

# Aprendizaje Automático

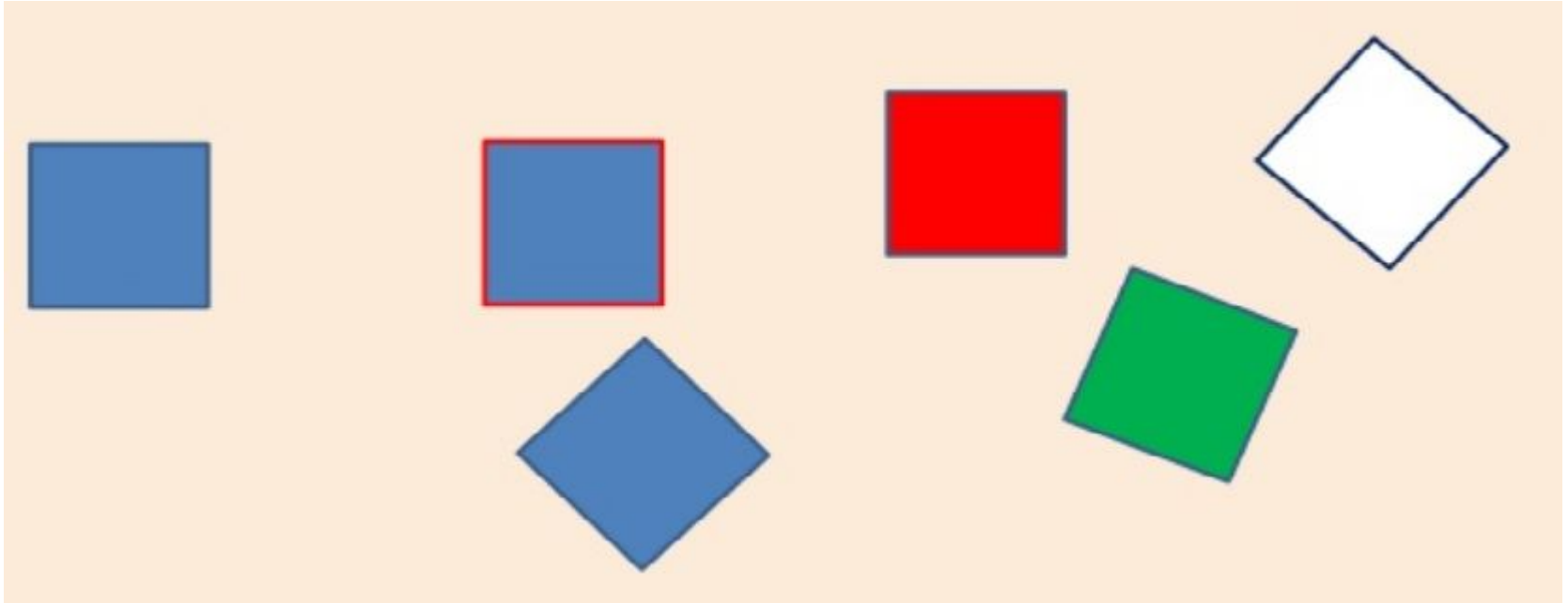
## Aprendizaje de conceptos

Viviana Cotik

1er cuatrimestre 2019

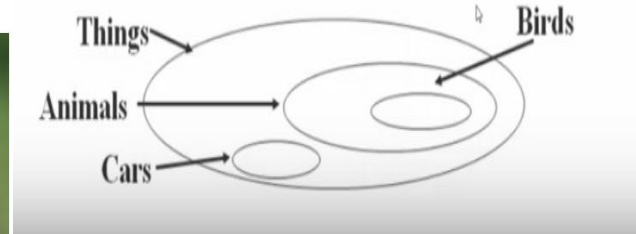


# Concepto



# Concepto

Subconjunto de objetos o eventos definidos sobre un conjunto mayor (aves/animales). ¿Cuáles de estos animales son aves?



# Aprendizaje de conceptos (concept learning)

**Concepto:** un subconjunto de objetos o eventos definidos sobre un conjunto mayor.

Más formalmente, **función booleana** definida sobre el conjunto mayor.

**Ej: EsAve: Animal  $\rightarrow$  Bool**

**Aprender un concepto:** inducir automáticamente una función booleana a partir de cjtos. de ejemplos o datos (clasificados como positivos o negativos). Dado un nuevo caso, devuelve su clase.

# Índice

- Concepto
- **Aprendizaje de conceptos**
  - Función objetivo; instancias; clases; atributos.
  - Hipótesis/modelo; espacio de hipótesis.
  - Ejemplo
- Aprendizaje inductivo
- Adquisición de conceptos como búsqueda
- Espacio de búsqueda y relación de orden General-Específico.
- Algoritmos
  - FIND-S
  - List-then-eliminate
  - Eliminación de candidatos
- Sesgo inductivo

# Aprendizaje de conceptos

Inducir automáticamente una función booleana a partir de ejemplos.

Se lo puede definir como un problema de búsqueda de la hipótesis que más se adecua a los ejemplos mostrados sobre un espacio predefinido de posibles hipótesis.

Construimos y evaluamos hipótesis para aproximar el concepto objetivo.

- $h = \text{Vuela}$
- $h' = \text{DosPatas}$
- $h'' = \text{DosPatas} \wedge \text{TienePlumas} \wedge \text{PoneHuevos} \wedge \text{TienePico}$
- ...

# Aprendizaje de conceptos

- Las hipótesis pertenecen a un **espacio de hipótesis H**
- Puede ocurrir que H no contenga al concepto objetivo.

**Algoritmo de aprendizaje:** buscar la hipótesis en H que mejor se ajuste a nuestros datos (D)

$$\operatorname{argmax}_{h \in H} P(h \mid D)$$

# Aprendizaje de conceptos - Notación

Aprender una función booleana a partir de ejemplos o datos (clasificados como positivos o negativos).

Conjunto de **instancias** o **casos**:  $X$

**Concepto objetivo**:  $c : X \rightarrow \{+, -\}$

**Experiencia**:  $E = \{(x, c(x))\}, x \in X$

Conjunto de **datos**:  $D \subset E$

Conjunto de **hipótesis posibles**:  $H$

**Hipótesis**:  $h \in H / h : X \rightarrow \{+, -\}$

**Objetivo**: hallar  $h / h(x)=c(x)$



# Ejemplo

## Ejemplos:

ATTRIBUTES						CONCEPT
Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

## 6 atributos

Sky = {Sunny, Cloudy, Rainy}

AirTemp = {Warm, Cold}

Humidity = {Normal, High}

Wind = {Strong, Weak}

Water = {Warm, Cool}

Forecast = {Same, Change}

**Posibles datos:**  $3 \times 2^5 = 96$

**Tarea:** predecir valor de Enjoy Sport para un día arbitrario basado en los valores de los atributos

# Aprendizaje de conceptos

**Asumimos que hipótesis consisten en conjunción de valores de los atributos.**

Cada hipótesis, vector de 6 dimensiones

Para cada atributo, la hipótesis contiene:

- valor en el rango del atributo
- ?: cualquier valor es válido
- $\emptyset$ : ningún valor es válido

Pedro disfruta su deporte favorito los días fríos con alta humedad se representa así:

<?, Cold, High, ?, ?, ?>

Hipótesis más general: <?, ?, ?, ?, ?, ?> (todos los días son ej. positivos)

Hipótesis más específica: < $\emptyset$ ,  $\emptyset$ ,  $\emptyset$ ,  $\emptyset$ ,  $\emptyset$ ,  $\emptyset$ > (ningún día es un ej. positivo)

# Aprendizaje de conceptos

**Tarea:** aprender días en que EnjoySport = yes.

**Forma:** escribir el conjunto como **conjunción** de restricciones a partir de los atributos de la instancia

# Índice

- Concepto
- Aprendizaje de conceptos
- **Aprendizaje inductivo**
- Adquisición de conceptos como búsqueda.
- Algoritmos
- Sesgo inductivo

# Inferencia lógica

**Inferir:** establecer relación entre premisas y conclusiones

## Razonamiento deductivo

-Todos los hombres son mortales  
-Sócrates es hombre.  
Por lo tanto, Sócrates es mortal

$p \Rightarrow q$

$p$

-----

$q$

## Razonamiento inductivo

-Sócrates es un hombre y es mortal.  
-Platón es un hombre y es mortal.  
-Aristóteles es un hombre y es mortal.  
Por lo tanto, probablemente todos los  
hombres sean mortales

$p \Rightarrow q$

$r \Rightarrow q$

-----

todo  $s$  en dominio  $\Rightarrow q$

# Aprendizaje inductivo

Consiste en construir un modelo general a partir de información específica.

**Principio de Aprendizaje Inductivo:** Cualquier hipótesis (modelo) que aproxime bien a una **función objetivo** sobre un conjunto suficientemente grande de **datos** también aproximará bien a la **función objetivo** sobre **datos no observados**.

# Sesgo inductivo (Bias)

**Cantidad de posibles ejemplos:**  $3 \cdot 2^5 = 96$

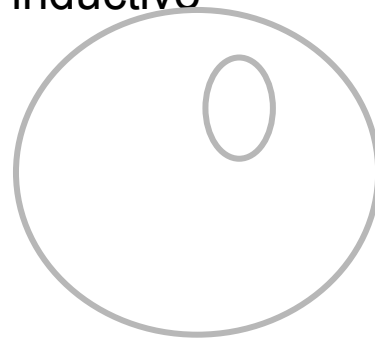
**Cantidad posible de hipótesis:**  $4 \cdot 3^5 + 1 = 973$ .

**Cantidad posible de conceptos :**  $2^{96}$ . gigante.

Hacemos una **suposición** para facilitar el algoritmo de ML: reducir el espacio de búsqueda. Lo hacemos asumiendo que las hipótesis tienen una forma en particular.

Reducción de espacio de conceptos grande a uno chico: sesgo inductivo  
(inductive bias)

Todos los algoritmos de aprendizaje tienen sesgo inductivo.



# Aprendizaje supervisado

Dada una función objetivo  $f$  desconocida, queremos aproximarla mediante una hipótesis  $h$  (o modelo).

Los algoritmos de aprendizaje automático son procedimientos para entrenar modelos a partir de un conjunto de datos.



# Índice

- Concepto
- Aprendizaje de conceptos
- Aprendizaje inductivo
- **Adquisición de conceptos como búsqueda.**
  - **Espacio de búsqueda y relación de orden General-Específico.**
- Algoritmos
- Sesgo inductivo

# Relación de orden General-Específico

**Relación de orden de hipótesis general a específico.** Estructura que existe para todos los problemas de aprendizaje de conceptos.

Se **aprovecha para diseñar algoritmos de aprendizaje** que buscan exhaustivamente aún en espacios de hipótesis infinitos sin necesidad de enumerar cada hipótesis.

Ej. En

$h1 = (\text{Sunny}, ?, ?, \text{Strong}, ?, ?)$

$h2 = (\text{Sunny}, ?, ?, ?, ?, ?)$

**h2 tiene menos restricciones (es más general)**, entonces clasifica más instancias como positivas. Todas las instancias clasificadas como positivas por h1, también lo serán por h2.

# Relación de orden General-Específico

**Definition:** Let  $h_j$  and  $h_k$  be boolean-valued functions defined over  $X$ . Then  $h_j$  is **more\_general\_than\_or\_equal\_to**  $h_k$  (written  $h_j \geq_g h_k$ ) if and only if

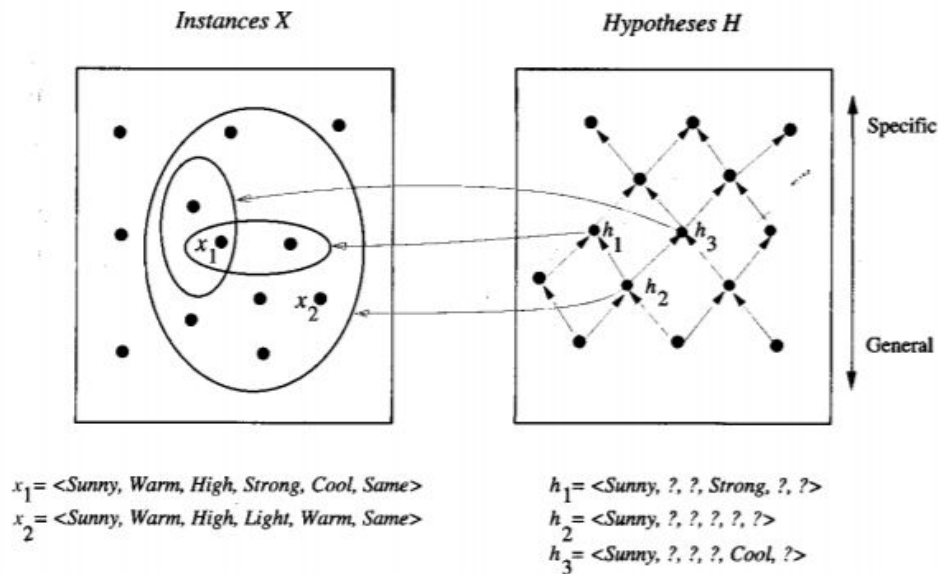
$$(\forall x \in X)[(h_k(x) = 1) \rightarrow (h_j(x) = 1)]$$

También una hipótesis puede ser more\_general\_than (estricto)  $h_j >_g h_k$

sii  $(h_j \geq_g h_k) \wedge (h_k \not\geq_g h_j)$ .

y more\_specific\_than

# Relación de orden General-Específico



**Relación  $\geq_g$ . Ejs.**

- $h_2 \geq_g h_1$  y  $h_2 \geq_g h_3$
- ¿ $h_3 \geq_g h_1$ ? ¿ $h_1 \geq_g h_3$ ?

**$\geq_g$  relación de orden parcial sobre H**

(reflexiva, antisimétrica y transitiva)

Puede haber pares de hipótesis  
 (ej.  $h_1, h_3$ ) tal que  $h_3 \geq_g h_1$  y  
 $h_1 \geq_g h_3$

# Índice

- Concepto
- Aprendizaje de conceptos
- Aprendizaje inductivo
- Adquisición de conceptos como búsqueda
- **Algoritmos**
  - **FIND-S**
  - List-then-eliminate
  - Eliminación de candidatos
- Sesgo inductivo

# Algortimos

**Tarea:** predecir si se disfrutarán deportes

**Medida de performance:** exactitud (accuracy) en la respuesta

**Experiencia:** ejemplos de días y sus resultado

**Conocimiento a adquirir:** Aprender función que mapea días con {+, -} (se practican deportes o no)

**Representación de conocimiento objetivo:** conjunción de valores de atributos

**Mecanismos de aprendizaje:** Find-S / List then Eliminate / Eliminación de candidatos

# Algoritmo Find - S

Algoritmo para buscar en el espacio de conjunciones:

- comienza con hipótesis más específica
- si falla en cobertura de ejemplo positivo, generaliza

## Algoritmo:

1. Inicializar  $h$  a la hipótesis más específica de  $H$
2. Para cada ejemplo de entrenamiento positivo  $x$  de  $X$   
    Para cada atributo  $a_i$  de  $h$   
        Si  $x$  y  $h$  coinciden en  $a_i$ : no hacer nada  
        sino reemplazar  $a_i$  por la siguiente restricción más general  
        satisfecha  
        por  $x$
3. Devolver hipótesis  $h$

# Algoritmo Find-S - Ejecución

Datos: anteriores

$h = \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$

## Paso 1:

$e1 = \langle \text{Sunny, Warm, Normal, Strong, Warm, Same} \rangle$  primer ejemplo positivo.

Ninguna restricción  $\emptyset$  es satisfecha, entonces generalizamos  $h$  lo menos posible para que lo cubra (lo clasifique correctamente como positivo)

$h = \langle \text{Sunny, Warm, Normal, Strong, Warm, Same} \rangle$ . Es muy específico sólo ese ejemplo es positivo.

## Paso 2:

$e2 = \langle \text{Sunny, Warm, High, Strong, Warm, Same} \rangle (+)$

$h = \langle \text{Sunny, Warm, ?, Strong, Warm, Same} \rangle$



# Algoritmo Find-S - Ejecución

## Paso 3:

e3 = <Rainy, Cold, High, Strong, Warm, Change> (negativo, sin cambios)

h no cambia

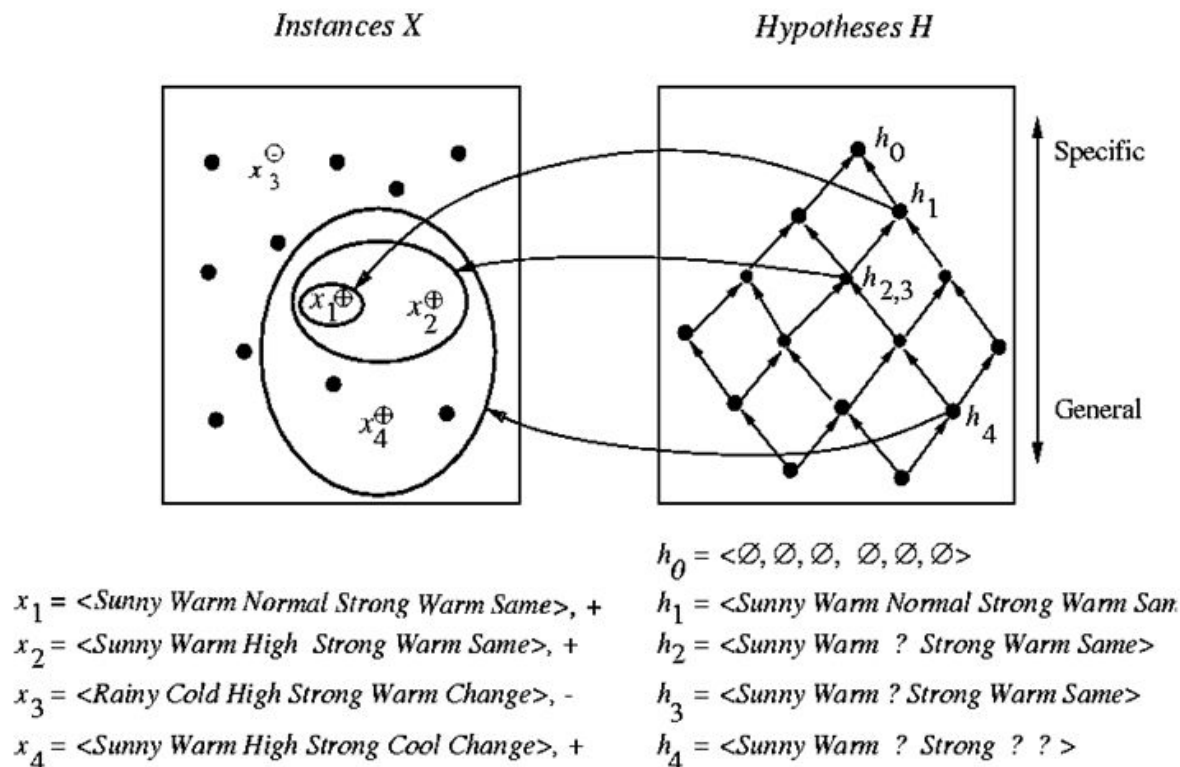
## Paso 4:

e4 = <Sunny, Warm, High, Strong, Cool, Change>

h = <Sunny, Warm, ?, Strong, ?, ?>

h: Si está soleado, caluroso y hay fuerte viento, practicamos el deporte

# Algoritmo Find-S - Ejecución



# Algoritmo Find-S

- Usa orden parcial general-específico para moverse por grilla buscando siguiente hipótesis más específica.

Garantiza:

- devolución **hipótesis más específica en H** consistente con ejemplos positivos
- (concepto objetivo está en  $H \wedge$  ejemplos de entrenamiento correctos)  $\rightarrow$  hipótesis consistente con valores negativos

# Find-S

## **¿Se convergió al concepto objetivo correcto?**

Find-S encuentra **hipótesis consistentes con datos de entrenamiento**. Pero **puede haber otras hipótesis en H consistentes con los datos** (el concepto objetivo). No sabemos si convergió, ni cuán cerca está del concepto objetivo

## **Prefiere hipótesis más específica**

Podría ser mejor la más general o alguna intermedia.

## **No es robusto a ruido**

Los ejemplos de entrenamiento pueden ser inconsistentes. Pueden hacer fallar a FIND-S, ya que ignora ejemplos negativos.

# Índice

- Concepto
- Aprendizaje de conceptos
- Aprendizaje inductivo
- Adquisición de conceptos como búsqueda.
  - Espacio de búsqueda y relación de orden General-Específico.
- **Algoritmos**
  - FIND-S
  - **List-then-eliminate**
  - Eliminación de candidatos

# Definiciones: consistencia y espacio de versiones

**Definition:** A hypothesis  $h$  is **consistent** with a set of training examples  $D$  if and only if  $h(x) = c(x)$  for each example  $\langle x, c(x) \rangle$  in  $D$ .

$$\text{Consistent}(h, D) \equiv (\forall \langle x, c(x) \rangle \in D) h(x) = c(x)$$

**Espacio de versiones (VS):** subconjunto de hipótesis de  $H$  consistentes con el conjunto de entrenamiento,  $H$ : espacio de hipótesis,  $D$ : ejemplos

$$VS_{H,D} \equiv \{h \in H \mid \text{Consistent}(h, D)\}$$

# Algoritmo List-Then-Eliminate

## Algoritmo:

1. Espacio de versiones (**VS**) = todas las hipótesis en **H**
2. Para cada ejemplo de entrenamiento  $(x, c(x))$   
eliminar de VS todas las hipótesis tal que  $h(x) \neq c(x)$

Salida: espacio de versiones **VS**

**Inmanejable el tamaño de H!**

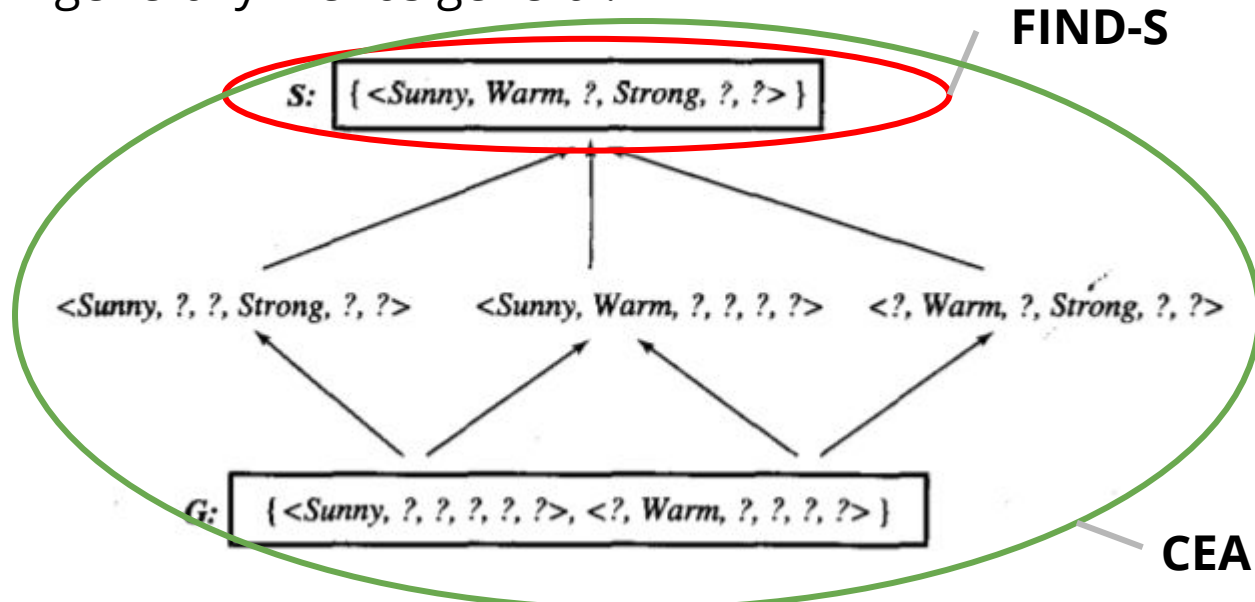
# Índice

- Concepto
- Aprendizaje de conceptos
- Aprendizaje inductivo
- Adquisición de conceptos como búsqueda
  - Espacio de búsqueda y relación de orden General-Específico.
- **Algoritmos**
  - FIND-S
  - List-then-eliminate
  - **Eliminación de candidatos**



# Algoritmo eliminación de candidatos (CEA)

Mismo principio que List-Then-Eliminate, pero con **representación más compacta del espacio de versiones (VS)**. Representado por hipótesis más general y menos general.



**VS con versiones más específicas y más generales** para nuestro ejemplo

Incluye las 6 hipótesis posibles, pero se puede representar sólo con S and G.

# Algoritmo eliminación de candidatos: definiciones

**Definition:** The **general boundary**  $G$ , with respect to hypothesis space  $H$  and training data  $D$ , is the set of maximally general members of  $H$  consistent with  $D$ .

$$G \equiv \{g \in H \mid \text{Consistent}(g, D) \wedge (\neg \exists g' \in H)[(g' >_g g) \wedge \text{Consistent}(g', D)]\}$$

**Definition:** The **specific boundary**  $S$ , with respect to hypothesis space  $H$  and training data  $D$ , is the set of minimally general (i.e., maximally specific) members of  $H$  consistent with  $D$ .

$$S \equiv \{s \in H \mid \text{Consistent}(s, D) \wedge (\neg \exists s' \in H)[(s >_g s') \wedge \text{Consistent}(s', D)]\}$$

# Algoritmo eliminación de candidatos (CEA)

El CEA mantiene dos listas de hipótesis consistentes con el conjunto de entrenamiento:

S: más específicas

G: más generales

Con esto alcanza para derivar todo el espacio de versiones  
VS

# Algoritmo eliminación de candidatos

Algoritmo:

1.  $G$  = al conjunto de hipótesis más general posible en  $H$
2.  $S$  = conjunto de hipótesis más específicas posibles en  $H$
3. Para cada ejemplo de entrenamiento  $d$ , hacer:

**Si es un ejemplo positivo:**

Eliminar de  $G$  las hipótesis inconsistentes con  $d$

Para cada hipótesis  $s$  en  $S$  inconsistente con  $d$

Eliminar  $s$  de  $S$

Agregar a  $S$  las generalizaciones mínimas  $h$  de  $s$ , tal que

$h$  es consistente con  $d$  y algún miembro de  $G$  es más general que  $h$

Eliminar de  $S$  todas las hipótesis que sean más generales que otras hipótesis en  $S$

Si es un ejemplo negativo: ....

# Algoritmo eliminación de candidatos (cont.)

**Si es un ejemplo negativo:**

- Eliminar de **S** las hipótesis inconsistentes con **d**

- Para cada hipótesis **g** en **G** inconsistente con **d**

  - Eliminar **g** de **G**

  - Agregar a **G** las especializaciones mínimas de **h** de **g**, tal que

    - h** es consistente con **d** y algún miembro de **S** es más específico que **h**

  - Eliminar de **G** todas las hipótesis que sean más generales que otras hipótesis en **G**

Devuelve **G**, **S**

# Algoritmo eliminación de candidatos - Ejecución

Datos: anteriores

$G_0 = \{<?, ?, ?, ?, ?, ?>\}$

$S_0 = \{<\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset>\}$

## Paso 1.

$e_1 = <\text{Sunny, Warm, Normal, Strong, Warm, Same}>$ , 1er ej. positivo

$G_1 = G_0$ .

$S_1 = \{<\text{Sunny, Warm, Normal, Strong, Warm, Same}>\}$

## Paso 2.

$e_2 = <\text{Sunny, Warm, High, Strong, Warm, Same}>$ , 2do ej. positivo

$G_2 = G_1 = G_0$

$S_2 = \{<\text{Sunny, Warm, ?, Strong, Warm, Same}>\}$

# Algoritmo eliminación de candidatos - Ejecución

## Paso 3

e3 <Rainy, Cold, High, Strong, Warm, Change> , 3er ej. negativo

G3= {<Sunny,?;?;?;?;?> <?,Warm;?;?;?;?>, <?;?;?;?;?,Same>} (G2 asume que el ejemplo es positivo)

S3 = {<Sunny, Warm, ?, Strong, Warm, Same>}

## Paso 4

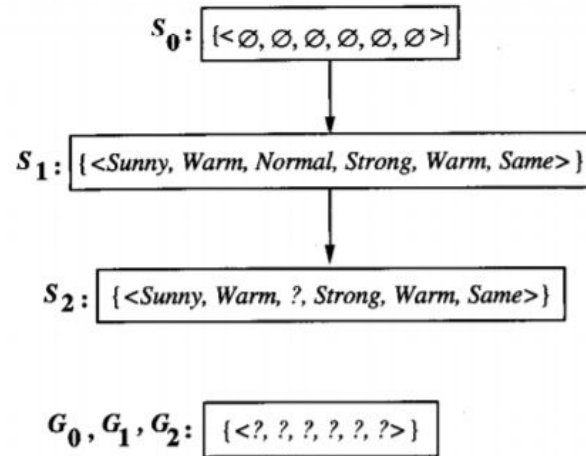
e4 = <Sunny, Warm, High, Strong, Cool, Change>

G4 = {<Sunny,?;?;?;?;?> <?,Warm;?;?;?;?>} (eliminamos una hipótesis porque no cubre el nuevo evento)

S4= <Sunny, Warm, ?, Strong, ?, ?>

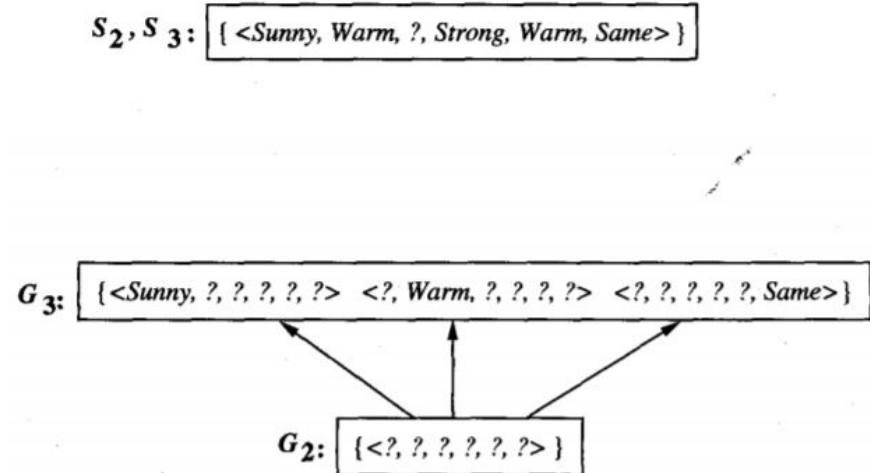
# Algoritmo eliminación de candidatos - Ejecución

Gráficamente:



Training examples:

1.  $\langle \text{Sunny}, \text{Warm}, \text{Normal}, \text{Strong}, \text{Warm}, \text{Same} \rangle$ ,  $\text{Enjoy Sport} = \text{Yes}$
2.  $\langle \text{Sunny}, \text{Warm}, \text{High}, \text{Strong}, \text{Warm}, \text{Same} \rangle$ ,  $\text{Enjoy Sport} = \text{Yes}$



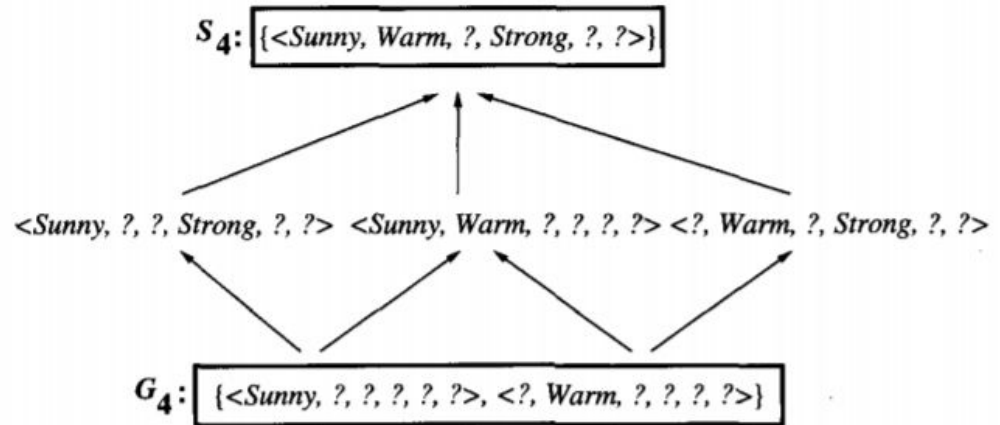
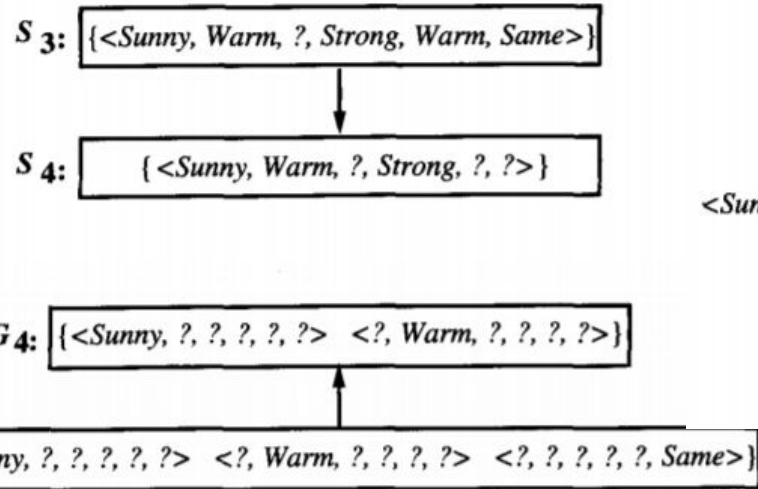
Training Example:

3.  $\langle \text{Rainy}, \text{Cold}, \text{High}, \text{Strong}, \text{Warm}, \text{Change} \rangle$ ,  $\text{EnjoySport} = \text{No}$



# Algoritmo eliminación de candidatos - Ejecución

Gráficamente



Training Example:

4.  $\langle \text{Sunny, Warm, High, Strong, Cool, Change} \rangle$ ,  $\text{EnjoySport} = \text{Yes}$

# Algoritmo eliminación de candidatos

**¿Converge a la hipótesis correcta?** Sí, si se satisface:

- ausencia de error en los ejemplos
- el concepto objetivo está incluido como hipótesis en el espacio de hipótesis  $H$
- 

**Si hay errores en los ejemplos:**

- si el 2do ej. se presenta como -, en vez de +, se removerá la hipótesis
- si  $S$  y  $G$  convergen a un espacio de versiones vacío. Esto indicaría que no hay hipótesis en  $H$  consistente con los ejemplos de entrenamiento

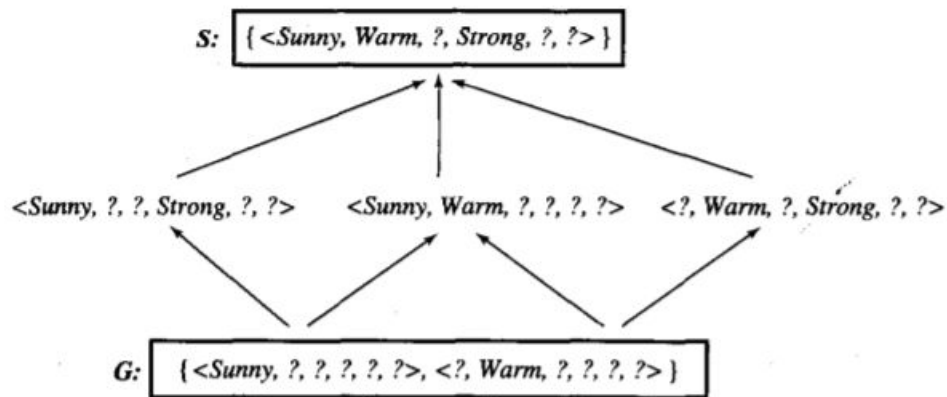
# Algoritmo eliminación de candidatos

- El espacio de versiones es independiente del orden en secuencia de ejemplos de entrenamiento
- Con más ejemplos,  $S$  y  $G$  se van acercando delimitando un espacio de versiones de hipótesis candidatas

# Clasificación de nuevas instancias

En el caso en que **no colapsaron S y G**. Clasificación de nuevas instancias:

- resolver por respuesta igual en todas
- votación



Instance	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
A	Sunny	Warm	Normal	Strong	Cool	Change	?
B	Rainy	Cold	Normal	Light	Warm	Same	?
C	Sunny	Warm	Normal	Light	Warm	Same	?
D	Sunny	Cold	Normal	Strong	Warm	Same	?

# Sesgo en el espacio de hipótesis

-no incluimos todas las posibles hipótesis, sólo permitimos conjunciones de atributos, no disyunciones (ej. sky=sunny  $\vee$  sky= cloudy)

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Cool	Change	Yes
2	Cloudy	Warm	Normal	Strong	Cool	Change	Yes
3	Rainy	Warm	Normal	Strong	Cool	Change	No

$S = \langle ?, \text{Warm}, \text{Normal}, \text{Strong}, \text{Cool}, \text{Change} \rangle$  es la **más específica consistente con ej 1 y ej2**. No funciona para ej3.

Sesgamos a nuestro método a considerar sólo hipótesis con conjunciones.

# Índice

- Concepto
- Aprendizaje de conceptos
- Aprendizaje inductivo
- Adquisición de conceptos como búsqueda
- Algoritmos
- **Sesgo inductivo**

# Sesgo inductivo (Bias)

**Cantidad de posibles ejemplos:**  $3 \cdot 2^5 = 96$

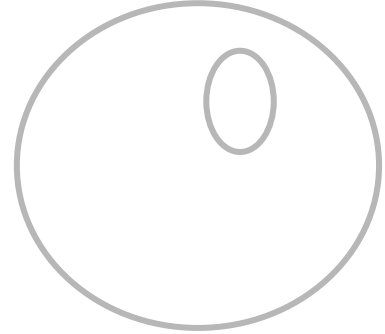
**Cantidad posible de hipótesis:**  $4 \cdot 3^5 + 1 = 973$ .

**Cantidad posible de conceptos :**  $2^{96}$ . gigante.

Si proveemos espacio de hipótesis capaz de representar cada posible concepto..  
tenemos partes de  $x$ ,  $P(X)$  conceptos =  $2^{|X|} = 2^{96}$

Hacemos una **suposición** para facilitar el algoritmo de ML: reducir el espacio de búsqueda. Lo hacemos asumimos que las hipótesis tiene una forma en particular.

Reducción de espacio de conceptos grande a uno chico: sesgo inductivo  
(inductive bias)



# Sesgo inductivo

Un conjunto finito de datos no suele alcanzar para inferir un modelo.

- El **sesgo inductivo** de un algoritmo de aprendizaje es el **conjunto de afirmaciones** que el algoritmo utiliza para construir un modelo.
- El sesgo inductivo incluye:
  - **forma de las hipótesis** (número y tipo de parámetros);
  - **características de funcionamiento del algoritmo** (cómo recorre el espacio de hipótesis hasta elegir un único modelo).



# Resumen

- Concepto y Aprendizaje de Conceptos o Concept Learning
- Relación de orden General-Específico
- Algoritmos de búsqueda de hipótesis
- Sesgo inductivo

# Bibliografía

Mitchell, Cap. 2