

APRENDIZAJE CONCEPTUAL

Índice

- Definición
- Aprendizaje conceptual como búsqueda
- Algoritmo FIND-S
- Algoritmo CANDIDATE-ELIMINATION
- Sesgo Inductivo

Definición

- **Aprendizaje Conceptual:**

Inferir un concepto —una función booleana— a partir de un conjunto de entrenamiento.

- **Ejemplo:**

¿Cuándo salva Pedro un examen?

- **Se define concepto objetivo c :**

$c: X \rightarrow \{0,1\}$ en donde X es el conjunto de instancias.

- En el ejemplo, $c(x) = 1$ implica que Pedro salva el examen para la instancia $x \in X$

Definición

- Supongamos que contamos con el siguiente conjunto de entrenamiento:

#Ej	x					c(x)
	Dedicación	Dificultad	Horario	Humedad	Humor Doc	
1	Alta	Alta	Nocturno	Media	Bueno	SÍ
2	Baja	Media	Matutino	Alta	Malo	NO
3	Media	Alta	Nocturno	Media	Malo	SÍ
4	Media	Alta	Matutino	Alta	Bueno	NO

- ➔ Alguien ya eligió por mí los atributos para representar las instancias en este problema.

Definición

- Puedo elegir cómo representar al concepto buscado.
- A los posibles candidatos los denominamos *hipótesis*.
- Para fijar ideas utilizaremos un espacio H con hipótesis de la forma:
 $h: \langle \text{Dedicación}, \text{Dificultad}, \text{Horario}, \text{Humedad}, \text{HumorDoc} \rangle$
 - ▶ h representa una conjunción de restricciones sobre cada atributo
 - ▶ Para cada atributo, se tienen los posibles valores:
 - ⊙ Valor específico: alto, medio,...
 - ⊙ Todo valor es aceptable: ?
 - ⊙ Ningún valor es aceptable: \emptyset

Definición

- La hipótesis “*Pedro salva un examen \Leftrightarrow estudia mucho*”, se puede representar por: $h: \langle \text{Alta}, ?, ?, ?, ? \rangle$
 - Hipótesis más general, “*Pedro salva siempre*”: $\langle ?, ?, ?, ?, ? \rangle$
 - Hipótesis más específica, “*Pedro siempre pierde*”: $\langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$, pero también $\langle \text{Alta}, \emptyset, \emptyset, \emptyset, \emptyset \rangle$, $\langle \emptyset, ?, \emptyset, \text{Media}, \text{Bueno} \rangle$, ...
- ➔ *Estos ejemplos son válidos para el espacio H elegido. ¿Qué sucede si h es una disyunción de restricciones en lugar de una conjunción?*

Definición

- Con el conjunto de entrenamiento dado... ¿ $\langle Alta, ?, ?, ? \rangle$ es una buena hipótesis para ser el concepto objetivo?
- ¿Esta es una buena elección para representar a las hipótesis si el concepto objetivo es “*Pedro salva cuando su dedicación es alta o la dificultad es baja*”?

Definición

- Datos:

- ▶ Dominio de instancias X

- ▶ Función objetivo:

$$c: X \rightarrow \{0, 1\}$$

- ▶ Espacio de hipótesis H :

$$H = \{h_0, h_1, \dots\}, h_i: X \rightarrow \{0, 1\}$$

- ▶ Conjunto de Entrenamiento D :

$$D = \{ [x_0, c(x_0)], \dots, [x_n, c(x_n)] \mid x \in X \}$$

- Objetivo:

Determinar una hipótesis $h \in H$ tal que $h(x) = c(x)$ para todo $x \in X$.

Definición

- En nuestro ejemplo:
 - ▶ X es el conjunto de instancias formadas por los atributos:
 - Dedicación: alta, media, baja.
 - Dificultad: alta, media, baja.
 - Horario: matutino, nocturno.
 - Humedad: alta, media, baja.
 - HumorDoc: bueno, malo.
 - ▶ $c: X \rightarrow \{0,1\} / c(x)=1 \Leftrightarrow$ Pedro salva bajo las condiciones x.
 - ▶ $H = \{h / h \text{ es la conjunción de restricciones sobre los atributos, tomando un valor específico, cualquier valor o ninguno}\}$
 - ▶ D está dado por la tabla de la transparencia 4.

Definición

- La única información que tenemos de c son los valores en el conjunto de entrenamiento.
- Nada nos garantiza que el concepto objetivo pertenezca al conjunto H que elegimos.
- Hipótesis de Aprendizaje Inductivo:

Toda hipótesis que aproxime correctamente el concepto objetivo en un conjunto de ejemplos lo suficientemente grande también lo hará sobre las instancias aún no observadas.

Aprendizaje = Búsqueda

- Nuestro problema es encontrar una hipótesis en el espacio elegido
- En nuestro ejemplo, tenemos 2000 $[5*5*4*5*4]$ hipótesis sintácticamente distintas
- ¿Cuántas semánticamente distintas hay?
- Tenemos que desarrollar estrategias para buscar en espacios muy grandes o infinitos.

Aprendizaje = Búsqueda

- Se necesitan estrategias para buscar en el espacio de hipótesis sin tener que listarlas todas.
- Definiciones:

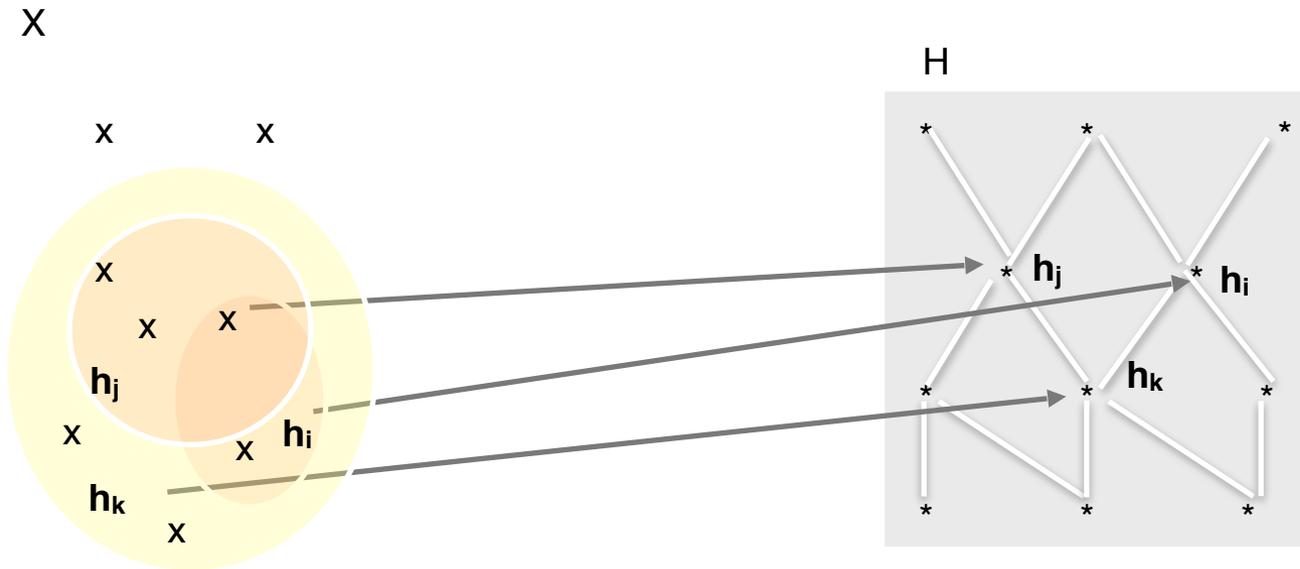
- ▶ Más general o igual: Sean h_j y h_k funciones booleanas sobre X

$$h_j \geq h_k \text{ sii } (\forall x \in X) h_k(x)=1 \rightarrow h_j(x)=1$$

- ▶ Análogamente, definimos más específica o igual:

$$h_j \leq h_k \text{ sii } (\forall x \in X) h_j(x)=1 \rightarrow h_k(x)=1 \text{ o } h_k(x)=0 \rightarrow h_j(x)=0$$

Aprendizaje = Búsqueda



se cumple: $h_k \geq h_i$
 $h_k \geq h_j$

pero no se cumple: $h_j \geq h_i$
 $h_i \geq h_j$

Recordamos el ejemplo

#Ej	x					c(x)
	Dedicación	Dificultad	Horario	Humedad	Humor Doc	
1	Alta	Alta	Nocturno	Media	Bueno	SÍ
2	Baja	Media	Matutino	Alta	Malo	NO
3	Media	Alta	Nocturno	Media	Malo	SÍ
4	Media	Alta	Matutino	Alta	Bueno	NO

Find-S

- ¿Cómo realizar la búsqueda de c en H ?
- Algoritmo FIND-S: empiezo con la hipótesis más específica y a medida que tengo ejemplos generalizo.
- En nuestro ejemplo:

$\langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$

+ $[x_1, \text{sí}]$

$\langle \text{Alta, Alta, Nocturno, Media, Bueno} \rangle$

+ $[x_3, \text{sí}]$

$\langle ?, \text{Alta, Nocturno, Media, ?} \rangle$

➔ *Esta forma de generalizar es válida para el H del ejemplo, pero no necesariamente para otro.*

Find-S

- Algoritmo FIND-S:

$h \equiv$ hipótesis más específica de H

Para cada instancia positiva x

Para cada restricción r de h

Si x no satisface r , sustituir r por una restricción más general que satisfaga x .

Devolver h

- ¿Qué sucede con los ejemplos negativos?
- Notar que en el H de nuestro ejemplo siempre tenemos una única hipótesis más específica, pero en principio podrían ser varias.

Find-S

- Llegamos a una hipótesis válida, pero...
 - ▶ ¿Llegamos al concepto correcto?
 - ▶ ¿Por qué preferir la hipótesis más específica?
 - ▶ ¿Qué pasa si hay varias hipótesis más específicas? ¿Por qué quedarnos con sólo una de ellas?
 - ▶ ¿Y si en el conjunto de entrenamiento hay ejemplos mal clasificados?

Candidate-Elimination

- Definiciones:

- ▶ Una hipótesis h es consistente con un conjunto de entrenamiento:

$$\text{Consistente}(h, D) \equiv \forall [x, c(x)] \in D, h(x)=c(x).$$

- ▶ Espacio de versiones $VS_{H,D}$

$$VS_{H,D} = \{h \in H / \text{consistente}(h,D)\}.$$

- El espacio de versiones representa a todas las hipótesis candidatas a ser el objetivo buscado, dado el conjunto de entrenamiento.
- ¿Cómo calcular el espacio de versiones?

Candidate-Elimination

- Algoritmo List-Then-Eliminate:

VS \equiv conjunto con TODAS las hipótesis.

Para cada ejemplo $[x, c(x)]$:

 Eliminar todas las h tq. $h(x) \neq c(x)$.

Devolver VS.

- ▶ Enumera todo el espacio de hipótesis.
- ▶ En espacios H infinitos...

Candidate-Elimination

- Representemos el espacio de versiones con las hipótesis consistentes más específicas y más generales.

- Límite general $G_{H,D}$:

$$G_{H,D} \equiv \{g \in H / \text{consistente}(g,D) \text{ y } (\neg \exists g' \in H) g' > g \wedge \text{consistente}(g',D)\}$$

- Límite específico $S_{H,D}$:

$$S_{H,D} \equiv \{s \in H / \text{consistente}(s,D) \text{ y } (\neg \exists s' \in H) s > s' \wedge \text{consistente}(s',D)\}$$

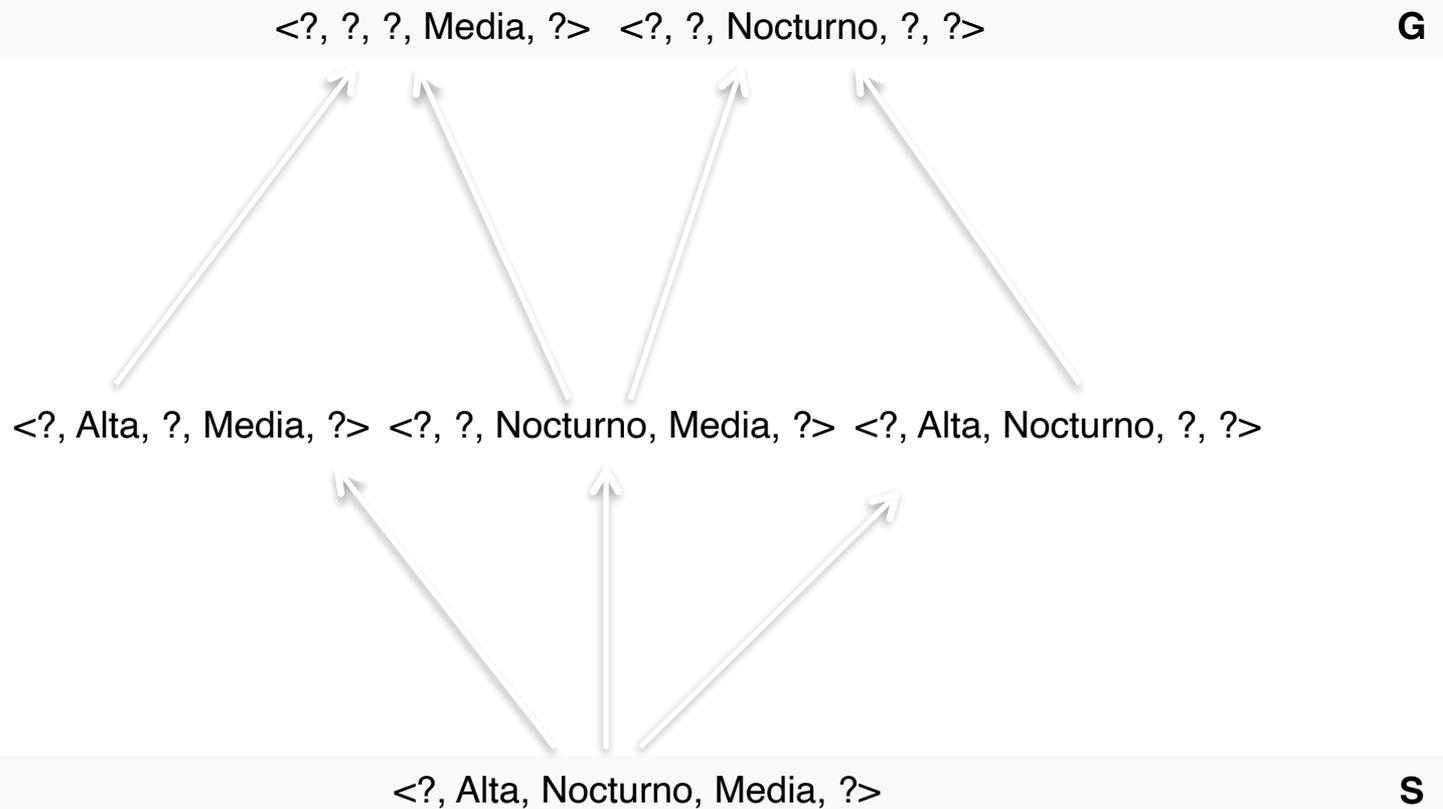
- Teorema de la representación del espacio de versiones

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

- ▶ En otras palabras, con $G_{H,D}$ y $S_{H,D}$ puedo representar a todo $VS_{H,D}$.

Candidate-Elimination

- En nuestro ejemplo:



Candidate-Elimination

- Algoritmo Candidate-Elimination:

S \equiv conjunto de hipótesis más específicas.

G \equiv conjunto de hipótesis más generales.

Para cada instancia x .

Si x es un ejemplo positivo.

Remover de G cualquier hipótesis inconsistente con x .

Para cada hipótesis s de S , inconsistente con x .

Cambiarla por todas las generalizaciones mínimas de s , consistentes con x , para las cuales haya una hipótesis en G más general que ellas.

Remover de S toda hipótesis más general que otra hipótesis de S .

Candidate-Elimination

- Algoritmo Candidate-Elimination (cont):

...

Si x es un ejemplo negativo.

Remover de S cualquier hipótesis inconsistente con x .

Para cada hipótesis g de G , inconsistente con x .

Cambiarla por todas las especializaciones mínimas de g , consistentes con x , para las cuales haya una hipótesis en S más específica que ellas.

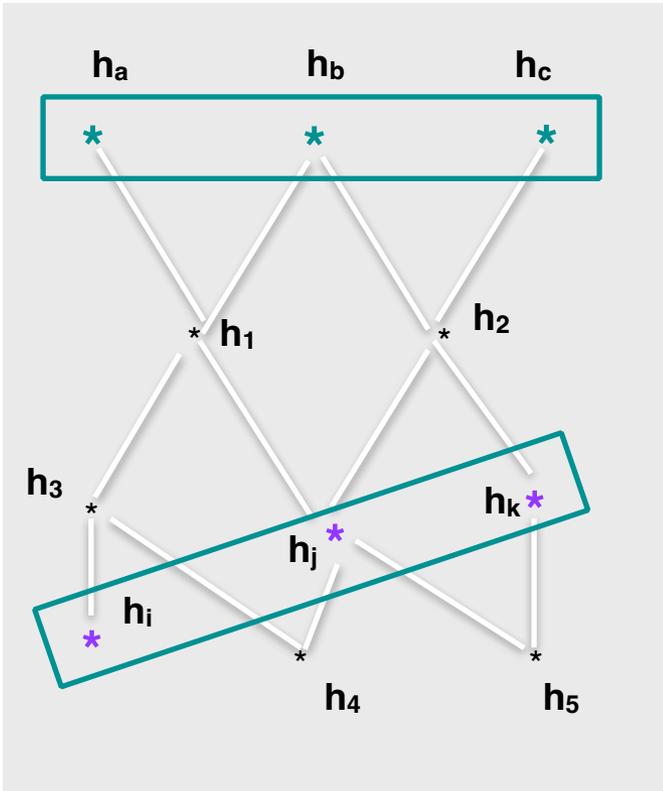
Remover de G toda hipótesis más específica que otra hipótesis de G .

Candidate-Elimination

- Observar que:
 - ▶ El cálculo de S es una generalización de FIND-S.
 - ▶ Las operaciones de generalización y especialización dependen del espacio de trabajo H elegido.

Ejemplo Candidate-Elimination

H



Supongamos

$$G_{H,D} = \{h_a, h_b, h_c\}$$

$$S_{H,D} = \{h_i, h_j, h_k\}$$

Llega un ejemplo x , tal que:

- $c(x) = \text{False}$
- $h_j(x) = h_a(x) = h_b(x) = \text{True}$
- $h_i(x) = h_k(x) = h_c(x) = \text{False}$
- $h_1(x) = h_2(x) = \text{False}$
- $h_3(x) = ?$

1. ¿Qué hipótesis forman $VS_{H,D}$?
2. ¿Cómo queda $VS_{H,D} \cup \{ \langle x, \text{False} \rangle \}$?

Volviendo a Pedro...

#Ej	x					c(x)
	Dedicación	Dificultad	Horario	Humedad	Humor Doc	
1	Alta	Alta	Nocturno	Media	Bueno	SÍ
2	Baja	Media	Matutino	Alta	Malo	NO
3	Media	Alta	Nocturno	Media	Malo	SÍ
4	Media	Alta	Matutino	Alta	Bueno	NO

Candidate-Elimination

- En nuestro ejemplo:

$S_0 = \{\langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle\}$

$G_0 = \{\langle ?, ?, ?, ?, ? \rangle\}$

$S_1 = \{\langle \text{Alta}, \text{Alta}, \text{Nocturno}, \text{Media}, \text{Bueno} \rangle\}$

$G_1 = \{\langle ?, ?, ?, ?, ? \rangle\}$

$S_2 = \{\langle \text{Alta}, \text{Alta}, \text{Nocturno}, \text{Media}, \text{Bueno} \rangle\}$

$G_2 = \{\langle \text{Alta}, ?, ?, ?, ? \rangle, \langle ?, \text{Alta}, ?, ?, ? \rangle, \langle ?, ?, \text{Nocturno}, ?, ? \rangle, \langle ?, ?, ?, \text{Media}, ? \rangle, \langle ?, ?, ?, ?, \text{Bueno} \rangle\}$

$S_3 = \{\langle ?, \text{Alta}, \text{Nocturno}, \text{Media}, ? \rangle\}$

$G_3 = \{\langle ?, \text{Alta}, ?, ?, ? \rangle, \langle ?, ?, \text{Nocturno}, ?, ? \rangle, \langle ?, ?, ?, \text{Media}, ? \rangle\}$

$S_4 = \{\langle ?, \text{Alta}, \text{Nocturno}, \text{Media}, ? \rangle\}$

$G_4 = \{\langle ?, ?, \text{Nocturno}, ?, ? \rangle, \langle ?, ?, ?, \text{Media}, ? \rangle\}$

- ¿qué pasa si procesamos los ejemplos en otro orden?

Candidate-Elimination

- ¿El algoritmo converge a una hipótesis correcta? La respuesta es afirmativa cuando:
 - ▶ Los datos de entrenamiento no contienen ruido (errores).
 - ▶ Hay una hipótesis en H que describe el concepto objetivo.
- Si tuviésemos un oráculo, ¿cuál ejemplo nos conviene elegir como siguiente a procesar?



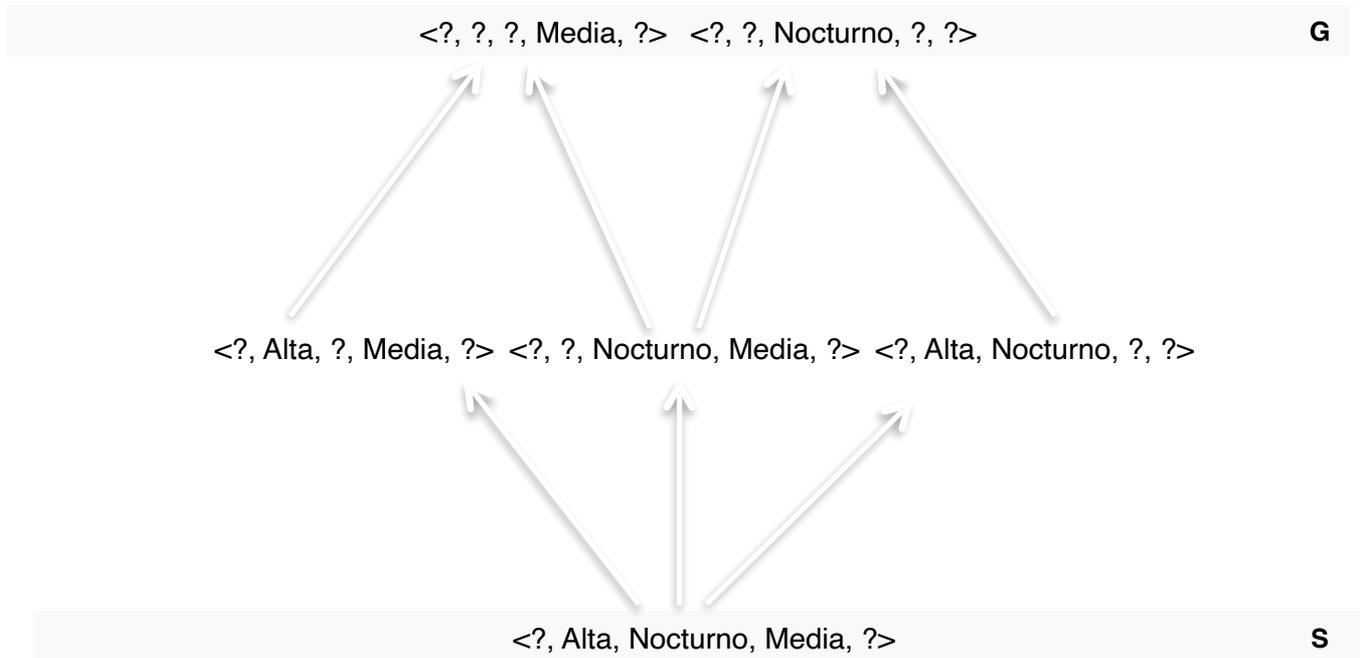
Candidate-Elimination

- ¿Cuál ejemplo elegiríamos para procesar?
 - ▶ El mejor ejemplo es aquel que hace que la mitad del espacio diga 'sí', y la otra mitad diga 'no'.
 - ▶ Esto reduce el tamaño del espacio de versiones a la mitad.

Candidate-Elimination

- Podemos utilizar conocimientos parciales:
 - ▶ Si una instancia satisface a toda hipótesis de S , con seguridad debe ser clasificada positiva.
 - ▶ Si una instancia no satisface a ninguna de las hipótesis de G , con seguridad debe ser clasificada negativa.

Candidate-Elimination



- Por ejemplo, ¿cómo se clasifica a...?
 - [Alta, Alta, Nocturno, Media, Malo]
 - [Alta, Baja, Matutino, Alta, Bueno]
 - [Alta, Baja, Nocturno, Media, Bueno]

Sesgo Inductivo

- ¿Qué sucede si el concepto objetivo no está considerado en H?
- Por ejemplo, este concepto no está en el H del ejemplo:

$\langle \text{Alta}, ?, ?, ?, ? \rangle \vee \langle ?, \text{Baja}, ?, ?, ? \rangle$

- Podríamos considerar el espacio con TODOS los conceptos posibles, pero... ¿cómo quedaría $V_{S_{H,D}}$?
- Si no se hacen suposiciones previas sobre la forma del concepto que se busca.... ¡nada se puede aprender!

Sesgo Inductivo

- Sesgo Inductivo del algoritmo L es el conjunto mínimo de suposiciones B tales que:

$$(\forall x \in X) [(B \wedge D \wedge x) \rightarrow L \text{ clasifica correctamente a } x]$$

- ¿Cuál es el sesgo inductivo de Find-S y Candidate–Elimination?