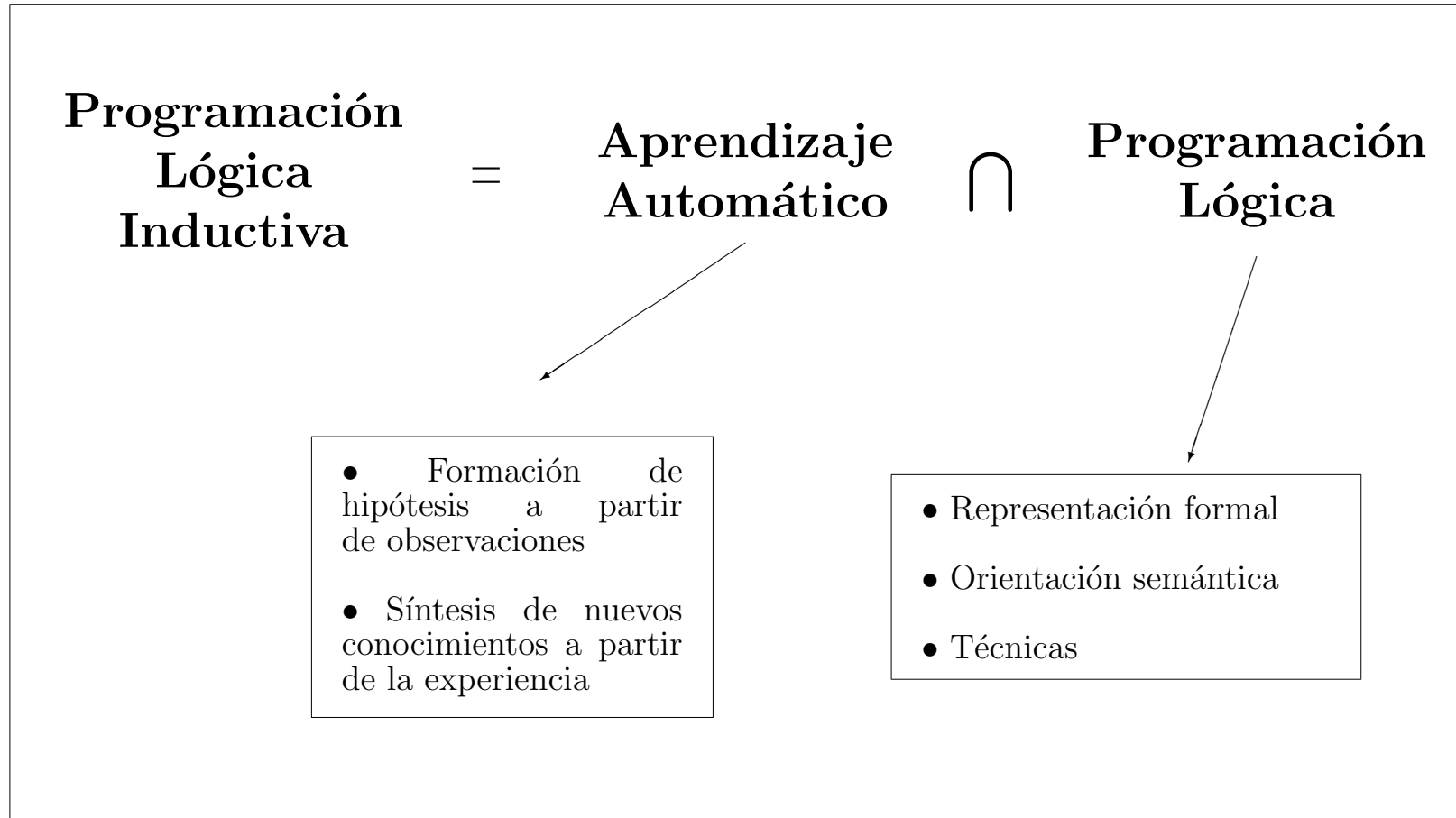


Tema 13: Introducción a la Programación Lógica Inductiva

José A. Alonso Jiménez
Miguel A. Gutiérrez Naranjo

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

Introducción



Desarrollo reciente

- Los sistemas de ILP han sido usados con éxito en una gran variedad de dominios, incluyendo
 - Ecología
 - Ingeniería
 - Biología molecular
 - Procesamiento del lenguaje natural
 - Control del tráfico
 - ...

Aprendizaje Automático

El Aprendizaje Automático estudia cómo construir programas que mejoren automáticamente con la experiencia.

-
- Recientes avances en la teoría y los algoritmos.
 - Crecimiento desbordante de datos “en línea” (on line).
 - Se dispone de máquinas suficientemente potentes.
 - Interés por parte de la industria.

Disciplinas relacionadas

- Inteligencia Artificial
- Métodos bayesianos
- Teoría de la complejidad
- Teoría de control
- Teoría de la información
- Filosofía
- Psicología y neurobiología
- Estadística
- ...

Aprendizaje de conceptos

Distintas aproximaciones:

- Un concepto no es más que el conjunto de todas sus instancias.
- 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.

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

Un ejemplo

Cielo	Temperatura	Humedad	Viento	Agua	Previsión	Hacer_deporte
Soleado	Templada	Alta	Débil	Fría	Sigue_igual	No
Soleado	Templada	Normal	Débil	Fría	Sigue_igual	Sí
Nublado	Templada	Normal	Débil	Fría	Sigue_igual	Sí
Lluvioso	Templada	Normal	Fuerte	Templada	Cambio	No
Lluvioso	Templada	Normal	Débil	Fría	Cambio	Sí

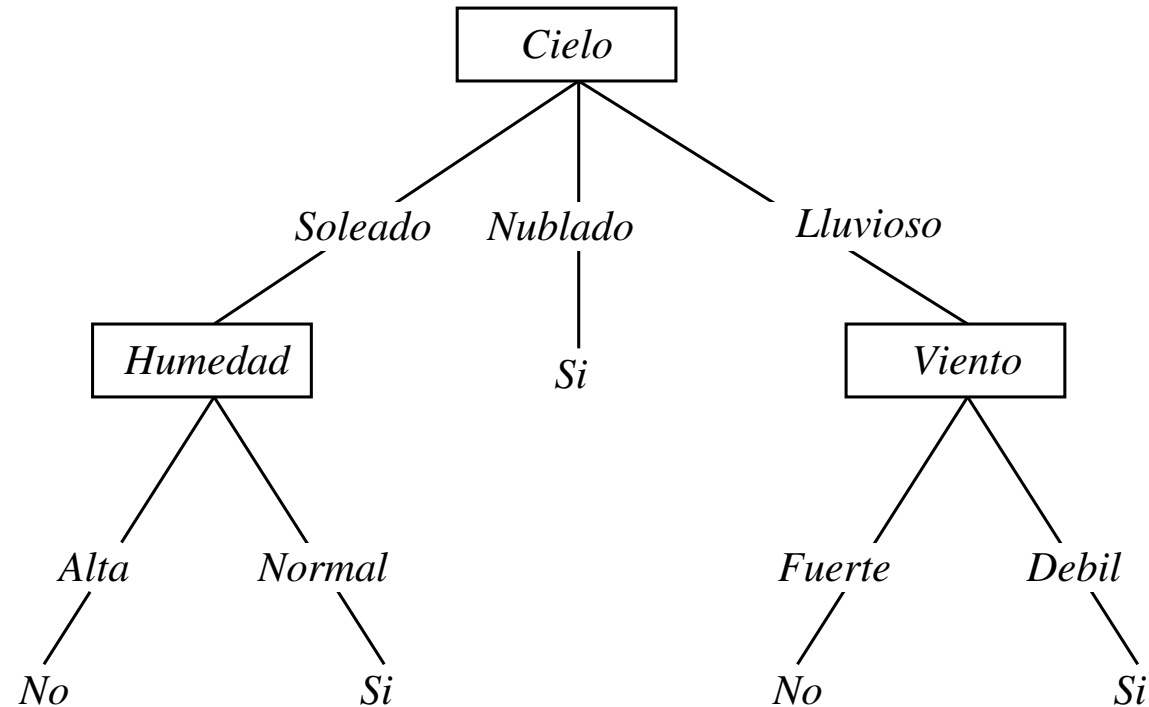
- ¿Cuándo hacemos deporte?
- ¿Podemos definir el concepto *Hacer_Deporte*?

El problema de la representación (I)

- Pares atributo–valor (Lógica proposicional)
- **Hipótesis:**
 - Disyunción de conjunciones de pares atributo–valor
$$\begin{aligned} & (\text{Cielo}=\text{Soleado} \wedge \text{Humedad}=\text{Normal}) \\ & \vee (\text{Cielo}=\text{Nublado}) \\ & \vee (\text{Cielo}=\text{Lluvioso} \wedge \text{Viento}=\text{Débil}) \end{aligned}$$
 - Como conjunto de reglas
 - * **Si** Cielo=Soleado **y** Humedad=Normal **entonces** Hacer_deporte
 - * **Si** Cielo=Nublado **entonces** Hacer_deporte
 - * **Si** Cielo=Lluvioso **y** Viento=Débil **entonces** Hacer_deporte

El problema de la representación (II)

- Como árbol de decisión



Limitaciones

Limitaciones:

- Una representación formal limitada (lenguaje de pares atributo–valor equivalente al de la lógica proposicional)
- Dificultad del manejo del conocimiento base

Programación Lógica Inductiva (I)

• Datos

- Dos conjuntos de átomos cerrados, E^{\oplus} (*ejemplos positivos*) y E^{\ominus} (*ejemplos negativos*)
- Una teoría T como conocimiento base
- Un conjunto L de fórmulas de primer orden

tales que cumplan:

$$* \textit{ Necesidad a priori: } \exists e^{\oplus} \in E^{\oplus} \quad T \not\vdash e^{\oplus}$$

$$* \textit{ Consistencia a priori: } \forall e^{\ominus} \in E^{\ominus} \quad T \not\vdash e^{\ominus}$$

Programación Lógica Inductiva (II)

● **Encontrar** un conjunto finito $H \subset L$ tal que se cumplan

* *Suficiencia a posteriori*: $\forall e^{\oplus} \in E^{\oplus} \quad T \cup H \vdash e^{\oplus}$

* *Consistencia a posteriori*: $\forall e^{\ominus} \in E^{\ominus} \quad T \cup H \not\vdash e^{\ominus}$

Ejemplos de ILP (I)

Representación estándar:

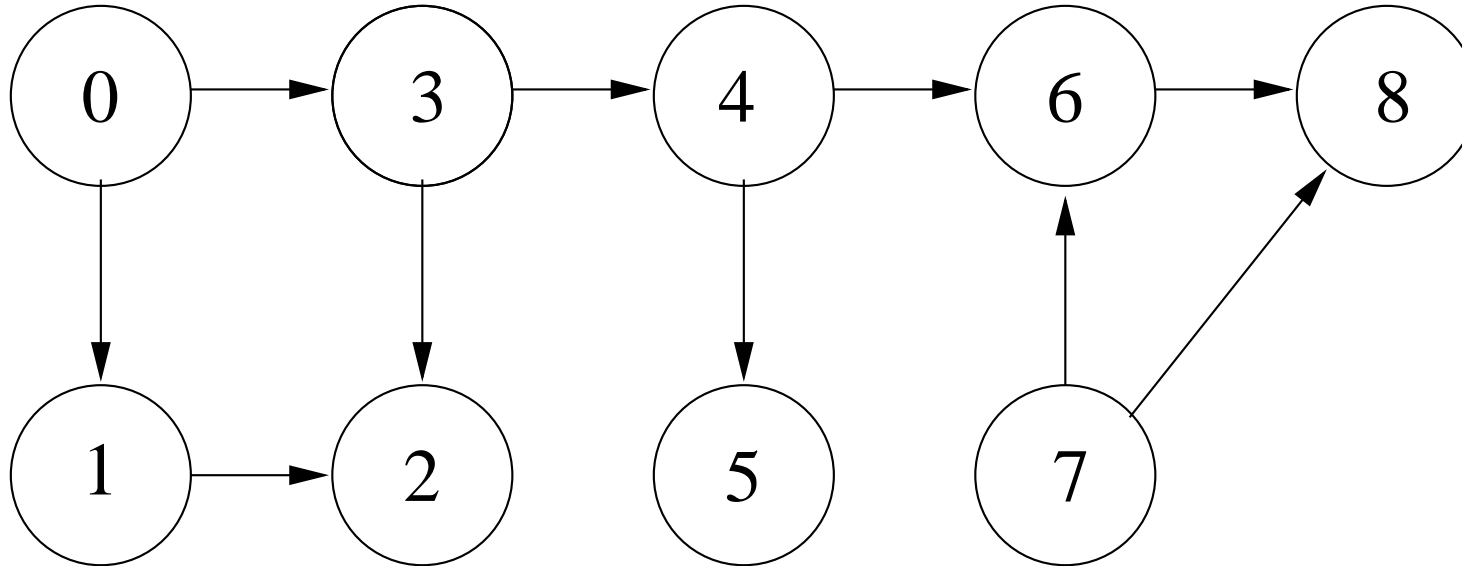
- EJEMPLOS: Átomos cerrados de la relación $hija(X, Y)$.
- CONOCIMIENTO BASE: Definiciones de las relaciones $progenitor(X, Y)$ y $mujer(X)$ (Átomos cerrados o definiciones de predicados).

Conjunto de entrenamiento	Conocimiento base
$hija(maria, ana) \oplus$	$progenitor(ana, maria)$ $mujer(ana)$
$hija(eva, tomas) \oplus$	$progenitor(tomas, eva)$ $mujer(maria)$
$hija(tomas, ana) \ominus$	$progenitor(ana, tomas)$ $mujer(eva)$
$hija(eva, ana) \ominus$	

Hipótesis inducida:

$$hija(X, Y) \leftarrow mujer(X), progenitor(Y, X)$$

Ejemplos de ILP (II)



$$H_0 = \begin{cases} \text{camino}(X,Y) \leftarrow \text{enlace}(X,Y) \\ \text{camino}(X,Y) \leftarrow \text{enlace}(X,Z), \text{camino}(Z,Y) \end{cases}$$

Ejemplos de ILP (III)

- $E^{\oplus} = \{par(0), par(s(s(0))), par(s(s(s(s(0))))), \dots\}$
- $E^{\ominus} = \{par(s(0)), par(s(s(s(0))))), par(s(s(s(s(s(0)))))), \dots\}$

- **Hipótesis:**

$$H_0 = \begin{cases} par(0) \leftarrow \\ par(s(s(X))) \leftarrow par(X) \end{cases}$$

Métodos (I)

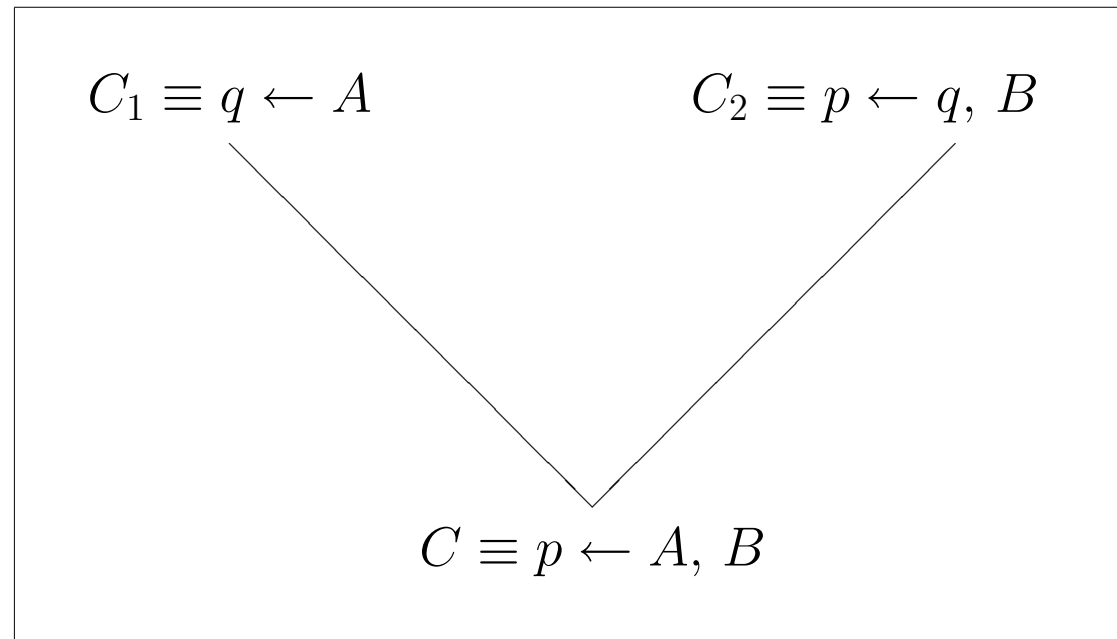
Ascendente:

- Comenzamos por una hipótesis demasiado específica, i.e., que no cubre todos los ejemplos positivos y por tanto, debe ser generalizada.

– *Técnicas:*

- * Resolución inversa (CIGOL)
- * Menor generalización (Golem)
- * ...

Resolución



Menor generalización (I)

Ejemplo 1:

$$\left. \begin{array}{l} \text{expr(una,mujer)} \\ \text{expr(una,niña)} \end{array} \right\} \Longrightarrow \text{expr(una,X)}$$

Ejemplo 2:

$$\left. \begin{array}{l} \text{expr(el,hombre)} \leftarrow \text{masc(el), masc(hombre)} \\ \text{expr(un,niño)} \leftarrow \text{masc(un), masc(niño)} \end{array} \right\}$$
$$\Longrightarrow \text{expr(X,Y)} \leftarrow \text{masc(X), masc(Y)}$$

Menor generalización (II)

Ejemplo 3:

	T		O^+		O^-
masc(el)←		fem(la)←		expr(el,niño)←	expr(el,mujer)←
masc(un)←		fem(una)←		expr(una,mujer)←	expr(una,niño)←
masc(niño)←		fem(niña)←		expr(el,hombre)←	expr(un,niña)←
masc(hombre)←		fem(mujer)←			

$$H_0 = \begin{cases} \text{expr}(X,Y) \leftarrow \text{masc}(X), \text{masc}(Y) \\ \text{expr}(X,Y) \leftarrow \text{fem}(X), \text{fem}(Y) \end{cases}$$

Métodos (II)

Descendente:

- Comenzamos por una hipótesis demasiado general, i.e., que cubre alguno de los ejemplos negativos y por tanto, debe ser especializada.

– *Técnicas:*

- * Inferencia de modelos (MIS)
- * Método extensional (Foil)
- * ...

Inferencia de modelos (I)

- Infiere una axiomatización finita de un *modelo desconocido* M a partir de átomos cerrados verdaderos o falsos en M

Algoritmo:

$P \leftarrow$ El programa cuya única cláusula es la cláusula vacía
repite

mientras P no sea consistente con el conjunto de entramiento **hacer**

- **Si** $P \rightarrow e$ y e es un ejemplo negativo **entonces**
hacer P *más específica*
- **Si** $P \not\rightarrow e$ y e es un ejemplo positivo **entonces**
hacer P *más general*

para siempre

Excepción: Si comprobar $P \rightarrow e$ requiere un número de pasos de resolución *excesivo* eliminar *selectivamente* una cláusula de P

Inferencia de modelos (II)

- Supongamos que tenemos los ejemplos

elemento(2,[4,3,2])	⊕
elemento(2,[2,3])	⊕
elemento(2,[4])	⊖

- La hipótesis actual es

$$H_0 = \begin{cases} \text{elemento}(X,[Y|Z]) \leftarrow Y=2 \\ \text{elemento}(X,[Y|Z]) \leftarrow \text{elemento}(X,Z) \end{cases}$$

- Si ahora consideramos un nuevo ejemplo

elemento(3,[4,2])	⊖
-------------------	---

Encontramos que la cláusula

$$\text{elemento}(X,[Y|Z]) \leftarrow Y=2$$

debe ser eliminada

KDD

El *Descubrimiento de conocimiento en bases de datos* es el proceso de identificar en los datos estructuras *válidas, novedosas, potencialmente útiles* y en última instancia *comprensibles*

(U. Fayyad)

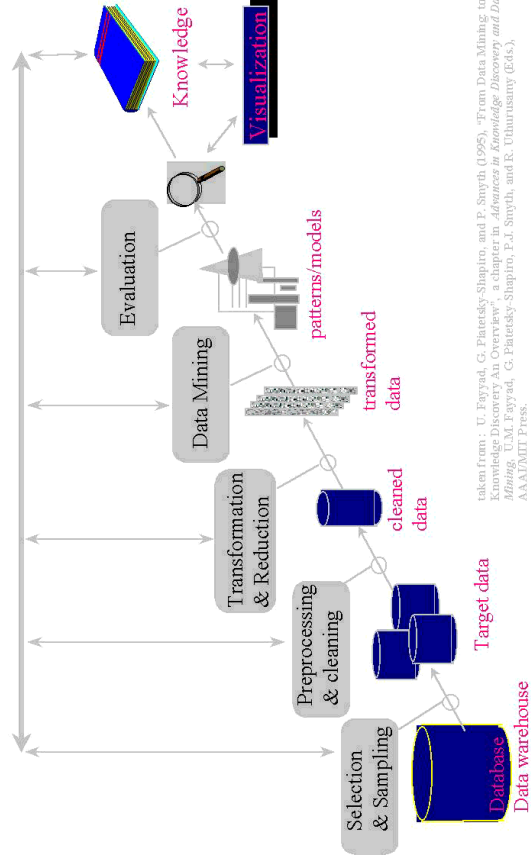
KDD: Fases

Fases de un proceso KDD

- Dominio de la aplicación
 - Conocimiento de la aplicación
 - Objetivos del proceso
- Creación de la base de datos
- Preproceso y limpieza de datos
- Reducción de variables y datos
- Elección del modelo resultante: Resumen, clasificación, regresión, ...
- **Minería de datos** (Aprendizaje, ILP, ...)
- Interpretación
- Uso del conocimiento adquirido

ILP y KDD

- **ILP en KDD:** Proceso de extracción de conocimiento una vez procesados los datos.



Bibliografía

- BERGADANO, F. y GUNETTI, D. *Inductive Logic Programming: From Machine Learning to Software Engineering* The MIT Press, 1996
- DE RAEDT, L. (Ed.) *Advances in Inductive Logic Programming* IOS Press, 1996
- FLACH, P. *Simply Logical* John Wiley & Sons Ltd., 1994
- LAVRAČ, N. y DŽEROSKI, S. *Inductive Logic Programming Techniques and Applications* Ellis Horwood Ltd., 1994
- MITCHELL, T. M. *Machine Learning* McGraw–Hill, 1997
- MUGGLETON, S. (Ed.) *Inductive Logic Programming* Academic Press, 1992
- NIENHUYS–CHENG, S-H. y DE WOLF, R. *Foundations of Inductive Logic Programming* Springer–Verlag, 1997
- SHAPIRO, E.Y. *Algorithmic program debugging* MIT Press, 1983