Aprendizaje como *generalización*

Introducción

El aprendizaje es un término muy amplio que cubre un extenso rango de procesos:

- El aprendizaje como reestructuración de conocimiento previo para ser utilizado de forma más eficaz (*aprendizaje deductivo*).

- El aprendizaje como ***generalización***: a partir de un número grande de observaciones específicas, extraer y retener las características comunes importantes que caracterizan una clase de estas observaciones (*aprendizaje inductivo*).

El aprendizaje por generalización: Definición del problema

Dados:

1. *Un lenguaje de instancias*.
2. *Un lenguaje de generalización*.
3. *Un test de correspondencia que hace corresponder generalizaciones con instancias*.
4. *Un conjunto de instancias de entrenamiento (positivas y negativas) de la generalización objetivo a ser aprendida*.

Determinar:

Generalizaciones expresables a partir del lenguaje de generalización escogido que sean consistentes con las instancias de entrenamiento presentadas.

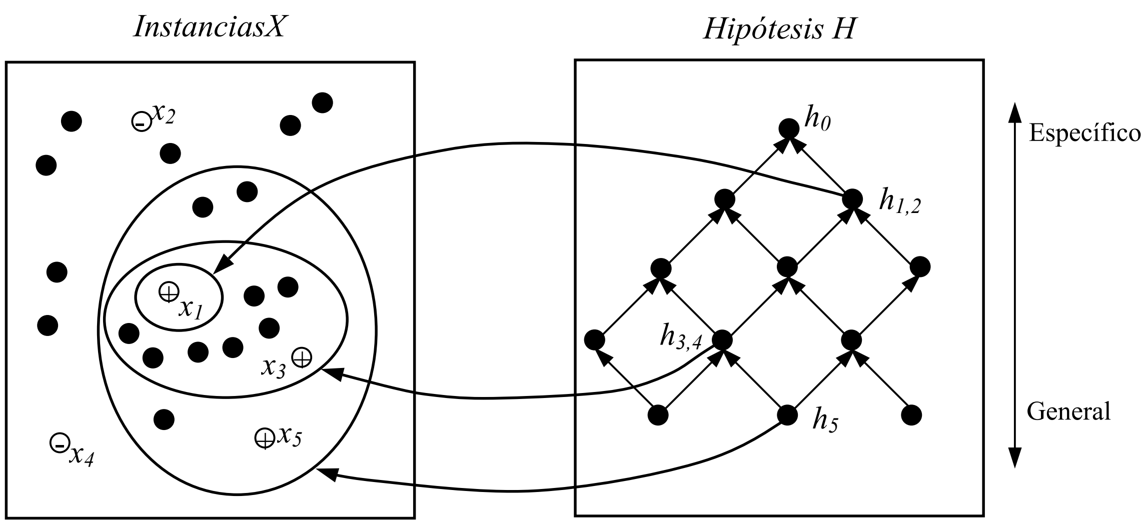
Generalización como búsqueda

El problema de aprendizaje por generalización puede verse como un ***problema de búsqueda***:

- El lenguaje de generalización corresponde a un espacio de hipótesis (espacio de búsqueda) de posibles soluciones.

- El lenguaje de instancias corresponde a un espacio de instancias al que pertenecen los ejemplos positivos/negativos de la generalización objetivo.

- La tarea de aprendizaje consiste en examinar el espacio de hipótesis, sujeto a las restricciones impuestas por las instancias de entrenamiento, para determinar generalizaciones plausibles.



Para que la búsqueda sea eficaz, se necesita definir un criterio de orden sobre el espacio de hipótesis.

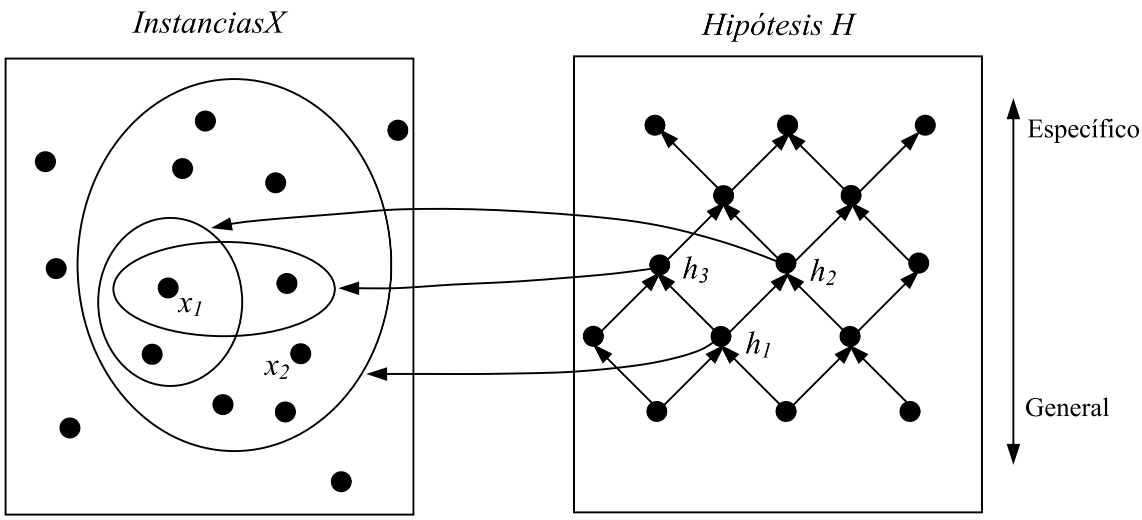
Ordenación

Existe una importante estructura de orden inherente al lenguaje de generalización: la que define la relación "*más general que*" o su relación inversa "*más específica que*".

Definición 1: Sea h_j y h_k funciones booleanas (hipótesis) definidas sobre X. Entonces h_j es ***igual o más general que*** h_k (escrito $h_j \geq_g h_k$) si y sólo si

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

Definición 2: Una hipótesis h_j es ***más general que*** una hipótesis h_k si el conjunto de todas las instancias *asociadas* a h_k está contenido en el conjunto de todas las instancias *asociadas* a h_j.



La relación $\geq_g$ define una ***estructura de orden parcial*** en el espacio de hipótesis H.

La relación $\geq_g$ es importante porque proporciona una base muy útil para organizar la búsqueda a través del espacio de hipótesis H incluso en espacios infinitos de hipótesis sin enumerar explícitamente todas las hipótesis.

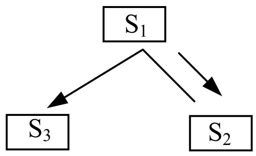
Métodos de generalización

La caracterización de la generalización como búsqueda permite describir métodos de generalización mediante estrategias que son independientes de los lenguajes de instancias y de generalización utilizados.

<u>Ejemplos de estrategias</u>:

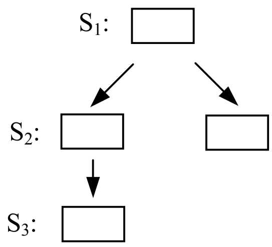
- Estrategias incrementales:

  - Búsqueda primero en profundidad (*Depth-first search*).



(Inst. +) $\rightarrow S_1$
(Inst. +) $\rightarrow S_2$
(Inst. -) $\rightarrow S_3$

  - Búsqueda primero en amplitud (*Breadth- first search*) de lo específico a lo general.



(Inst. +) $\rightarrow S_1$
(Inst. +) $\rightarrow S_2$
(Inst. -) $\rightarrow S_3$

  - Estrategia del espacio de versiones.

- Estrategias no incrementales

- Estrategias estadísticas

Generalización: *Aprendizaje de Conceptos*

Definición de concepto (categoría)

Un concepto o categoría se puede definir como:

- La descripción de algún subconjunto de objetos o eventos definidos sobre un conjunto mayor (por ejemplo, el subconjunto de animales que constituyen mamíferos).

- Alternativamente, se puede pensar en un concepto como una función booleana definida sobre un conjunto X, es decir

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

  A partir de esta definición, se puede operativizar la definición conjuntista anterior. Concretamente, un concepto C_0 estaría formado por el subconjunto

  $$C_0=\{x \in X \text{ t.q. } c_0(x)=1\}$$

Elementos que definen la tarea de aprendizaje de conceptos

La tarea aprendizaje de conceptos se puede describir por:

- Un conjunto de instancias, X.
- Un función objetivo c tal que

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

- Un conjunto de hipótesis candidatas, H. Cada hipótesis es una función h tal que

$$h\text{: } X \rightarrow \{0,1\}$$

- Un conjunto de ejemplos de entrenamiento, $D \subset X$, de la función objetivo. Cada ejemplo se puede representar por x y, alternativamente, por el par $<x, c(x)>$.

Una definición operativa de
la tarea aprendizaje de conceptos

Dados un conjunto de instancias, X, una función objetivo, c, un espacio de hipótesis, H, y un conjunto de ejemplos de entrenamiento, D, la tarea de aprendizaje de conceptos ***consiste en determinar una hipótesis h tal que sea consistente con D***.

Consistencia

Una hipótesis h es **consistente** con un conjunto de ejemplos de entrenamiento D si y sólo si $h(x) = c(x)$ para cada ejemplo $<x, c(x)>$ en D.

$$\text{Consistente}(h, D) \equiv (\forall <x, c(x)> \in D)\ h(x) = c(x)$$

El aprendizaje de conceptos como búsqueda

El aprendizaje de conceptos puede ser visto como la tarea de búsqueda a través de un espacio grande de hipótesis que queda definido por la forma de representar las hipótesis.

Formas de representación de hipótesis

Implícitamente, define el espacio de hipótesis que el algoritmo sólo puede representar y, por tanto, que sólo puede aprender.

Ejemplo: cada hipótesis se representa como una *conjunción de restricciones* sobre los atributos de una instancia. Para cada atributo, la hipótesis almacenará

- Una variable x_i (cualquier valor es aceptable) ó
- Un valor específico ó
- El valor ∅ (ningún valor es aceptable).

Si tenemos un problema de aprendizaje de conceptos en el que cada instancia se representa por 5 atributos, los cuales tienen 5, 4, 5, 3 y 4 valores posibles, respectivamente, entonces existirán

5x4x5x3x4 = 1200 instancias distintas en *X*

7x6x7x5x6 = 8820 hipótesis sintácticamente distintas en *H*

1+(6x5x6x4x5) = 3601 hipótesis semánticamente distintas en *H*

Algoritmo *Find-S*: Encontrar la hipótesis más específica

Utiliza el orden parcial definido por la relación $\geq_g$ para buscar aquella hipótesis consistente con los ejemplos de entrenamiento observados.

1. Inicializa h a la hipótesis más específica.
2. Para cada ejemplo de entrenamiento x

   2.1 Si x es negativo, no hacer nada.

   2.2 Si x es positivo, para cada restricción de atributo a_i en h

   IF la restricción a_i es satisfecha por x
   THEN no hacer nada.
   ELSE reemplazar a_i en h por la siguiente restricción más general que es satisfecha por x.
3. Hipótesis de salida h.

"El algoritmo *Find-S* no trata los ejemplos negativos"

Algoritmo *Find-S*: Ejemplo

Vocabulario para el dominio de animales exóticos					
Origen	Á(frica	AM(érica)	AS(ia)	E(uropa)	O(ceanía)
Clase	Mamífero	Ave	Pez	Reptil	
Alimentación	Carnívoro	Hervíboro	Omnívoro	Insectívoro	Piscívoro
Valor	Alto	Normal	Bajo		
Situación	Peligro	Normal	Extinguido	Desconocida	

Dominios de definición de los atributos

Origen	A(frica)	A(frica)	A(frica)	E(uropa)	A(frica)
Clase	Mamífero	Reptil	Reptil	Mamífero	Mamífero
Alimentación	Carnívoro	Herbívoro	Herbívoro	Herbívoro	Carnívoro
Valor	Alto	Bajo	Alto	Bajo	Normal
Situación	Peligro	Normal	Peligro	Peligro	Peligro
Ejemplo	+	-	+	-	+

Conjunto de entrenamiento

En el espacio de hipótesis *H* cada hipótesis está formada por una conjunción de restricciones sobre los valores de sus atributos.

Inicio
Se inicializa *h* a la hipótesis más específica ⇒ $\boldsymbol{h_0 = (\varnothing, \varnothing, \varnothing, \varnothing, \varnothing)}$

Primer ejemplo (A, M, C, A, P; +) ⇒ $\boldsymbol{h_1 = (A, M, C, A, P)}$

Segundo ejemplo (A, R, H, B, N; -) ⇒ $\boldsymbol{h_2 = h_1 = (A, M, C, A, P)}$

Tercer ejemplo (A, R, H, A, P; +) ⇒ $\boldsymbol{h_3 = (A, x2, x3, A, P)}$

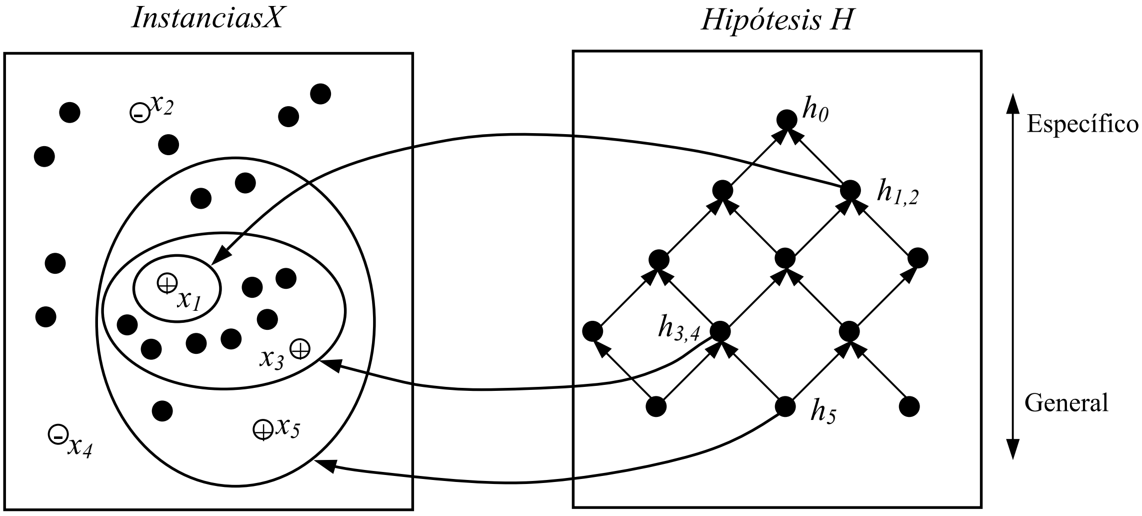
Cuarto ejemplo (E, M, H, B, P; -) ⇒ $\boldsymbol{h_4 = h_3 = (A, x2, x3, A, P)}$

Quinto ejemplo (A, M, C, N, P; +) ⇒ $\boldsymbol{h_5 = (A, x2, x3, x4, P)}$

Concepto aprendido:

"***Animales africanos en peligro de extinción***"

Algoritmo *Find-S*: Ejemplo (Cont.)



$x1 = (A, M, C, A, P, +)$
$x2 = (A, R, H, B, N, -)$
$x3 = (A, R, H, A, P, +)$
$x4 = (E, M, H, B, P, -)$
$x5 = (A, M, C, N, P, +)$

$h_0 = (\emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$
$h1 = (A, M, C, A, P)$
$h2 = (A, M, C, A, P)$
$h3 = (A, x2, x3, A, P)$
$h4 = (A, x2, x3, A, P)$
$h5 = (A, x2, x3, x4, P)$

Inconvenientes del algoritmo Find-S

- La hipótesis de salida del algoritmo no garantiza la unicidad

- La hipótesis de salida del algoritmo siempre es la más específica

- El algoritmo Find-S es muy sensible al ruido.

- El algoritmo Find-S falla cuando en el espacio de hipótesis elegido existen varias hipótesis específicas, consistentes y maximales

Espacio de Versiones

Definición: El ***espacio de versiones***, denotado por $VS_{H,D}$, con respecto al espacio de hipótesis *H* y los ejemplos de entrenamiento *D*, es el subconjunto de hipótesis de *H* consistente con todos los ejemplos de entrenamiento en *D*.

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

Definición: el ***límite general G***, con respecto a H y D, es el subconjunto de hipótesis generales maximales de H consistentes con D.

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

Definición: el ***límite específico S***, con respecto a H y D, es el subconjunto de hipótesis generales minimales (específicas maximales) de H consistentes con D.

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

Espacio de versiones = f(*G*, *S*)
Si los conjuntos *G* y *S* están bien definidos, el espacio de versiones está formado por el conjunto de hipótesis contenidas en *G*, más las contenidas en *S*, más aquellas que caen entre *G* y *S* en el espacio de hipótesis parcialmente ordenado

Algoritmo de aprendizaje "Eliminación-candidatos"

- Soluciona algunas de las limitaciones de *Find-S*.

- La idea clave es producir una descripción del conjunto de ***todas las hipótesis consistentes con los ejemplos de entrenamiento***, es decir, del espacio de versiones.

ALGORITMO ELIMINACIÓN-CANDIDATOS (CANDIDATE-ELIMINATION):

[1] Inicialización

  [1.1] Inicializar G al conjunto de hipótesis generales maximales de H.

  [1.2] Inicializar S al conjunto de hipótesis específicas maximales de H.

[2] Para cada ejemplo de entrenamiento $d \in D$ hacer:

  [2.1] Si d es un ejemplo positivo

    [2.1.1] Eliminar de G cualquier hipótesis inconsistente con d.

    [2.1.2] Para cada hipótesis $s \in S$ que no sea consistente con d

      [2.1.2.1] Eliminar s de S.

      [2.1.2.2] Añadir a S todas las generalizaciones minimales h de s tales que h sea consistente con d y algún miembro de G sea más general que h.

    [2.1.3] Eliminar de S cualquier hipótesis que sea más general que otra hipótesis de S.

  [2.2] Si d es un ejemplo negativo:

    [2.2.1] Eliminar de S cualquier hipótesis inconsistente con d.

    [2.2.2] Para cada hipótesis $g \in G$ que no sea consistente con d

      [2.2.2.1] Eliminar g de G.

      [2.2.2.2] Añadir a G todas las especializaciones minimales h de g tales que h sea consistente con d y algún miembro de S es más específico que h.

    [2.2.3] Eliminar de G cualquier hipótesis que sea menos general que otra hipótesis de G.

  [2.3] Si al procesar d se tiene que $S = G$, fin del algoritmo (S ó G es el concepto aprendido).

Algoritmo "Eliminación de Candidatos": Ejemplo

Inicio

$$\mathbf{S_0=\{(\varnothing, \varnothing, \varnothing, \varnothing, \varnothing)\}}$$

$$\mathbf{G_0=\{(x1, x2, x3, x4, x5)\}}$$

Primer ejemplo, d = (A,M,C,A,P,+)

G_0 es consistente con d $\Rightarrow$ $\mathbf{G_1=G_0\{(x1, x2, x3, x4, x5)\}}$.

$s_{01}\in S_0$ es inconsistente con d, se elimina de S y se añaden todas las generalizaciones minimales de s_{01} para contener a d

$$\mathbf{S_1=\{(A, M, C, A, P)\}}$$

Se cumple que g_{11} es más general que s_{11} y, además, no aplica [2.1.3].

Segundo ejemplo, d = (A,R,H,B,N,-)

S_1 es consistente con d $\Rightarrow$ $\mathbf{S_2=S_1=\{(A, M, C, A, P)\}}$.

$g_{11}\in G_1$ es inconsistente con d. Se elimina de G y se hacen todas las especializaciones minimales de g_{11} que sean consistentes con d.

Puesto que d es un contraejemplo, el conjunto de hipótesis que no incluyen el contraejemplo son:

$$\overline{(A \wedge R \wedge H \wedge B \wedge N)} = \overline{A} \vee \overline{R} \vee \overline{H} \vee \overline{B} \vee \overline{N} =$$
= "no A" o "no R" o "no H" o "no B" o "no N"

La especialización minimal de g_{11} es:

G_2'={(no A, x2, x3, x4, x5), (x1, no R, x3, x4, x5), (x1, x2, no H, x4, x5), (x1, x2, x3, no B, x5), (x1, x2, x3, x4, no N)}

Para todo $h\in G_2$', algún miembro de S_2 debe ser más específico que h:

$$\mathbf{G_2=\{(x1, M, x3, x4, x5), (x1, x2, C, x4, x5), (x1, x2, x3, A, x5), (x1, x2, x3, x4, P)\}}$$

Tercer ejemplo, d = (A,R,H,A,P,+)

Como {(x1, M, x3, x4, x5), (x1, x2, C, x4, x5)} $\in G_2$ son inconsistentes con *d*, se eliminan de G

$$\mathbf{G_3=\{ (x1, x2, x3, A, x5), (x1, x2, x3, x4, P)\}}$$

$s_{21} \in S_2$ es inconsistente con *d*. Se elimina de S y se añaden todas las generalizaciones minimales de s_{21} que contengan a *d*

$$\mathbf{S_3=\{(A, x2, x3, A, P)\}}$$

Cualquier hipótesis contenida en G_3 es más general que la obtenida en S_3.

Cuarto ejemplo, *d* = (E,M,H,B,P,-)

S_3 es consistente con *d* $\Rightarrow$ $\mathbf{S_4=S_3=\{(A, x2, x3, A, P)\}}$.

Qué hipótesis de G_3 son consistentes con *d*:

g_{31} = (x1, x2, x3, A, x5) sí lo es.

g_{32} = (x1, x2, x3, x4, P) no lo es. Se elimina de G y se hacen todas las especializaciones minimales de g_{32} que sean consistentes con *d*:

$$G_4' = \{(\text{no E}, x2, x3, x4, P), (x1, \text{no M}, x3, x4, P), (x1, x2, \text{no H}, x4, P), (x1, x2, x3, \text{no B}, P)\}$$

Por tanto

$$G_4'' = g_{31} \cup G_4'$$

Para todo $h \in G_4''$, algún miembro de S_4 tiene que ser más específico que h:

$$\mathbf{G_4''' = \{(x1, x2, x3, A, x5), (A, x2, x3, x4, P), (x1, x2, x3, A, P)\}}$$

Finalmente, como g'''_{41} es más general que g'''_{43}

$$\mathbf{G_4= \{(x1, x2, x3, A, x5), (A, x2, x3, x4, P) \}}$$

Quinto ejemplo, d = (A,M,C,N,P,+)

De las hipótesis de G_4 sólo (A, x2, x3, x4, P) es consistente con d:

$$\mathbf{G_5= G_4=\{(A, x2, x3, x4, P)\}}$$

$s_{41} \in S_4$ es inconsistente con d. Se elimina de S y se añaden todas las generalizaciones minimales de s_{41} que contengan a d

$$\mathbf{S_5=\{(A, x2, x3, x4, P)\}}$$

Cualquier hipótesis contenida en G_5 es más general que la obtenida en S_5.

<u>**Fin del algoritmo**</u>

Como $\boldsymbol{G} = \boldsymbol{S}$, el algoritmo finaliza obteniendo como concepto aprendido:

"***Animales (de origen) A-fricano en (situación) P-eligro de extinción***"

Comparativa en cuanto al tiempo de cálculo y costes máximos de almacenamiento de los algoritmos *Find-S* y *Eliminación de candidatos*

Estrategia	Tiempo	Espacio de almacenamiento
Find-S	$O(spn + s^2p)$	$O(s + n)$
Espacio de Versiones	$O(sg (p + n) + s^2p + g^2n)$	$O(s + g)$

p = No. de instancias positivas
n = No. de instancias negativas
s = tamaño máximo alcanzado por el conjunto *S*
g = tamaño máximo alcanzado por el conjunto *G*

Limitaciones del algoritmo *eliminación-candidatos*

El espacio de versiones aprendido por el algoritmo converge hacia la hipótesis que describe el concepto objetivo siempre que:

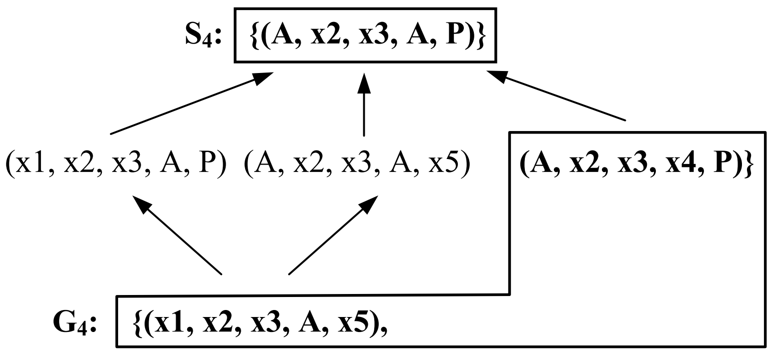
[1] No hay errores en el conjunto de ejemplos de entrenamiento.

[2] El concepto objetivo buscado pertenece al espacio de hipótesis *H*.

Corolario: Si, dados H y D, el algoritmo converge a un espacio de versiones vacío, entonces la causa es debida al no cumplimiento de las condiciones [1] y/o [2].

Cómo utilizar conceptos parcialmente aprendidos (espacio de versiones) para clasificar nuevos ejemplos

Si en el caso anterior sólo existiesen los 4 primeros ejemplos, el algoritmo no aprendería el concepto objetivo. No obstante, existe un conocimiento parcial del concepto igual a su espacio de versiones.



Espacio de Versiones

	[I]	[II]	[III]
Origen	A(frica)	A(frica)	AM(érica)
Clase	Pez	Pez	Pez
Alimentación	Insectívoro	Insectívoro	Insectívoro
Valor	Alto	Normal	Alto
Situación	Peligro	Peligro	Normal
Ejemplo	?	?	?

Nuevas instancias de clasificación desconocida

La instancia [I], (A, P, I, A, P) $\notin D$, es clasificada como positiva por todas las hipótesis en el actual espacio de versiones $\Rightarrow$ **Instancia positiva**.

La instancia [II], (A, P, I, N, P) $\notin D$, es clasificada como negativa por todas las hipótesis en el espacio de versiones $\Rightarrow$ **Instancia negativa**.

La instancia [III], (AM, P, I, A, N) $\notin D$, es clasificada como positiva por sólo una hipótesis del espacio de versiones y negativa por las otras cuatro hipótesis restantes. Si se asume que todas las hipótesis en H son igualmente probables a priori, entonces:

$$p = 1/5 \text{ instancia positiva}$$

$$p = 4/5 \text{ instancia negativa}$$

Formas de suministrar los ejemplos al algoritmo

[1] El profesor selecciona previamente el conjunto de entrenamiento. Los ejemplos se suministran secuencialmente al aprendiz.

[2] En cada iteración, es el propio aprendiz el que elige la instancia siguiente a procesar. Un oráculo externo establece su clasificación correcta. El aprendiz la procesa.

Si se elige la opción [2], la mejor estrategia que debería seguir el aprendiz para aprender con el mínimo número de ejemplos, sería aquella que permita reducir las hipótesis contenidas en el espacio de versiones, en el momento actual, justo a la mitad.

El número de instancias necesarias para aprender el concepto objetivo será:

$$\lceil \log_2 |H|_{ini} \rceil$$

donde $|H|_{ini}$ es el número de hipótesis totales contenidas en el espacio de hipótesis.

Bias inductivas

Definición de *bias*: "*cualquier factor que influye en la creación o selección de hipótesis*".

Se pueden establecer de dos formas diferentes:

1. ***Restringiendo el espacio de hipótesis (restriction bias)***. Hay que garantizar que la hipótesis objetivo se encuentre en dicho espacio restringido.

   Ejemplo: En el aprendizaje de conceptos: la hipótesis buscada sólo puede tener una forma conjuntiva.

2. ***Limitando la búsqueda en el espacio mediante algún criterio (preference bias)***. Hay que garantizar que el criterio de búsqueda encuentra la hipótesis objetivo.

   Ejemplo: Algoritmo ID3.

Algunos algoritmos de aprendizaje combinan ambos tipos de bias.

Sólo introduce *bias* el espacio de hipótesis y/o el criterio de búsqueda que es incompleto.

Definición formal de *bias* inductiva

Sea:

- L un algoritmo de aprendizaje.
- $D_c=\{<x,c(x)>\}$un conjunto de ejemplos de entrenamiento.

Después del entrenamiento, L es preguntado para clasificar una nueva instancia x_i. Sea:

- $L(x_i, D_c)$ la clasificación que L asigna a x_i

La ***bias inductiva*** de L es cualquier conjunto mínimo se afirmaciones B tal que para cualquier concepto objetivo c y correspondientes ejemplos de entrenamiento D_C

$$(\forall x_i \in X)[(B \wedge D_c \wedge x_i) \vdash L(x_i, \mathrm{D_c})]$$

Ventajas de ver los sistemas de inferencia inductivos en términos de sus bias inductivas

[1] Proporciona un medio no procedimental de caracterizar su política generalizando más allá de los datos observados.

[2] Permite la comparación de diferentes sistemas de aprendizaje de acuerdo a la *potencia* de la bias inductiva que utilizan. A mayor potencia, permitirá clasificar una mayor proporción de instancias no vistas.

Ejemplos de bias inductivas

Bias inductiva del algoritmo "*aprendizaje de memoria*" (*rote-learner)*:
El algoritmo corresponde simplemente a almacenar cada ejemplo de entrenamiento observado en memoria. Subsecuentes instancias se clasifican mirando en memoria: si la instancia es encontrada, se devuelve la clasificación almacenada. En otro caso, el sistema reusa clasificar la nueva instancia.

Bias: $B=\{\varnothing\}$, es decir, este algoritmo no tiene *bias*.

Bias inductiva del algoritmo "*eliminación-candidatos"*:
Si suponemos que el algoritmo sólo producirá una clasificación de x_i si el voto de las hipótesis del espacio de versiones es unánimemente positivo o negativo y no producirá salida de clasificación en otro caso, entonces:

$B = \{c \in H\}$, es decir, "el concepto objetivo c está contenido en el espacio de hipótesis H dado".

Bias inductiva del algoritmo *"Find-S*":
Tiene dos *bias*:

[1] $c \in H$.

[2] Todas las instancias son negativas a menos que lo opuesto sea acarreado por su otro conocimiento.

Espacio de hipótesis con *bias* inadecuadas

	[I]	[II]	[III]
Origen	A(frica)	AS(ia)	AM(érica)
Clase	Mamífero	Mamífero	Mamífero
Alimentación	Carnívoro	Carnívoro	Carnívoro
Valor	Alto	Alto	Alto
Situación	Peligro	Peligro	Peligro
Ejemplo	+	+	-

Un nuevo conjunto de entrenamiento

Si el espacio de hipótesis se restringe sólo a la inclusión de hipótesis formadas por conjunciones de valores de atributos, sería incapaz de representar conceptos disyuntivos tales como

"*Origen=Africa o Origen=Asia*".

El espacio de versiones, tras procesar los tres ejemplos, se reduce al conjunto vacío.

Tras procesar los dos primeros ejemplos, la hipótesis más específica consistente con ellos es

$$S_2=\{(x1, M, C, A, P)\}$$

Esta hipótesis, aunque es la hipótesis máximamente específica de *H* consistente con los 2 primeros ejemplos, es ya demasiado general: cubre también de forma errónea el 3er. ejemplo negativo.

El problema es que hemos ***sesgado*** al aprendiz a considerar sólo hipótesis conjuntivas. Para solucionar este problema, tendríamos que recurrir a un espacio de hipótesis más expresivo.

Espacio de hipótesis sin *bias*

Solución al problema anterior: elegir un espacio de hipótesis que sea capaz de representar todos los subconjuntos del conjunto de instancias *X*.

En general, el conjunto de todos los subconjuntos de *X* es llamado *conjunto potencia de X*.

Puede demostrarse que el número de distintos subconjuntos que puede ser definido sobre un conjunto *X* que contiene $|X|$ elementos es $2^{|x|}$.

En nuestro ejemplo anterior, el espacio conjuntivo de hipótesis era sólo capaz de representar

$$3601 \text{ de las } 2^{1200} \approx 10^{361} \text{ hipótesis posibles.}$$

La elección del espacio de hipótesis *completo* elimina todo problema de expresabilidad. Sin embargo, cae en un nuevo problema:

El algoritmo de aprendizaje de conceptos es completamente incapaz de generalizar más allá de ejemplos observados.

Ejemplo: supongamos que se presentan tres ejemplos positivos {x1, x2, x3} y dos ejemplos negativos {x4, x5}, entonces

$$S=\{(x1 \vee x2 \vee x3)\} \text{ y } G=\{\neg(x4 \vee x5)\}$$

Los únicos ejemplos que serán clasificados sin ningún tipo de ambigüedad por el *VS* serán los propios ejemplos de entrenamiento observados.

Para converger a un único concepto objetivo final, tendremos que presentar todas las instancias en X como ejemplos de entrenamiento, entonces:

$$c = S \wedge G$$

Se podría pensar en un *VS* parcialmente aprendido y clasificar cada instancia no vista mediante la votación de cada una de las hipótesis pertenecientes a *VS*:

- Las únicas instancias que darán lugar a una votación unánime serán las previamente observadas.

- Cada instancia no observada previamente será clasificada positiva justo por la mitad de las hipótesis de *VS* y como negativa justo por la otra mitad.

Propiedad de la inferencia inductiva: Un aprendiz que no haga suposiciones *a priori* respecto del concepto objetivo, no tiene ninguna base racional para clasificar cualquier instancia no vista previamente.