Tema AA–2: Aprendizaje de conceptos

José A. Alonso Jiménez
Miguel A. Gutiérrez Naranjo

Dpto. de Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

Aprendizaje de conceptos

- **Distintas aproximaciones**
- **Un concepto no es más que el conjunto de todas sus instancias (Leibnitz)**
  - **La** *Humanidad* **es el conjunto de todos los** *hombres***.**
  - **La** *sociedad andaluza* **es el conjunto de habitantes de Andalucía.**
  - **La relación** *mayor_que* **definida sobre el conjunto de los números reales es el conjunto de pares** (x, y) **tales que** x **es mayor que** y**.**

Aprendizaje de conceptos

- Para conocer un conjuntos tenemos dos tipos de definiciones:
  - *Extensiva:* Enumerando uno tras otro sus elementos

    $$Vocales = \{a,\ e,\ i,\ o\ ,u\}$$

  - *Intensiva:* Dando una propiedad que tengan todos aquellos y sólo aquellos elementos del conjunto

    $$Pares_{\mathbb{N}} = \{n \in \mathbb{N} | \exists m \in \mathbb{N}(n = 2 \times m)\}$$

Aprendizaje de conceptos

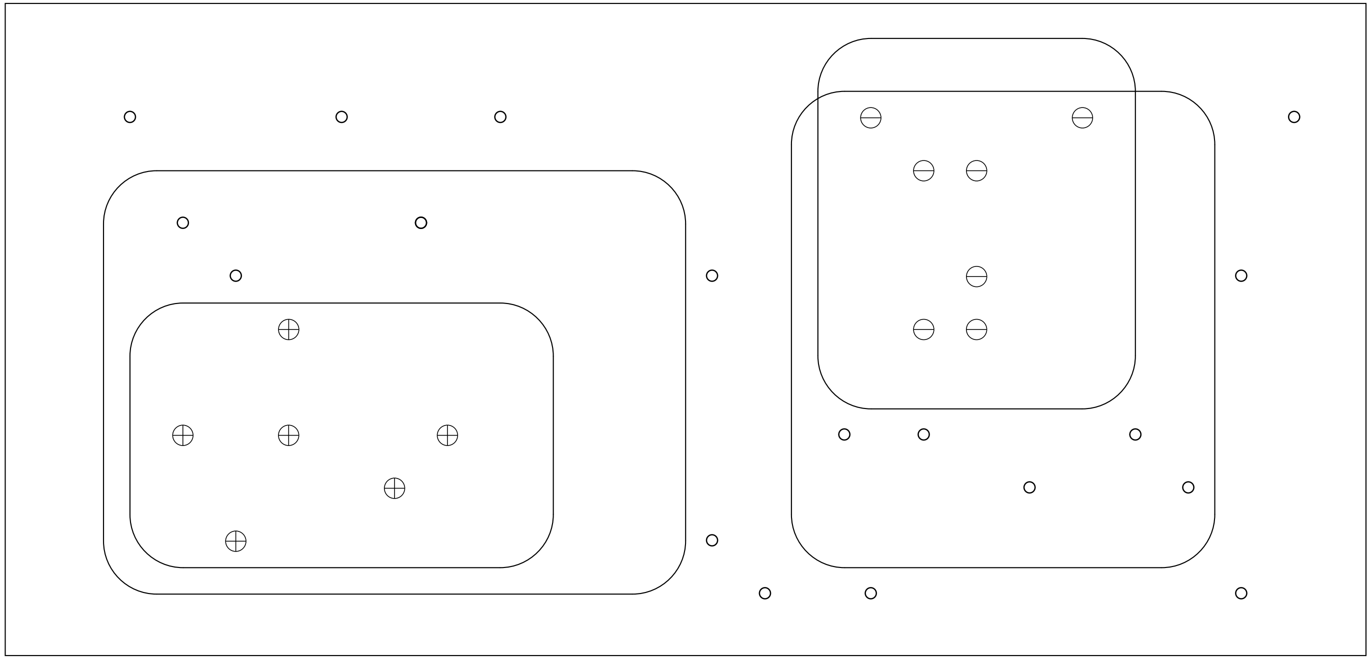
- **El aprendizaje de conceptos estudia cómo conseguir la definición de una categoría a partir de ejemplos positivos y negativos de esa categoría.**
- **El aprendizaje de conceptos estudia cómo inferir automáticamente una función general sobre el conjunto de ejemplos que tome valores booleanos y caracterice los ejemplos conocidos.**

$$f: \quad Ejemplos \longrightarrow \{0, 1\}$$

- **Dos objetivos:**
  - **Compresión de la información**
  - **Capacidad predictiva**

Aprendizaje de conceptos

- **Gráficamente**



Ejemplo

Problema: Un amigo nuestro practica deportes acuáticos. Unos días los practica y otros no. Nos gustaría saber si hoy va a ir a practicarlos o no.

o o o O o o o

- *Universo:* Conjunto de todos los dias posibles
- *Objetivo:* Conjunto de todos los dias que practica deporte
- *Función objetivo:* Función característica del conjunto objetivo

$$Hacer\_deporte : \quad Días \longrightarrow \{0, 1\}$$

El problema de la representación

¿Cómo representamos un *día?*

- Pares atributo–valor
- **Selección de atributos:** *Cielo, Temperatura, Humedad, Viento Agua, Previsión*
- **Selección de valores:**

Cielo: Soleado, lluvioso.
Temperatura: Alta, templada, fría.
Humedad: Alta, normal, baja.
Viento: Fuerte, débil, sin_viento.
Agua: Caliente, templada, fría.
Previsión: Cambio, Sigue_igual.

Ejemplo

Conjunto de entrenamiento:

Cielo	Temperatura	Humedad	Viento	Agua	Previsión	Hacer_Deporte
Soleado	Templada	Normal	Fuerte	Templada	Sigue_igual	Sí
Soleado	Templada	Alta	Fuerte	Templada	Siguel_igual	Sí
Lluvioso	Fría	Alta	Fuerte	Templada	Cambio	No
Soleado	Templada	Alta	Fuerte	Fría	Cambio	Sí

- ¿Cuándo hacemos deporte?
- ¿Podemos definir el concepto *Hacer_Deporte*?

El problema del aprendizaje de conceptos (I) *(Inf.)*

- **Dados:**
  - Un *universo* o conjunto de instancias X *(Días posibles, cada uno descrito por los atributos Cielo, Temperatura, Humedad, Viento, Agua y Previsión).*
  - Una función de clasificación definida sobre X *desconocida:*

$$c : X \longrightarrow Clasif$$

  donde $Clasif$ es el conjunto de posibles clasificaciones que tiene una instancia. En nuestro ejemplo podemos considerar

$$\textit{Clasif}=\{\textit{Sí, No}\} \quad \text{o} \quad Clasif = \{0, 1\}$$

El problema del aprendizaje de conceptos *(Inf.)*

– Un conjunto de entrenamiento *(conocido)*

$$D = \{(x_1, c(x_1)), (x_2, c(x_2)), \ldots, (x_n, c(x_n))\}$$

formado por pares $(x_i, c(x_i))$ donde $x_i \in X$ y $c(x_i)$ es la clasificación de la instancia x_i por la función c.

- **Encontrar:**

– Una *función objetivo*

$$h : X \longrightarrow Clasif$$

tal que si $(x_i, c(x_i)) \in D$ entonces $h(x_i) = c(x_i)$

Hipótesis

La Hipótesis del Aprendizaje Inductivo:
Cualquier hipótesis que aproxime la función objetivo sobre un conjunto suficientemente grande de ejemplos de entrenamiento también aproximará la función objetivo sobre el resto de los ejemplos.

El problema de la representación

Seguimos con nuestro ejemplo

- Muchas representaciones posibles
- Pares valor–atributo
- **Hipótesis:** Conjunción de restricciones sobre las instancias de los atributos
- Cada restricción puede ser:
  - Un valor específico (p. e. $Agua = Templada$)
  - Cualquier valor (p. e. $Agua = ?$)
  - No se permite ningún valor (p. e. $Agua = \emptyset$)

El problema de la representación

Ejemplos

- Podemos considerar que los días para hacer deporte son aquellos en los que

  Cielo = Soleado **y**
  Temperatura = ? **y**
  Humedad = ? **y**
  Viento = Fuerte **y**
  Agua = ? **y**
  Previsión = Sigue_igual

- O con una representación más compacta

  (Soleado,?,?,Fuerte,?,Sigue_igual)

- Que representa el subconjunto de X formado por aquelos días en los que el atributo *Cielo* toma el valor *Soleado*, el *Viento* es *Fuerte* y la *Previsión* es *Sigue_igual*, y los demás atributos pueden tomar cualquier valor.

Aprendizaje inductivo

- **Un sistema formal con un lenguaje L y tres sublenguajes:**
  - L_B *(el lenguaje del conocimiento base)*
  - L_H *(el lenguaje de las hipótesis)*
  - L_E *(el lenguaje de los ejemplos)*
- **El problema se define de la siguiente manera:**
  - **Dados**
    - $B \in L_B$ Conocimiento base
    - $E^{\oplus} \in L_E$ Ejemplos positivos
    - $E^{\ominus} \in L_E$ Ejemplos negativos

    tales que cumplan:

    * *Necesidad a priori:* $\exists e^{\oplus} \in E^{\oplus} \quad T \not\vdash e^{\oplus}$
    * *Consistencia a priori:* $\forall e^{\ominus} \in E^{\ominus} \quad T \not\vdash e^{\ominus}$
  - **Encontrar** un conjunto finito $H \in L_H$ tal que se cumplan
    * *Suficiencia a posteriori:* $\forall e^{\oplus} \in E^{\oplus} \quad T \bigcup H \vdash e^{\oplus}$
    * *Consistencia a posteriori:* $\forall e^{\ominus} \in E^{\ominus} \quad T \bigcup H \not\vdash e^{\ominus}$

Aprendizaje de conceptos

- En nuestro ejemplo no consideramos conocimiento base
- **Dados:**
- L_E es el conjunto de todas las sextuplas formadas por los corrrespondientes valores, esto es, todas las instancias expresables.
- L_H será el conjunto de todas la aplicaciones

$$h: L_E \longrightarrow \{0,1\}$$

cada aplicación se representará como una sextupla del tipo

(Soleado,?,?,Fuerte,?,Sigue_igual)

que represeanta el conjunto de instancias clasificadas positivamente.

Aprendizaje de conceptos

- Un conjunto de entrenamiento

$$E^{\oplus} = \{(x_i, 1\}_{i \in I} \qquad E^{\ominus} = \{(x_i, 0\}_{i \in I}$$

(o de manera resumida $D = \{(x_i, c(x_i)\}_{i \in I}$)

- **Encontrar:**

$$h \in H \text{ tal que } \forall i \in I(h(x_i) = c(x_i))$$

Aprendizaje como búsqueda

- Número de instancias del conjunto $X = 3 \times 2 \times 2 \times 2 \times 2\times = 96$ (*Cielo* puede tomar tres posibles valores, el resto de los atributos dos.
- Número de hipótesis sintácticamente distintas $= 5\times4\times4\times4\times4\times4 = 5120$ (En cada atributo podemos considerar además los valores "?" y"$\emptyset$"
- Número de hipótesis semánticamente distintas $= 1+(4\times3\times3\times3\times3\times3) = 973$ (Basta con que $\emptyset$ aparezca una vez en la hipótesis para que ningún ejemplo sea positivo.)

Estructura de retículo (I)

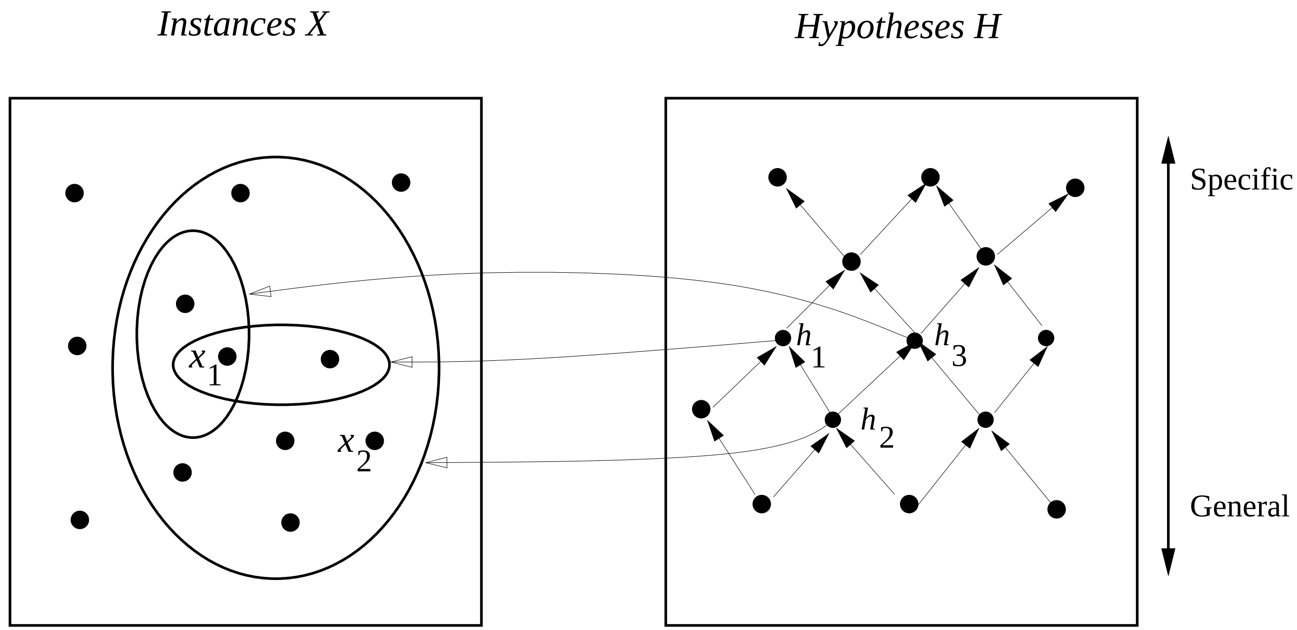
- **Definición 1: Una relación R es un orden parcial sobre un conjunto X si se satisfacen las siguientes condiciones:**
  - $(\forall x \in X)\,[xRx]$ *Reflexiva*
  - $(\forall x, y \in X)\ [(xRy \wedge yRx) \Rightarrow x = y]$ *Antisimétrica*
  - $(\forall x, y, z \in X)\ [(xRy \wedge yRz) \Rightarrow xRz]$ *Transitiva*
- **Definición 2: Sea S un conjunto con un orden parcial R y T un subconjunto de S. Entonces, se dice que $a \in S$ es un *cota superior* de T si para todo $x \in T$ se tiene xRa. Análogamente se dice que $b \in S$ es una *cota inferior* de T si para tod $x \in T$ se tiene que bRx.**

Estructura de retículo (I)

- **Definición 3: Sea S un conjunto con un orden parcial R y T un subconjunto de S. Entonces, se dice que $a \in S$ es el supremo de T, $sup(T)$ si a es una cota superior, y para cualquier otra cota superior a', se tiene que aRa'. De manera análoga, se dice que $b \in S$ es el ínfimo de T, $inf(T)$ si b es una cota inferior, y para cualquier otra cota inferior b', se tiene que $b'Rb$.**

- **Definición 4: Sea S un conjunto con un orden parcial R. Decimos que el par (S, R) es un retículo si para cualquier subconjunto *finito* T de S, existen $inf(T)$ y $sup(T)$**

Instancias, hipótesis y órdenes



x_1 = <*Sunny, Warm, High, Strong, Cool, Same*>
x_2 = <*Sunny, Warm, High, Light, Warm, Same*>

h_1 = <*Sunny, ?, ?, Strong, ?, ?*>
h_2 = <*Sunny, ?, ?, ?, ?, ?*>
h_3 = <*Sunny, ?, ?, ?, Cool, ?*>

Espacios de versiones

Una hipótesis h es **consistente** con el conjunto de entrenamiento D del concepto objetivo c si y sólo si $\phi_h(x) = c(x)$ para cada ejemplo de entrenamiento $(x, c(x))$ de D.

$$Consistente(h, D) \equiv (\forall (x, c(x)) \in D) \phi_h(x) = c(x)$$

donde ϕ_h es la función objetivo asociada a la hipótesis h.
Nota: En lo que sigue, identificamos h con ϕ_h

Espacios de versiones

El **espacio de versiones** $VS_{H,D}$ correspondiente al espacio de hipótesis H y el conjunto de entrenamiento D es el subconjunto de H consistente con el conjunto de entrenamiento D.

$$VS_{H,D} \equiv \{h \in H | Consistente(h, D)\}$$

- **El *espacio de versiones* es el conjunto de *todas* las posibles soluciones al problema de aprendizaje.**

Ejemplo

Cielo	Temperatura	Humedad	Viento	Agua	Previsión	Hacer_Deporte
Soleado	Templada	Normal	Fuerte	Templada	Sigue_igual	Sí
Soleado	Templada	Alta	Fuerte	Templada	Siguel_igual	Sí
Lluvioso	Fría	Alta	Fuerte	Templada	Cambio	No
Soleado	Templada	Alta	Fuerte	Fría	Cambio	Sí

Espacio de versiones:

(Soleado, Templada, ?, Fuerte, ?,?)
(Soleado, Templada, ?, ?, ?,?)
(Soleado, ?, ?, ?, ?,?)
(Soleado, ?, ?, Fuerte, ?,?)
(?, Templada, ?, Fuerte, ?,?)
(?, Templada, ?, ?, ?,?)

El algoritmo Enumera–y–elimina

ALGORITMO PARA ENCONTRAR EL ESPACIO DE VERSIONES DE UN PROBLEMA DE APRENDIZAJE

1. Inializa *Espacio–de–versiones* como una lista conteniendo todas las hipótesis de H.
2. Para cada ejemplo de entrenamiento $(x, c(x))$
   - Elimina de *Espacio–de–versiones* todas las hipótesis h para las que $h(x) \neq c(x)$
3. Salida: *Espacio–de–versiones*

Representación

Sea H un conjunto de hipótesis y $\leq$ el orden de generalidad.

- Se dice que $g \in H$ es un elemento de máxima generalidad de H si

$$\neg\exists g' \in H \, (g < g')$$

- Se dice que $s \in H$ es un elemento de máxima especificidad de H si

$$\neg\exists s' \in H \, (s' < s)$$

Representación

- La **Cota General**, G, de un espacio de hipótesis H respecto de un conjunto de entrenamiento D es el conjunto de los elementos de máxima generalidad de H consistentes con D

$$G = \{g \in H | consistente(g, D) \wedge (\neg \exists g' \in H)[(g' > g) \wedge Consistenete(g', D)]\}$$

- La **Cota Específica**, S, de un espacio de hipótesis H respecto de un conjunto de entrenamiento D es el conjunto de los elementos de máxima especificidad de H consistentes con D

$$S = \{s \in H | consistente(s, D) \wedge (\neg \exists s' \in H)[(s > s') \wedge Consistenete(s', D)]\}$$

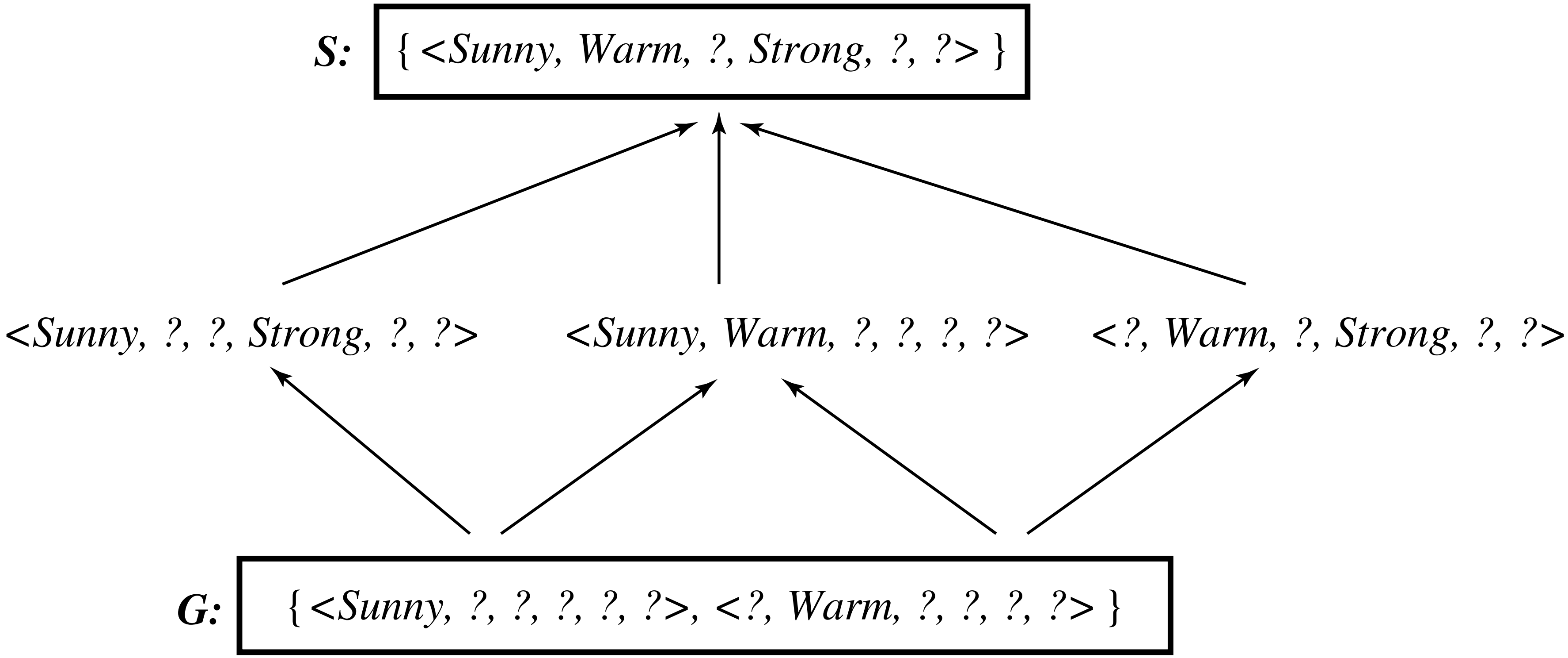
Teorema

Teorema de representación del espacio de versiones:

Sea X un conjunto arbitrario de instancias y sea H un conjunto de hipótesis de valores booleanos definido sobre X. Sea $c : X \leftarrow \{0,1\}$ una función de clasificación definida sobre X y $D = \{(x_i, c(x_i)\}_{i \in I}$ un conjunto de entrenamiento. Sean G y S las cotas general y específica resp. del espacio de hipótesis H respecto de un conjunto de entrenamiento D. Entonces

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

Ejemplo de espacio de versiones



Generalización y especialización

Sea H un espacio de versiones y $h \in H$

- $h' \in H$ es una *generalización minimal* de h verificando la propiedad P si
  - $h' > h$, h' verifica la propiedad P y no existe $h'' \in H$ verificando la propiedad P cumpliendo $h < h'' < h'$
- $h' \in H$ es una *especialización minimal* de h verificando la propiedad P si
  - $h > h'$, h' verifica la propiedad P y no existe $h'' \in H$ verificando la propiedad P cumpliendo $h' < h'' < h$

Algoritmo Eliminación–de–candidatos

- **Inicio:** Dados H y D
  - $G \leftarrow$ El conjunto de elementos de máxima generalidad de H
  - $S \leftarrow$ El conjunto de elementos de máxima especificidad de H
- Para cada ejemplo d del conjunto de entrenamiento D,
  - Si d es un ejemplo positivo
    - ∗ Elimina de G cualquier hipótesis inconsistente con d
    - ∗ Para cada hipótesis s de S que no sea consistente con d
      - · Elimina s de S
      - · Añade a S todas las generalizaciones minimales h de s tales que

Algoritmo Eliminación–de–candidatos

1. h sea consistente con d
2. Algún elemento de G es más general que h

· Elimina de S cualquier hipótesis más general que otra hipótesis de S

– Si d es un ejemplo negativo

∗ Elimina de S cualquier hipótesis inconsistente con d

∗ Para cada hipótesis g de G que no sea consistente con d

· Elimina g de G

· Añade a G todas las especializaciones minimales h de g tales que

1. h sea consistente con d
2. Algún elemento de S es más específico que h

· Elimina de G cualquier hipótesis menos general que otra hipótesis de G

Ejemplo (I)

S_0: {<Ø, Ø, Ø, Ø, Ø, Ø>}

G_0: {<?, ?, ?, ?, ?, ?>}

S_0: {<∅, ∅, ∅, ∅, ∅, ∅>}

↓

S_1: {<*Sunny, Warm, Normal, Strong, Warm, Same*>}

↓

S_2: {<*Sunny, Warm, ?, Strong, Warm, Same*>}

G_0, G_1, G_2: {<*?, ?, ?, ?, ?, ?*>}

Training examples:

1 . <*Sunny, Warm, Normal, Strong, Warm, Same*>, *Enjoy Sport = Yes*

2 . <*Sunny, Warm, High, Strong, Warm, Same*>, *Enjoy Sport = Yes*

Ejemplo (II)

S_2, S_3: { <*Sunny, Warm, ?, Strong, Warm, Same*> }

G_3: {<*Sunny, ?, ?, ?, ?, ?*> <*?, Warm, ?, ?, ?, ?*> <*?, ?, ?, ?, ?, Same*>}

G_2: {<*?, ?, ?, ?, ?, ?*> }

Training Example:

3. <*Rainy, Cold, High, Strong, Warm, Change*>, *EnjoySport=No*

S_3: {<*Sunny, Warm, ?, Strong, Warm, Same*>}

S_4: { <*Sunny, Warm, ?, Strong, ?, ?*> }

G_4: {<*Sunny, ?, ?, ?, ?, ?*> <*?, Warm, ?, ?, ?, ?*>}

G_3: {<*Sunny, ?, ?, ?, ?, ?*> <*?, Warm, ?, ?, ?, ?*> <*?, ?, ?, ?, ?, Same*>}

Training Example:

4.<*Sunny, Warm, High, Strong, Cool, Change*>, *EnjoySport = Yes*

Convergencia del algoritmo

Si el conjunto de entrenamiento es suficientemtente grande, ¿Podemos garantizar que el espacio de versiones contiene como único elemento la función de clasificación? Sí si se verifica:

- No hay errores en la clasificación de los ejemplos de entrenamiento
- La hipótesis correcta está en el espacio de hipótesis

Bibliografía

MITCHELL, T. M. *Machine Learning*

- *Texto:* McGraw–Hill, 1997. Capítulos I y II.
- *Web:* `http://www.cs.cmu.edu/ tom/mlbook.html`