Programación Lógica Inductiva

Aprendizaje Automático

Ingeniería Informática

Fernando Fernández Rebollo y Daniel Borrajo Millán

Grupo de Planificación y Aprendizaje (PLG)
Departamento de Informática
Escuela Politécnica Superior
Universidad Carlos III de Madrid

27 de febrero de 2009

En Esta Sección:

Motivación

- ¿Qué ocurre cuando queremos extraer conocimiento que:
  - no esté representado como atributo-valor o
  - tenga una relación entre los datos que sea difícil expresarla como preguntas por valores de atributos?

A_1	A_2	Clase
0	0	+
1	0	-
2	2	+
0	1	-
2	0	-
1	1	+
2	1	-
0	2	-

¿Qué ocurre?

A_1	A_2	Class
0	0	+
1	0	-
2	2	+
0	1	-
2	0	-
1	1	+
2	1	-
0	2	-

If A_1=0 Y A_2=0 Then +
If A_1=0 Y A_2=1 Then -
If A_1=0 Y A_2=2 Then -
If A_1=1 Y A_2=0 Then -
If A_1=1 Y A_2=1 Then +
If A_1=1 Y A_2=2 Then -
If A_1=2 Y A_2=0 Then -
If A_1=2 Y A_2=1 Then -
If A_1=2 Y A_2=2 Then +

Solución con lógica de predicados

A_1	A_2	Clase
0	0	+
1	0	-
2	2	+
0	1	-
2	0	-
1	1	+
2	1	-
0	2	-

If A_1=0 Y A_2=0 Then +
If A_1=0 Y A_2=1 Then -
If A_1=0 Y A_2=2 Then -
If A_1=1 Y A_2=0 Then -
If A_1=1 Y A_2=1 Then +
If A_1=1 Y A_2=2 Then +
If A_1=2 Y A_2=0 Then -
If A_1=2 Y A_2=1 Then -
If A_1=2 Y A_2=2 Then +

If A_1=A_2 Then +

Análisis de Datos Relacional

Tabla Clientes					
ID	Género	Edad	Ingresos	Gastos	Gran_Cliente
c1	hombre	30	214000	18800	Sí
c2	mujer	19	139000	15100	Sí
c3	hombre	55	50000	12400	No
c4	mujer	48	26000	8600	No
c5	hombre	63	191000	28100	Sí
c6	hombre	63	114000	20400	Sí
c7	hombre	58	38000	11800	No
c8	hombre	22	39000	5700	No
c9	mujer	65	12000	23500	Sí
c10	hombre	30	30000	9000	Sí

Tabla Casado Con	
Miembro 1	Miembro 2
c1	c2
c3	c4
c5	c9
c10	c6

Soluciones

- Regla Proposicional:
  IF ($ingresos > 100000$) THEN $Gran\_Cliente = si$.
- Reglas Relacionales:
  Gran_Cliente(C1,Edad1,Ingresos1,Gastos1) $\leftarrow$
  Cliente(C1,Edad1,Ingresos1,Gastos1) $\wedge$ $Ingresos1 > 100000$.
  Gran_Cliente(C1,Edad1,Ingresos1,Gastos1) $\leftarrow$
  Cliente(C2,Edad2,Ingresos2,Gastos2) $\wedge$casado_con(C1,C2) $\wedge$
  $Ingresos2 > 100000$.

Soluciones

- Regla Proposicional:
  IF ($ingresos > 100000$) THEN $Gran\_Cliente = si$.
- Reglas Relacionales:
  Gran_Cliente(C1,Edad1,Ingresos1,Gastos1) $\leftarrow$
  Cliente(C1,Edad1,Ingresos1,Gastos1) $\wedge$ $Ingresos1 > 100000$.
  Gran_Cliente(C1,Edad1,Ingresos1,Gastos1) $\leftarrow$
  Cliente(C2,Edad2,Ingresos2,Gastos2) $\wedge$casado_con(C1,C2) $\wedge$
  $Ingresos2 > 100000$.

Soluciones

- Regla Proposicional:
  IF ($ingresos > 100000$) THEN $Gran\_Cliente = si$.
- Reglas Relacionales:
  Gran_Cliente(C1,Edad1,Ingresos1,Gastos1) $\leftarrow$
  Cliente(C1,Edad1,Ingresos1,Gastos1) $\wedge$ $Ingresos1 > 100000$.
  Gran_Cliente(C1,Edad1,Ingresos1,Gastos1) $\leftarrow$
  Cliente(C2,Edad2,Ingresos2,Gastos2) $\wedge$casado_con(C1,C2) $\wedge$
  $Ingresos2 > 100000$.

Proposicionalización

- Agregación
  - Sumar todos los ingresos (propios y de la pareja)
  - Pero se pierde información sobre quién cobra qué
- Introducir nuevos atributos en la tabla original
  - ¿Qué ocurre con las relaciones n-m?

Elementos de representación

- Términos: constantes (a), variables (X), funciones ($f(X, Y)$)

  pepe, juan, *Cliente*, cliente-de(X, Y)

- Fórmulas atómicas: predicados definidos sobre términos

  tipo-cliente(X,bueno)

- Fórmulas bien formadas (*wff*): fórmulas atómicas unidas por conectivas ($\wedge, \vee, \rightarrow, \neg$) y cuantificadas ($\forall, \exists$)

$$\forall X, \exists Z$$
$$\text{cliente}(X) \wedge \text{compra}(X, Z) \wedge \text{caro}(Z) \rightarrow \text{tipo-cliente}(X,\text{bueno})$$

Más elementos de representación

- Cláusula de Horn: implicación lógica

```
abuelo(X,Y) :- padre(X,Z), padre(Z,Y).
```

$\forall X, Y \exists Z$ `padre(`X, Z`)`$\wedge$`padre(`Z, Y`)`$\rightarrow$`abuelo(`X, Y`)`

- Regla de Horn: conjunto de cláusulas de Horn

```
abuelo(X,Y) :- padre(X,Z), madre(Z,Y).
abuelo(X,Y) :- padre(X,Z), padre(Z,Y).
```

- Definición implícita (o intensional) de un concepto

  Ejemplo anterior de `abuelo`

- Definición explícita (o extensional) de un concepto

`abuelo(`X, Y`)={(pepe,juan),(pepe,ana),...,(luis,javier)}`

Tarea de aprendizaje

- Entradas:
  - E, conjunto de ejemplos positivos y negativos del concepto meta

    $\texttt{abuelo}(X,Y)^{+}=\{\texttt{(pepe,juan)},\texttt{(pepe,ana)},\texttt{(pepe,andrés)}\}$
    $\texttt{abuelo}(X,Y)^{-}=\{\texttt{(pepe,luis)},\ldots,\texttt{(ana,andrés)}\}$
  - D, conocimiento del dominio, que puede ser extensional o intensional

    $\texttt{padre}(X,Y)=\{\texttt{(pepe,luis)},\texttt{(pepe,ana)},\texttt{(luis,ana)},\texttt{(luis,juan)}\}$
    $\texttt{madre}(X,Y)=\{\texttt{(ana,andrés)}\}$

    o bien:

    $\texttt{progenitor}(X,Y)\ \texttt{:-}\ \texttt{padre}(X,Y).$
    $\texttt{progenitor}(X,Y)\ \texttt{:-}\ \texttt{madre}(X,Y).$

Tarea de aprendizaje

- Salidas: H, hipótesis tal que $D \wedge H \vdash E$ ($D \wedge H \vdash E^+$ y $D \wedge H \nvdash E^-$)

```
abuelo(X,Y) :- padre(X,Z),
               madre(Z,Y).
abuelo(X,Y) :- padre(X,Z),
               padre(Z,Y).
```

```
abuelo(X,Y) :- padre(X,Z),
               progenitor(Z,Y).
```

Algoritmo de FOIL

Función FOIL (E^+, E^-, P, D): *REGLA*
Hasta que $E^+ = \phi$:
  $N = E^-$
  CUERPO=ϕ
  Hasta que $N = \phi$
    L=`literal-máxima-ganancia`$(E^+, E^-, P, D, N, CUERPO)$
    $CUERPO = CUERPO, L$
    $N = N - \{e \in N \mid L \not\forall e\}$
  $REGLA = REGLA \cup [P :- CUERPO]$
  $E^+ = E^+ - \{e \in E^+ \mid CUERPO \vdash e\}$
Devolver *REGLA*

Heurística de FOIL

$$G(L) = k \times [I(n^+, n^-) - I(n_L^+, n_L^-)]$$

donde:

- $G(L)$ es la ganancia que se obtiene al añadir el literal L a la clausula
- k es el número de ejemplos positivos de E que cumplen L,
- n^+ es el número de ejemplos positivos cubiertos por la cláusula
- n^- es el número de ejemplos negativos cubiertos por la cláusula
- $$I(n^+, n^-) = -\log_2(\frac{n^+}{n^+ + n^-})$$
- n_L^+ es el número de ejemplos positivos cubiertos por la cláusula si se añade el literal L,
- n_L^- es el número de ejemplos negativos cubiertos por la cláusula si se añade el literal L,
- $$I(n_L^+, n_L^-) = -\log_2(\frac{n_L^+}{n_L^+ + n_L^-})$$

Tipos de literales

- $Q(V_1, V_2, \ldots, V_n)$: se cumple el literal Q para los V_i dados
- $\sim Q(V_1, V_2, \ldots, V_n)$: no se cumple el literal Q para los V_i dados
- $X_i = X_j$, $X_i \neq X_j$: una varible es igual/diferente que la otra
- $X_i = c$, $X_i \neq c$: una variable es igual/diferente que una constante del mismo tipo
- $X_i < X_j$, $X_i \geq X_j$: una varible es menor/mayor/igual que la otra
- $X_i < X_j$, $X_i \geq c$: una varible es menor/mayor/igual que una constante

donde

- X_i son variables que ya aparecieran en *CUERPO*,
- V_i son variables nuevas o ya existentes en *CUERPO* o en P, y
- Q es algún predicado de la teoría del dominio (P o alguno que esté descrito en D)

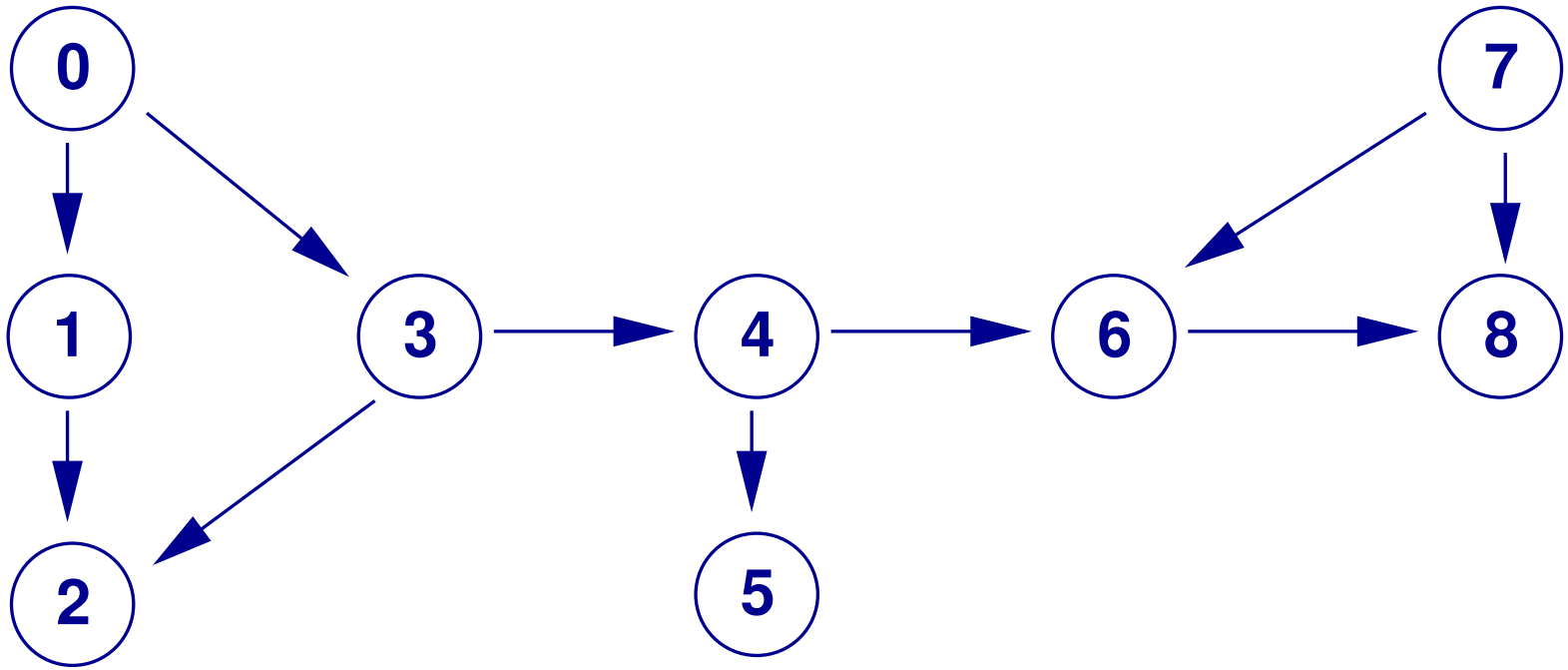
Otras restricciones

- Q debe tener al menos una variable ya existente;
- FOIL permite obtener definiciones recursivas (con algunas restricciones)

```
miembro(X,Z) :- formado-por(Z,X,Y).
miembro(X,Z) :- formado-por(Z,A,Y), miembro(X,Y).
```

- se puede hacer poda cuando la ganancia esté por debajo de un umbral
- no permite funciones ni constantes en las reglas aprendidas; y
- el conocimiento de dominio, D, debe estar definido extensionalmente.

Ejemplo



Ejemplo. Tarea

Dados:

E_0: `alcanzable`=[(0,1),(0,2),(0,3),(0,4),(0,5),(0,6),(0,8),(1,2),(3,2),(3,4),
(3,5),(3,6),(3,8),(4,5),(4,6),(4,8),(6,8),(7,6),(7,8)]
D: `conectado`=[(0,1),(0,3),(1,2),(3,2),(3,4),(4,5),(4,6),(6,8),(7,6),(7,8)]

Obtener: `alcanzable`.

Generación de ejemplos negativos (mundo cerrado):

E_0^-=[(0,0),(0,7),(1,0),(1,1),(1,3),(1,4),(1,5),(1,6),(1,7),(1,8),(2,0),(2,1),(2,2),(2,3),
(2,4),(2,5),(2,6),(2,7),(2,8),(3,0),(3,1),(3,3),(3,7),(4,0),(4,1),(4,2),(4,3),(4,4),
(4,7),(5,0),(5,1),(5,2),(5,3),(5,4),(5,5),(5,6),(5,7),(5,8),(6,0),(6,1),(6,2),(6,3),
(6,4),(6,5),(6,6),(6,7),(7,0),(7,1),(7,2),(7,3),(7,4),(7,5),(7,7),(8,0),(8,1),(8,2),
(8,3),(8,4),(8,5),(8,6),(8,7),(8,8)]

Ejemplo

P=`alcanzable`(X_1, X_2)

Posibles literales a añadir:

`conectado`(X_1, X_2)	`conectado`(X_2, X_1)	`conectado`(X_1, X_3)
`conectado`(X_3, X_2)	`conectado`(X_3, X_1)	`conectado`(X_2, X_3)
`alcanzable`(X_1, X_3)	`alcanzable`(X_2, X_3)	`alcanzable`(X_3, X_1)
`alcanzable`(X_3, X_2)	`alcanzable`$(X_2, X_1$	
$X_1 = X_2$	$X_1 \neq X_2$	

Ejemplo. Evaluación de heurística para conectado(X_1, X_2)

- n^+=19, número de ejemplos positivos iniciales
- n^-=62, número de ejemplos negativos iniciales
- k=10, número de ejemplos para los que si se cumple *conectado*(X_1, X_2), se cumple *alcanzable*(X_1, X_2)
- n_L^+=10, cada una de las 10 tuplas de conectado anteriores, genera un ejemplo positivo en el siguiente conjunto de tuplas
- n_L^-=0, no hay ninguna tupla de conectado que genere ejemplos negativos.

$$G(L_i) = k \times [I(n^+, n^-) - I(n_L^+, n_L^-)]$$

$$10 \times [-log_2(\frac{19}{19+62}) + log_2(\frac{10}{10+0})] = 20{,}9$$

Ejemplo. Primera cláusula

$$\texttt{alcanzable}(X_1, X_2) \texttt{ :- conectado}(X_1, X_2).$$

E_1^+: alcanzable=[(0,2),(0,4),(0,5),(0,6),(0,8),(3,5),(3,6),(3,8),(4,8)]
$E_1^- = E_0^-$

Ejemplo. Evaluación de heurística para conectado(X_1, X_3)

- n^+=9, los ejemplos positivos que quedan
- n^-=62, todos los ejemplos negativos iniciales
- k=9, `alcanzable`(X_1, X_2) `:-` `conectado`(X_1, X_3) describe a todos los ejemplos positivos pendientes
- n_L^+=18, ya que ahora hay que considerar todas las tuplas formadas por tres valores (hay tres variables en la cláusula que se generaría con este literal, X_1, X_2 y X_3):

  E_L^+: [(0,2,1),(0,2,3),(0,4,1),(0,4,3),(0,5,1),(0,5,3),(0,6,1),(0,6,3),(0,8,1),(0,8,3), (3,5,2),(3,5,4),(3,6,2),(3,6,4),(3,8,2),(3,8,4),(4,8,2),(4,8,4)]

Ejemplo. Segunda cláusula

n_L^-=54 ya que, al hacer lo mismo que con los positivos, hay que tener en cuenta que de los ejemplos negativos habrá que borrar los que tengan como primer argumento 2, 5, y 8:

$E_2^- :=$ [(0,0,1),(0,0,3),(0,7,1),(0,7,3),(1,0,2),(1,1,2),(1,3,2),(1,4,2),(1,5,2),(1,6,2),
(1,7,2),(1,8,2),(3,0,2),(3,0,4),(3,1,2),(3,1,4),(3,3,2),(3,3,4),(3,7,2),(3,7,4),
(4,0,5),(4,0,6),(4,1,5),(4,1,6),(4,2,5),(4,2,6),(4,3,5),(4,3,6),(4,4,5),(4,4,6),
(4,7,5),(4,7,6),(6,0,8),(6,1,8),(6,2,8),(6,3,8),(6,4,8),(6,5,8),(6,6,8),(6,7,8),
(7,0,6),(7,0,8),(7,1,6),(7,1,8),(7,2,6),(7,2,8),(7,3,6),(7,3,8),(7,4,6),(7,4,8),
(7,5,6),(7,5,8),(7,7,6),(7,7,8)]

$$G(L) = k \times [I(n^+, n^-) - I(n_L^+, n_L^-)] =$$

$$9 \times [-log_2(\frac{9}{9+62}) + log_2(\frac{18}{18+54})] = 8{,}8$$

Ejemplo. Solución

- Se elige $\texttt{conectado}(X_1, X_3)$:

  $$\texttt{alcanzable}(X_1, X_2) \texttt{ :- } \texttt{conectado}(X_1, X_3)$$

- Al generar las siguientes posibilidades, el mejor literal es: $\texttt{alcanzable}(X_3, X_2)$
- Esto deja el conjunto $E^+ = \phi$, por lo que la solución es:

  $\texttt{alcanzable}(X_1, X_2) \texttt{ :- } \texttt{conectado}(X_1, X_2).$
  $\texttt{alcanzable}(X_1, X_2) \texttt{ :- } \texttt{conectado}(X_1, X_3), \texttt{alcanzable}(X_3, X_2).$

Sesgo o bias

- Espacio de búsqueda mucho más amplio que con representación atributo-valor
- Retringir:
  - Lenguaje en las cláusulas: no permitir funciones ni constantes, o que en cada nuevo literal aparezca al menos una variable utilizada anteriormente
  - Heurísticas de búsqueda: definida por la función de ganancia (similar a ID3)