Inteligencia Artificial II Curso 2004–2005

# Tema 4: Aprendizaje de conceptos

José A. Alonso Jiménez Francisco Jesús Martín Mateos José Luis Ruiz Reina

Dpto. de Ciencias de la Computación e Inteligencia Artificial
UNIVERSIDAD DE SEVILLA

# Contenido

- Introducción al aprendizaje automático
- Aprendizaje de conceptos
  - Notación y terminología
  - Aprendizaje como búsqueda
  - Orden de generalidad
  - Algoritmo Find-S
  - Espacio de versiones
  - Eliminación de candidatos
  - Clasificación de nuevas instancias
  - Sesgo inductivo

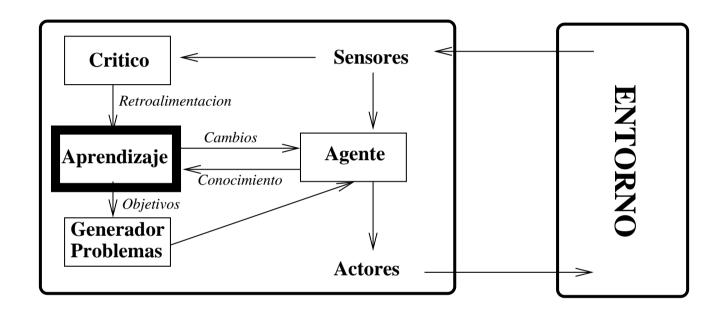
# Aprendizaje

- Definiciones de aprendizaje:
  - Cualquier cambio en un sistema que le permita realizar la misma tarea de manera m'as eficiente la próxima vez  $(H.\ Simon)$
  - Modificar la representación del mundo que se está percibiendo (R. Michalski)
  - Realizar cambios útiles en nuestras mentes (M. Minsky)
- Aprendizaje automático: construir programas que mejoran automáticamente con la experiencia
- Ejemplos de tareas:
  - Construcción de bases de conocimiento a partir de la experiencia

 $\mathbf{C}_{\mathbf{C}}\mathbf{I}_{\mathbf{A}}$ 

- Clasificación y diagnóstico
- Minería de datos, descubrir estructuras desconocidas en grandes grupos de datos
- Resolución de problemas, planificación y acción

# Agente con aprendizaje (Russel y Norvig, 1998)



- Sistema de aprendizaje: realiza cambios al sistema en función del rendimiento
- Agente: actúa
- Crítico: evalúa el rendimiento
- Generador de Problemas: sugiere nuevas experiencias que "entrenan"

# Tipos de aprendizaje y paradigmas

- Tipos de aprendizaje
  - Supervisado
  - No supervisado
  - Con refuerzo
- Paradigmas
  - Aprendizaje por memorización
  - Clasificación (Clustering)
  - Aprendizaje inductivo
  - Aprendizaje por analogía
  - Descubrimiento
  - Algoritmos genéticos, <u>redes neuronales</u>

# Aprendizaje de conceptos (ejemplo)

- Ejemplos de días en los que hacer (o no hacer) deportes acuáticos:
  - Representación como una lista de pares atributo-valor

Cielo	Temperatura	Humedad	Viento	Agua	Previsión	Hacer_Deporte
Soleado	Templada	Normal	Fuerte	Templada	Igual	Sí
Soleado	Templada	Alta	Fuerte	Templada	Igual	Sí
Lluvia	Fría	Alta	Fuerte	Templada	Cambio	No
Soleado	Templada	Alta	Fuerte	Fría	Cambio	Sí

• Objetivo: aprender el concepto "Días en los que se hace deporte"

 $\mathbf{C}_{\mathbf{C}}\mathbf{I}_{\mathbf{A}}$ 

# Notación y terminología (I)

- ullet Conjunto X de instancias
  - Ejemplos de instancias:

```
< Sol, Templada, Normal, Fuerte, Templada, Igual > 
< Nublado, Fria, Alta, Fuerte, Templada, Cambio > 
< Lluvia, Alta, Baja, Sin\_viento, Caliente, Igual >
```

- Un concepto es un subconjunto de X, usualmente dado por su función característica  $c: X \to \{1,0\}$ 
  - Instancias positivas y negativas
  - Ejemplo de concepto: si x es un día en el que se hace deporte, c(x)=1. En caso contrario c(x)=0
- Concepto (o función) objetivo: el que se desea aprender
  - En principio, no se conoce
  - Sólo se conoce el valor de c para algunas instancias (ejemplos)

# Notación y terminología (II)

## • Conjunto de entrenamiento D:

- ejemplos de instancias x para las que se conoce el valor de la función objetivo c(x) (representado por  $\langle x, c(x) \rangle$ )
- En la tabla anterior, 3 ejemplos positivos y 1 negativo

## • Espacio de hipótesis H:

- Conjunto de funciones  $h:X\to\{1,0\}$  que en el proceso de aprendizaje se pueden considerar como posibles definiciones del concepto objetivo
- Usualmente, en H no están todos los posibles conceptos. Sesgo inductivo

## Objetivo del aprendizaje:

- Encontrar  $h \in H$  tal que para cualquier ejemplo  $< x, c(x) > \in D$ , se tenga h(x) = c(x) (hipótesis consistente con los ejemplos)
- Ejemplo de hipótesis aprendida: < Soleado, Templada, ?, Fuerte, ?, ? >

 $\mathbf{C}_{\mathbf{C}}\mathbf{I}_{\mathbf{A}}$ 

• Hipótesis del Aprendizaje Inductivo

# Ejemplo de espacio de hipótesis

- Un posible espacio de hipótesis:
  - Conjunción de restricciones sobre los valores de los atributos
  - Ejemplo de hipótesis: "Los días para hacer deporte son los de cielo soleado, viento fuerte y con previsión de que el tiempo siga igual"
- Representación compacta:
  - $\bullet$ usar ? para representar un valor cualquiera <br/>ó $\emptyset$ para representar que ningún valor es posible
  - Ejemplos:

```
< Soleado, Templada, ?, Fuerte, ?, ?>, <?, Fria, Alta, ?, ?, ?>, <\emptyset, \emptyset, \emptyset, \emptyset, \emptyset >
```

• Atención: los algoritmos y definiciones de este tema *NO* son específicos de este tipo de espacio de hipótesis

# Aprendizaje como búsqueda

- Aprendizaje de conceptos:
  - búsqueda en el espacio de hipótesis de una o varias hipótesis consistentes con los ejemplos del conjunto de entrenamiento.
- ¿Aprendizaje por enumeración?
- Problema: espacio de hipótesis exponencialmente grande

 $\mathbf{C}_{\mathbf{C}}\mathbf{I}_{\mathbf{A}}$ 

- Solución: estructurar el espacio de hipótesis
  - Permite explorar el espacio de hipótesis de manera exhaustiva sin necesidad de enumerarlo

# El orden de generalidad

- Una instancia  $x \in X$  satisface la hipótesis h si h(x) = 1
- Dadas  $h_1, h_2 \in H$ ,  $h_1$  es más general que  $h_2$  (y  $h_2$  es más específica que  $h_1$ ) si cualquier instancia que satisface  $h_2$  satisface  $h_1$ .

 $\mathbf{C}_{\mathbf{C}}\mathbf{I}_{\mathbf{A}}$ 

- Notación:  $h_1 \ge_g h_2$ . Generalidad estricta:  $h_1 >_g h_2$
- Ejemplo:

# El algoritmo Find-S

## • Pseudocódigo:

- 1. Inicialmente, h es la hipótesis más específica de H.
- 2. Por cada ejemplo positivo x del conjunto de entrenamiento:
  - Si h(x)=1, no hacer nada.
  - En otro caso, reemplazar h por la menor generalización h' de h, tal que h'(x)=1
- 3. Devolver h

# Find-S (ejemplo)

#### • Paso 0:

$$h0 = \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$$

#### • Paso 1:

#### • Paso 2:

#### • Paso 3:

#### • Paso 4:

## Comentarios sobre Find-S

- Propiedades de Find-S sobre espacios de hipótesis como conjunción de atributos:
  - Find-S encuentra una hipótesis de máxima especificidad que es consistente con todos los ejemplos positivos
  - Los ejemplos negativos se ignoran
- ¿Consistente con los ejemplos negativos?
  - Sí, si el concepto objetivo está en H (expresividad del espacio de hipótesis) y los ejemplos de entrenamiento son correctos (ausencia de ruido)
  - Ejemplo problemático

```
Positivos: < Sol, Templ, Fuerte >, < Lluvia, Fria, Fuerte > Negativos: < Luvia, Templ, Fuerte >
```

• ¿Por qué devolver sólo una hipótesis de las consistentes?

# Espacio de versiones

• Espacio de versiones:  $VS_{H,D} \equiv \{h \in H | h \text{ es consistente con } D\}$ 

## • Ejemplo:

Cielo	Temperatura	Humedad	Viento	Agua	Previsión	Hacer_Deporte
Soleado	Templada	Normal	Fuerte	Templada	Igual	Sí
Soleado	Templada	Alta	Fuerte	Templada	Igual	Sí
Lluvia	Fría	Alta	Fuerte	Templada	Cambio	No
Soleado	Templada	Alta	Fuerte	Fría	Cambio	Sí

#### Espacio de versiones:

# Una representación compacta del espacio de versiones

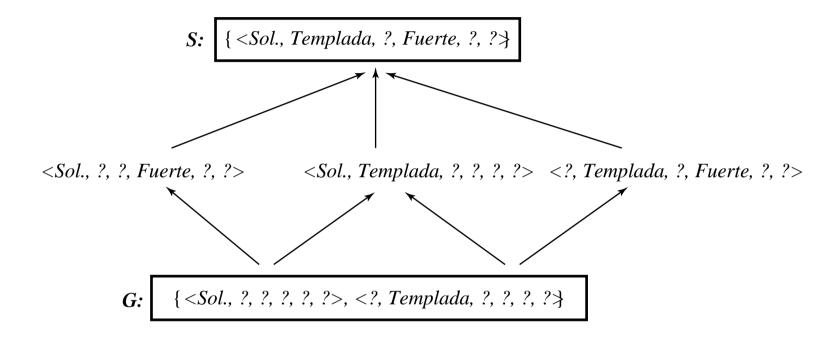
#### Definiciones

- $h \in H$  es una hipótesis de máxima generalidad (resp. de máxima especificidad) de H si no existe  $h' \in H$  tal que  $h' >_g h$  (resp.  $h >_g h'$ )
- Cota general G de un espacio de hipótesis respecto de D: elementos de máxima generalidad del espacio de versiones
- Cota específica S de un espacio de hipótesis respecto de D: elementos de máxima especificidad del espacio de versiones
- Teorema de representación del espacio de versiones
  - El espacio de versiones es el conjunto de hipótesis que están entre la cota general y la cota específica

 $\mathbf{C}_{\mathbf{C}}\mathbf{I}_{\mathsf{A}}$ 

•  $VS_{H,D} = \{ h \in H | (\exists s \in S) (\exists g \in G) (g \geq_q h \geq_q s) \}$ 

# Ejemplo de espacio de versiones



# Algoritmo de eliminación de candidatos

- 1. Sea G el conjunto de elementos de máxima generalidad de H.
- 2. Sea S el conjunto de elementos de máxima especificidad de H.
- 3. Para cada ejemplo d del conjunto de entrenamiento D:
  - 3.1 Si d es un ejemplo positivo, entonces:
    - 3.1.1 Eliminar de G cualquier hipótesis inconsistente con d.
    - 3.1.2 Para cada hipótesis s de S inconsistente con d:
      - \* Eliminar s de S.
      - \* Incluir en S todas las generalizaciones minimales h de s, tales que h es consistente con d y existe una hipótesis en G más general que h.
      - \* Eliminar de S aquellas hipótesis tales que exista en S otra hipótesis más específica.
  - 3.2 Si d es un ejemplo negativo, entonces:
    - 3.2.1 Eliminar de S cualquier hipótesis inconsistente con d.
    - 3.2.2 Para cada hipótesis g de G inconsistente con d:
      - \* Eliminar g de G.
      - \* Incluir en G todas las especializaciones minimales h de g, tales que h es consistente con d y existe una hipótesis en S más específica que h.
      - \* Eliminar de G aquellas hipótesis tales que exista en G otra hipótesis más general.

# Eliminación de candidatos (ejemplo)

- **Paso 0:** S0 =  $\{<\emptyset,\emptyset,\emptyset,\emptyset,\emptyset>\}$ , G0 =  $\{<?,?,?,?,?,?,\}$
- Paso 1:
  - Ejemplo positivo: < Sol, Templ, Normal, Fuerte, Templ, Igual >
  - Nada que eliminar de GO
  - Generalización minimal de S0: < Sol, Templ, Normal, Fuerte, Templ, Igual > Esta generalización es más específica que la hipótesis de G0
  - Luego: S1 = {< Sol, Templ, Normal, Fuerte, Templ, Igual >} G1 = {<?,?,?,?,?,? >}

#### • Paso 2:

- Ejemplo positivo: < Sol, Templ, Alta, Fuerte, Templ, Igual >
- Nada que eliminar de G1
- Generalización minimal de S1: < Sol, Templ, ?, Fuerte, Templ, Igual > Esta generalización es más específica que la hipótesis de G1
- Luego:

```
S2 = \{ < Sol, Templ, ?, Fuerte, Templ, Igual > \}

G2 = \{ <?, ?, ?, ?, ?, ? > \}
```

# Eliminación de candidatos (ejemplo)

#### • Paso 3:

- Ejemplo negativo:  $\langle Lluvia, Fria, Alta, Fuerte, Templada, Cambio \rangle$
- Nada que eliminar de S2.
- Especializaciones minimales de G2 que son más generales que la hipótesis de S2: < Sol, ?, ?, ?, ?, ? >, <?, Templ, ?, ?, ?, ? > y <?, ?, ?, ?, ?, ? | Iqual >.
- Luego:

```
\label{eq:sol} \begin{array}{ll} \mathtt{S3} &= \{ < Sol, Templ, ?, Fuerte, Templ, Igual > \} \\ \mathtt{G3} &= \{ < Sol, ?, ?, ?, ?, ?, >, < ?, Templ, ?, ?, ?, >, < ?, ?, ?, ?, ?, ?, Igual > \} \\ \end{array}
```

#### • Paso 4:

- Ejemplo positivo: < Sol, Templ, Alta, Fuerte, Fria, Cambio >
- Eliminamos de G3 la hipótesis <?,?,?,?,?,Igual>
- Generalización minimal de S3: < Sol, Templ, ?, Fuerte, ?, ?> Esta generalización es más específica que hipótesis de G3.
- Luego:

```
S4 = \{ \langle Sol, Templ, ?, Fuerte, ?, ? \rangle \}
G4 = \{ \langle Sol, ?, ?, ?, ?, ?, . \rangle, . \langle ?, Templ, ?, ?, ?, ?, . \rangle \}
```

# Algoritmo de eliminación de candidatos (propiedades)

- Sean S y G obtenidos por eliminación de candidatos
  - $\bullet$  Si S y G son no vacíos, resultan ser respectivamente la cota específica y cota general del espacio de versiones (respecto del conjunto de entrenamiento)
  - Si  $S = G = \{h\}$ , entonces h es la única hipótesis de H consistente con todos los ejemplos
  - Si  $S = G = \emptyset$ , no existe  $h \in H$  consistente con los ejemplos
- Convergencia hacia el concepto objetivo, siempre que:

 $\mathbf{C}_{\mathbf{C}}\mathbf{I}_{\mathsf{A}}$ 

- Conjunto de entrenamiento suficientemente grande
- Ejemplos sin errores (ausencia de *ruido*)
- ullet El concepto objetivo está en H

# Elección de ejemplos

- Supongamos posible elegir el siguiente ejemplo
  - Estrategia óptima: requerir un ejemplo satisfecho por la mitad del espacio de versiones
- ullet Ejemplo: < Soleado, Templado, Normal, Ligero, Templado, Igual >
- Convergencia hacia el concepto objetivo (siempre que sea posible)
  - Con  $log_2|VS|$  nuevos ejemplos

## Clasificación de nuevas instancias

- Usamos S y G obtenidos por eliminación de candidatos para clasificar nuevas instancias:
  - Si es consistente con todo S, positivo
  - Si no es consistente con ninguno de G, negativo
  - En otro caso, voto mayoritario o simplemente no se clasifica

## • Ejemplos:

```
 < Sol, Templ, Normal, Fuerte, Fria, Cambio > & -> Si \\ < Lluvia, Fria, Normal, Suave, Templ, Igual > & -> No \\ < Sol, Fria, Normal, Fuerte, Templ, Igual > & -> Si (por mayoría) \\ < Sol, Templ, Normal, Suave, Templ, Igual > & -> ? \\ \end{aligned}
```

 $\mathbf{C}_{\mathbf{C}}\mathbf{I}_{\mathbf{A}}$ 

# Sesgo inductivo

- Siempre que  $H \neq 2^X$ , se tiene un sesgo en el tipo de conceptos que se pueden aprender
- Sesgo inductivo: cualquier medio que el sistema de aprendizaje pueda usar para tener preferencia entre dos hipótesis consistentes con los ejemplos
- Tipos de sesgo inductivo:
  - Sesgo en el lenguaje: el lenguaje disponible para expresar las hipótesis define un espacio de hipótesis que excluye conceptos (por ejemplo, conjunción de restricciones)
  - Sesgo preferencial: el algoritmo de búsqueda en el espacio de hipótesis incorpora implícitamente alguna preferencia de algunas hipótesis sobre otras

 $\mathbf{C}_{\mathbf{C}}\mathbf{I}_{\mathsf{A}}$ 

# Sesgo inductivo y aprendizaje

- Inutilidad del aprendizaje insesgado
  - Espacio de versiones en un espacio de hipótesis insesgado
  - Eliminación de candidatos obtendría  $S = \{(p_1 \lor \ldots \lor p_n)\}, G = \{\neg (n_1 \lor \ldots \lor n_m)\}$
  - Una instancia nueva sería clasificada como positiva por la mitad del espacio de versiones y negativa por la otra mitad
- Un sistema de aprendizaje que no asume conocimiento *a priori* sobre el concepto objetivo no puede clasificar nuevas instancias

 $\mathbf{C}_{\mathbf{C}}\mathbf{I}_{\mathsf{A}}$ 

• Concimiento a priori en eliminación de candidatos: el concepto objetivo está en el espacio de hipótesis

# Bibliografía

- Mitchell, T.M. Machine Learning (McGraw-Hill, 1997)
  - Cap. 1: "Introduction"
  - Cap. 2: "Concept Learning and the General-to-Specific Ordering"
- Russell, S. y Norvig, P. *Inteligencia artificial (Un enfoque moderno)* (Prentice–Hall Hispanoamericana, 1996)
  - Cap. 18: "Aprendiendo de observaciones"