA puede resolver:

- ▶ **Problemas de clasificación:** El objetivo consiste en crear un procedimiento mediante el cual un nuevo caso representado por unos atributos observados o características que constituyen los datos de entrada a la red, se asigne a una de entre un conjunto de clases predefinidas. Se trata en definitiva de un proceso de *reconocimiento de patrones*.
- ▶ **Problemas de regresión:** Tratan de encontrar una función continua de ciertas variables, para lo que obtiene un número en función de los atributos de entrada a la red. Se trata en definitiva de un proceso de *ajuste de funciones*.

## RNA: Entrenamiento o Aprendizaje

Las RNA reciben unos datos de entrada que se transforman para producir una salida para clasificar o ajustar un función:

- ▶ Suponiendo que la red tenga un número suficiente de neuronas, podrá ajustar cualquier función continua con una cierta precisión con tan sólo escoger los valores adecuados para los parámetros ajustables de dicha red: sus pesos sinápticos.
- ▶ Dichos pesos sinápticos serán el medio donde la red almacene su conocimiento sobre el problema a resolver.
- ▶ La modificación de dichos parámetros se realiza a través de un proceso de entrenamiento o aprendizaje de un conjunto de ejemplos representativos del problema.

## RNA: Aprendizaje Supervisado

En este tipo de aprendizaje se modela la correspondencia entre un conjunto de  $N$  ejemplos o patrones de entrada  $x^1, x^2, \dots, x^N$  y las salidas deseadas  $t^1, t^2, \dots, t^N$ , minimizando la función de error que mide la diferencia entre esas salida y las obtenidas por la red  $y^1, y^2, \dots, y^N$ :

- ▶ El *Perceptrón* es la red de neuronas artificiales más sencilla: compuesta únicamente por una capa de neuronas de entrada y otra capa de neuronas de salida.
- ▶ Las limitaciones de las redes de una sólo capa hicieron que se plantease la necesidad de implementar redes en las que se aumentase el número de capas introduciendo otras intermedias entre la capa de entrada y la de salida, de manera que se pudiese implementar cualquier función con el grado de precisión deseado: Aparición del *Perceptrón Multicapa*.

# RNA: Aprendizaje No Supervisado

- ▶ En este tipo de aprendizaje se trata de representar los datos de entrada del sistema de forma que esta representación refleje la estructura estadística que los define: encontrar patrones o características que sean significativas en los datos de entrada.
- ▶ No disponemos de ninguna salida con la que comparar el rendimiento del método, por lo que la única información que tendrá el sistema es la que proporcionen los datos de entrada.

# Máquina de Vectores Soporte (SVM)

- ▶ Las máquinas de vectores soporte están fundamentadas en sólidos principios teóricos que proporcionan un buen rendimiento en una gran variedad de aplicaciones prácticas.
- ▶ Basan su funcionamiento en el principio de minimización del riesgo estructural: No se centran en construir sistemas que cometan pocos errores, sino que intentan construir modelos fiables en los cuales se pueda tener una gran confianza, aunque se cometan algunos errores más en el entrenamiento del sistema.
- ▶ Las SVM tratan de obtener modelos que estructuralmente tengan poco riesgo de cometer errores ante datos futuros.



# Índice

Aprendizaje de observaciones

Aprendizaje inductivo

Aprendizaje de árboles de decisión

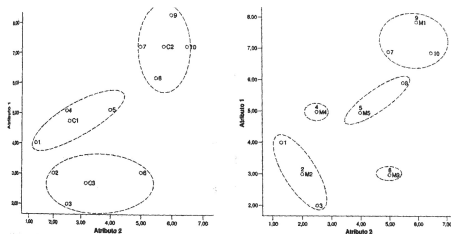
Redes Neuronales

Otros Métodos de Aprendizaje

Medida del rendimiento del aprendizaje

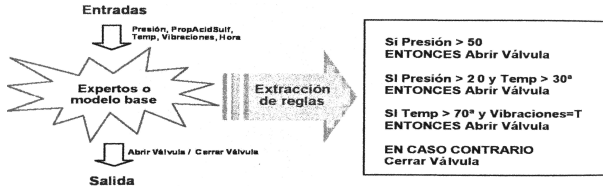
# Técnicas de Agrupamiento

- ▶ Las técnicas de agrupamiento o *clustering* se pueden definir como *técnicas de clasificación no supervisada de elementos (observaciones, datos o vectores de características) en grupos (clusters)*.
- ▶ Dos elementos pertenecientes a un agrupamiento válido deben ser más parecidos entre si que aquellos que estén en grupos distintos.
- ▶ Partiendo de esta idea se desarrollan técnicas de agrupamiento que dependerán de cómo sean los datos de partida, de qué medidas de semejanza se estén utilizando y de qué clases de problemas se estén resolviendo.



# Técnicas de Extracción de Reglas

- ▶ Permiten convertir distintos tipos de modelos en conjuntos de reglas.
- ▶ Los sistemas de reglas son más fáciles de entender por los humanos.
- ▶ Son más fáciles de integrar en sistemas expertos, ontologías y otros tipos de sistemas basados en el conocimiento.



# Técnicas de Extracción de Reglas

- ▶ Técnicas de extracción de reglas a partir de modelos de caja negra.
  - ▶ Se desarrollaron inicialmente para la extracción de conocimiento en sistemas expertos.
  - ▶ Los expertos emplean generalmente reglas explícitas, *a ojo* o *reglas instintivas con parte subconsciente*, lo que hace que un modelo transcrito no pueda usarse de forma automatizada.
  - ▶ También se han desarrollado para la extracción de reglas a partir de redes neuronales.
- ▶ Pasos seguidos por estos métodos:
  1. Definición del dominio del problema, recopilación de casos de uso verificados por expertos.
  2. Formulación del conocimiento, en el que a partir de técnicas de aprendizaje automático se obtienen prototipos.
  3. Verificación del conocimiento obtenido.

# Índice

Aprendizaje de observaciones

Aprendizaje inductivo

Aprendizaje de árboles de decisión

Redes Neuronales

Otros Métodos de Aprendizaje

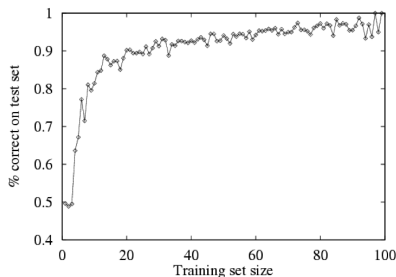
Medida del rendimiento del aprendizaje

# Medida de rendimiento

Cómo podemos saber que  $h \approx f$ ?

1. Utilizando teoremas de teoría de aprendizaje computacional o estadístico.
2. Probar  $h$  en un conjunto nuevo de ejemplos.

La curva de aprendizaje es el porcentaje de acierto en el conjunto de test en base al tamaño del conjunto de entrenamiento.



# Resumen

- ▶ El aprendizaje es necesario para entornos desconocidos y para post-poner el diseño.
- ▶ Un agente que aprende está compuesto por un elemento de rendimiento y un elemento de aprendizaje.
- ▶ En aprendizaje supervisado, el objetivo es encontrar la hipótesis más simple que sea aproximadamente consistente con el conjunto de entrenamiento.
- ▶ El rendimiento del aprendizaje se determina midiendo la precisión en un conjunto de test.