

TEMA 3 APRENDIZAJE DEDUCTIVO

Francisco José Ribadas Pena, Santiago Fernández Lanza

Modelos de Razonamiento y Aprendizaje

5º Informática

ribadas@uvigo.es, sflanza@uvigo.es

25 de febrero de 2013

- 3.1 Introducción
 - Aprendizaje basado en la explicación
 - Características, Funcionamiento e Inconvenientes
- 3.2 Generalización Basada en la Explicación (EBG)
 - Funcionamiento
 - Ejemplo
- 3.3 Aprendizaje de Macro-Operadores
 - Funcionamiento
 - Ejemplo

3.1 Introducción

- Otros nombres: aprendizaje analítico, aprendizaje basado en la explicación (EBL: *explanation based learning*)
- Sigue el esquema del razonamiento deductivo:

Conocimiento General
Conjunto de Premisas

Conclusiones

- **IDEA:** Uso del razonamiento deductivo para generar nuevo conocimiento que “mejore” el anterior.
 - Se construyen nuevas hipótesis más eficientes a partir del estudio (análisis) del funcionamiento de las actuales sobre ejemplos concretos.
Deducción vs. construcción estadística
 - Orientado a trabajar sobre hipótesis representadas en lógica formal, pero la misma idea es aplicable sobre otros tipos de representaciones.

3.1.1 Aprendizaje Basado en las Explicaciones (EBL)

- Métodos de aprendizaje en los que se emplea una secuencia de inferencias deductivas en las que se usan hechos o reglas ya conocidos para generar otros nuevos
 - Normalmente explicación \equiv demostración lógica
 - los nuevos hechos/reglas se derivan lógicamente de los conocidos
 - En general, es aplicable a cualquier procedimiento/razonamiento con pasos bien definidos

■ **Elementos:**

Teoría del dominio (Q): Conjunto de conocimiento previo sobre el problema que tratamos

- Se supone:

completa disponemos de todo el conocimiento necesario

todo ejemplo positivo puede ser deducido del conocimiento Q

$$\forall e^+, \text{ ejemplo positivo en } E, Q \models e^+$$

consistente el conocimiento no es incoherente

a partir de Q no se pueden deducir 2 ejemplos contradictorios

Concepto objetivo (P): Hipótesis que deseamos aprender (**mejorar**)

ya está definida en la teoría del dominio, Q , inicial

Ejemplo/s (E): instancia/s positiva/s del concepto objetivo P

Criterio de operatividad (O): indica si las nuevas hipótesis son

“útiles” o no

- **Objetivo:** Partiendo de Q , P y E queremos construir una nueva Teoría del Dominio Q' reformulando el conocimiento que se tenía inicialmente a partir del análisis de la explicación/demostración que la Q inicial hace de los ejemplos positivos de E .

- el conocimiento aprendido se añade al inicial
- los casos donde sea aplicable el conocimiento adquirido serán usados, reduciendo el tiempo de demostración
- para el resto de situaciones se usa el conocimiento inicial

Se pretende que esa nueva Teoría del Dominio Q' sea más eficiente que la inicial, de acuerdo al criterio de operatividad O (\equiv eficiencia).

- Idea general: nueva hipótesis se considera operativa si permite aumentar la eficiencia en un número razonablemente alto de situaciones
 - Contraposición entre operatividad (eficiencia) vs. generalidad
- En EBL basado en representaciones lógicas la nueva hipótesis aprendida es consecuencia lógica del conocimiento previo (Teoría del Dominio).

La diferencia es que ofrece una definición más directa y eficiente del concepto objetivo P

- (1) $Q \vdash E$,
- (2) mediante el análisis y generalización de esa demostración se define P
- (3) resultando que: $Q \models P \models E$

3.1.2 Características generales métodos EBL

- Requiere de conocimiento previo del dominio
 - Supone que se dispone de una Teoría del Dominio completa y consistente
 - Problema: en la práctica exige un alto grado de conocimiento previo
- Basta un ejemplo para poder analizarlo y generalizar correctamente
- Se obtiene una nueva hipótesis deducida a partir de la existente
 - no se incorpora conocimiento realmente nuevo
 - asegura que el conocimiento incorporado es siempre correcto

Diferencias con aprendizaje inductivo (AI)

AI: basado en esquemas de razonamiento inductivo
(hipótesis respaldadas por *inferencia estadística*)

EBL: basado en esquemas de razonamiento deductivo
(hipótesis respaldadas por *deducción lógica*)

AI: generaliza a partir de ejemplos de entrenamiento

EBL: mejora eficiencia de la hipótesis a partir de ejemplos,
pero no genera conocimiento nuevo

AI: no se dispone (excepto Prog. Lógica Inductiva) de conocim. previo

EBL: exige "mucho" conocimiento previo

AI: requiere un número relativamente alto de ejemplos de entrenamiento
para ofrecer conclusiones con suficiente fundamento estadístico

EBL: basta un núm. limitado de ejemplos (al menos 1) para generalizar
y mejorar la hipótesis

AI: puede generar hipótesis incorrectas (con poca base estadística o
válida sólo para ejemplos de entrenamiento [sobreadaptación])

EBL: garantiza la corrección de las hipótesis construidas (son consecuencia lógica)

3.1.3 Esquema de Funcionamiento General

Proceso en dos fases:

(1) Explicación. Dado un ejemplo de entrenamiento e , se usa la Teoría del Dominio, Q , actual para construir una explicación/demostración para confirmar que se trata de un ejemplo verdadero del Concepto Objetivo P .

- Se construye una “traza” del proceso de explicación/demostración
- Caso general: se consideran todos los caminos de demostración intentados (tanto fracasados como no)

(2) Generalización. Análisis de la “traza” de la explicación/demostración del ejemplo para identificar posible conocimiento útil.

Objetivo: Generalizar esa explicación/demostración concreta mediante un proceso de regresión.

- Básicamente: sustitución de constantes por variables, preservando validez lógica

Se pretende crear una nueva hipótesis deducida del conocimiento previo que permita demostrar ese ejemplo (y otros parecidos) de forma más eficiente (menos pasos)

3.1.4 Inconvenientes métodos EBL

Principal problema: exige especificar completamente el dominio

Problemas de las hipótesis creadas:

- Derivados de problemas con el conocimiento previo (Teoría del dominio)
 - Pueden ser *incompletas*: Si el conocimiento previo no es capaz de cubrir todos los posibles ejemplos, el generado tampoco será capaz
 - Pueden ser *incorrectas*: El sistema falla en la clasificación de los ejemplos de entrenamiento (conocim. previo es incompleto/incorrecto)
 - Pueden ser *incosistentes*: El sistema puede acabar por demostrar el concepto objetivo y su contrario
- Derivados del conocimiento aprendido
 - Crecimiento de la base de conocimiento: Incorporar sistemáticamente nuevo conocimiento incrementa el tamaño de la Base de Reglas
 - Implica aumento del factor de ramificación en búsquedas / demostraciones
 - Degradación de la Teoría del Dominio:
 - Las nuevas reglas tienden a ser más complejas que las iniciales
 - Pueden incorporar reglas poco útiles
 - ◇ derivadas de ejemplos muy concretos
 - ◇ con baja frecuencia de aplicación

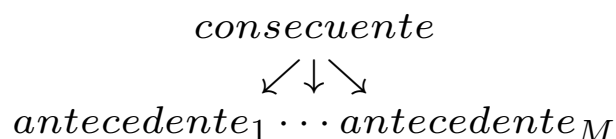
3.2 Generalización Basada en la Explicación (EBG)

EBG: *Explanation Based Generalization*, (Tom Mitchel, 1986)

- Tipo concreto de método EBL
- Aplicable sobre demostraciones de programas Prolog.

Funcionamiento (2 pasos)

1. Demostración de que el ejemplo de entrenamiento es una instancia verdadera del concepto objetivo (P), empleando la Teoría del Dominio (Q)
 - La **traza** de la explicación/demostración considera para el paso de generalización sólo el camino de demostración que lleva a la solución.
 - Traza con estructura de árbol.
 - raíz única: ejemplo demostrado
 - nodos + arcos: reglas usadas en la demostración



- traza estructurada en niveles, identificando la regla y la sustitución empleada en cada paso

2. Generalización de la demostración

- Generalización del único camino hacia la demostración del ejemplo
 - Diferencia: El EBL clásico $\left\{ \begin{array}{l} \text{puede estudiar todos los caminos} \\ \text{(de éxito ó fracaso)} \\ \text{genera 1 ó más reglas} \end{array} \right.$
- Se recorre la **traza** nivel a nivel empezando por la raíz.
Se construye la sustitución más general que garantiza la demostración del ejemplo actual (*proceso de regresión*)
 - Parte de las constantes que aparecen en el árbol de demostración (traza) que serán reemplazadas por variables.
 - Se trata de encontrar el conjunto más general de hechos iniciales A , para los cuales el ejemplo E sería consecuencia lógica [$A \models E$]
- La explicación (traza) así generalizada dará lugar a una nueva regla.

CONSECUENTE \equiv raíz del árbol generalizado

ANTECEDENTES \equiv conjunción (AND) de los nodos hoja

Método de análisis (regresión) en EBG

Procedimiento para generar la nueva regla:

- Dividir árbol de demostración en “niveles”
- Aplicar el siguiente procedimiento nivel a nivel, comenzando en la raíz:
 - FRONTERA: conjunto de literales (hojas) en el nivel actual
 - REGLA \equiv CABEZA :- CUERPO (*regla aplicada en el paso actual*)
 - ÁTOMO: literal de FRONTERA correspondiente a CABEZA
 - $\theta_{c,i}$: sustitución que fue usada en el i -ésimo paso de la demostración para unificar CABEZA con el literal que le correspondiera
 - Construcción de NUEVA_FRONTERA:
 1. Encontrar una sustitución $\theta_{c,a}$ que sea el unificador más general (u.m.g.) de CABEZA con ÁTOMO que verifique que exista otra sustitución θ_{aux} (no vacía) tal que:

$$sust(\theta_{aux}, sust(\theta_{c,a}, CABEZA)) = sust(\theta_{c,i}, CABEZA)$$

Nota: $\theta_{c,a}$ es una especie de “sustitución inversa” de la usada en la demostración.

2. Crear NUEVA_FRONTERA (*aplicar la sustitución del paso anterior*)

$$NUEVA_FRONTERA := sust(\theta_{c,a}, [FRONTERA - CABEZA + CUERPO])$$

- La regla final tendrá el siguiente aspecto:

$$REGLA_FINAL \equiv \text{OBJETIVO} : -\text{FRONTERA_FINAL}$$

Ejemplo EBG:

Predicado objetivo (P): `apilable(X,Y)`

Teoria del dominio (Q):

```
apilable(X,Y):-ligero(X,Y).                apilable(X,Y):-not(fragil(Y)).
ligero(X,Y):-peso(X,PX),peso(Y,PY),menor(PX,PY).  fragil(X):-material(X,porcelana).
peso(X,10):-IS_A(X,mesa).                    fragil(X):-material(X,cristal).
peso(X,PX):-volumen(X,VX),densidad(X,DX),igual(PX,VX*DX).
```

```
IS_A(caja_roja, caja).    material(caja_roja, cartón).    volumen(caja_roja, 2).
IS_A(mesa_cocina, mesa). material(mesa_cocina, madera).  densidad(caja_roja, 0.6).
```

Ejemplo (E): `apilable(caja_roja, mesa_cocina)`

Demostración (traza)

```
                                apilable(caja_roja, mesa_cocina)
                                {X1/caja_roja, Y1/mesa_cocina} |
                                |
                                ligero(caja_roja, mesa_cocina)
                                {X2/caja_roja, Y2/mesa_cocina, PX2/1.2, PY2/10} |
                                -----|-----
                                peso(caja_roja, 1.2)    menor(1.2, 10)    peso(mesa_cocina, 10)
                                {X3/caja_roja, PX3/1.2,  |    {X4/mesa_cocina}
                                DX3/0.6, VX3/2}          |
                                -----|-----
                                volumen(caja_roja, 2) densidad(caja_roja, 0.6) igual(1.2, 2*0.6) IS_A(mesa_cocina, mesa)
```

Proceso de análisis/regresión

paso 1 | FRONTERA: $\text{apilable}(\text{caja_roja}, \text{mesa_cocina})$
 REGLA: $\text{apilable}(X1, Y1) : \neg \text{ligero}(X1, Y1)$
 ATOMO: $\text{apilable}(\text{caja_roja}, \text{mesa_cocina})$
 $\theta_{c,1} : \{\text{X1}/\text{caja_roja}, \text{Y1}/\text{mesa_cocina}\}$

1. $\theta_{c,a} = \{\text{caja_roja}/X1, \text{mesa_cocina}/Y1\}$

$$\begin{aligned} \text{sust}(\theta_{c,1}, \text{apilable}(X1, Y1)) &= \text{apilable}(\text{caja_roja}, \text{mesa_cocina}) = \\ &= \text{sust}(\theta_{aux}, \text{sust}(\theta_{c,a}, \text{apilable}(X1, Y1))) \\ \text{con } \theta_{aux} &= \{\text{X1}/\text{caja_roja}, \text{Y1}/\text{mesa_cocina}\} \end{aligned}$$

2. $\text{NUEVA_FRONTERA} = \text{ligero}(X1, Y1)$

$$\equiv \text{sust}(\theta_{c,a}, \text{apilable}(\text{caja_roja}, \text{mesa_cocina}) + \text{ligero}(X1, Y1) - \text{apilable}(X1, Y1))$$

paso 2 | FRONTERA: $\text{ligero}(X1, Y1)$
 REGLA: $\text{ligero}(X2, Y2) : \neg \text{peso}(X2, PX2), \text{peso}(Y2, PY2), \text{menor}(PX2, PY2)$
 ATOMO: $\text{ligero}(X1, Y1)$
 $\theta_{c,2} : \{\text{X2}/\text{caja_roja}, \text{Y2}/\text{mesa_cocina}, \text{PX2}/1.2, \text{PY2}/10\}$

1. $\theta_{c,a} = \{\text{X1}/X2, \text{Y1}/Y2\}$

$$\begin{aligned} \text{sust}(\theta_{c,2}, \text{ligero}(X2, Y2)) &= \text{ligero}(\text{caja_roja}, \text{mesa_cocina}) = \\ &= \text{sust}(\theta_{aux}, \text{sust}(\theta_{c,a}, \text{ligero}(X2, Y2))) \\ \text{con } \theta_{aux} &= \{\text{X2}/\text{caja_roja}, \text{Y2}/\text{mesa_cocina}\} \end{aligned}$$

2. $\text{NUEVA_FRONTERA} = \text{peso}(X2, PX2), \text{peso}(Y2, PY2), \text{menor}(PX2, PY2)$

$$\equiv \text{sust}(\theta_{c,a}, \text{ligero}(X1, Y1) + \text{peso}(X2, PX2), \text{peso}(Y2, PY2), \text{menor}(PX2, PY2) - \text{ligero}(X2, Y2))$$

paso 3

FRONTERA: peso(X2,PX2), peso(Y2,PY2), menor(PX2,PY2)
 REGLA: peso(X3,PX3):-volumen(X3,VX3), densidad(X3,DX3), igual(PX3,VX3*DX3)
 ATOMO: peso(X2,PX2)
 $\theta_{c,3}$: {X3/caja_roja, PX3/1.2, DX3/0.6, VX3/2}

$$1. \theta_{c,a} = \{X2/X3, PX2/PX3\}$$

$$\begin{aligned} sust(\theta_{c,3}, peso(X3, PX3)) &= peso(caja_roja, 1,2) = \\ &= sust(\theta_{aux}, sust(\theta_{c,a}, peso(X3, PX3))) \\ \text{con } \theta_{aux} &= \{X3/caja_roja, PX3/1.2\} \end{aligned}$$

$$2. \text{NUEVA_FRONTERA} =$$

$$\begin{aligned} &\text{volumen}(X3, VX3), \text{densidad}(X3, DX3), \text{igual}(PX3, VX3*DX3), \text{peso}(Y2, PY2), \text{menor}(PX3, PY2) \\ \equiv &sust(\theta_{c,a}, peso(X2, PX2), peso(Y2, PY2), menor(PX2, PY2) + \\ &+ \text{volumen}(X3, VX3), \text{densidad}(X3, DX3), \text{igual}(PX3, VX3*DX3) - \\ &- peso(X3, PX3)) \end{aligned}$$

paso 4

FRONTERA:
 volumen(X3,VX3), densidad(X3,DX3), igual(PX3,VX3*DX3), peso(Y2,PY2), menor(PX3,PY2)
 REGLA: peso(X4,10):-IS_A(X4,mesa)
 ATOMO: peso(Y2,PY2)
 $\theta_{c,4}$: {X4/mesa_cocina}

$$1. \theta_{c,a} = \{Y2/X4, PY2/10\}$$

$$\begin{aligned} sust(\theta_{c,4}, peso(X4, 10)) &= peso(mesa_cocina, 10) = \\ &= sust(\theta_{aux}, sust(\theta_{c,a}, peso(X4, 10))) \\ \text{con } \theta_{aux} &= \{X4/mesa_cocina\} \end{aligned}$$

$$2. \text{NUEVA_FRONTERA} =$$

$$\begin{aligned} &\text{volumen}(X3, VX3), \text{densidad}(X3, DX3), \text{igual}(PX3, VX3*DX3), \text{IS_A}(X4, mesa), \text{menor}(PX3, 10) \\ \equiv &sust(\theta_{c,a}, \text{volumen}(X3, VX3), \text{densidad}(X3, DX3), \\ &\text{igual}(PX3, VX3 * DX3), peso(Y2, PY2), menor(PX3, PY2) + \\ &+ \text{IS_A}(X4, mesa) - peso(X4, 10)) \end{aligned}$$

Nueva regla aprendida:

- Combinar las sustituciones $\theta_{c,a}$ parciales

$$\theta = \left\{ \begin{array}{l} caja_roja/X1, mesa_cocina/Y1, X1/X2, Y1/Y2, \\ X2/X3, PX2/PX3, Y2/X4, PY2/10 \end{array} \right\}$$

- **Antecedente:** aplicar la sustitución resultante sobre el ejemplo inicial
apilable(caja_roja,mesa_cocina)
- Regla resultante:

apilable(X3,X4) :- volumen(X3,VX3),densidad(X3,DX3),
igual(PX3,VX3*DX3),IS_A(X4,mesa),menor(PX3,10).

“Sobre las mesas se pueden apilar cajas que no pesen más de 10 kg.”

3.3 Aprendizaje de macro-operadores

Técnica de aprendizaje analítico/deductivo empleada en sistemas basados en búsquedas en espacio de estados.

Idea básica: Analizar las búsquedas realizadas para agrupar y generalizar el conjunto de operadores utilizados, dando lugar a un nuevo operador (*macro-operador*)

- tendrá sus propias precondiciones
(subconjunto de las precondiciones de los operadores empleados en la búsqueda)
- se ahorran pasos de búsqueda
- incrementa el factor de ramificación

Técnica usada originalmente en planificador STRIPS (Fikes & Nilsson, 71)

- Planificador basado en búsqueda en espacios de estados (espacio de planes)
- Aplicaciones en robótica
- El estado del sistema está representado mediante conjuntos de átomos lógicos (predicados o negación de predicados de lógica de 1er. orden)
- Operador STRIPS: representa posibles acciones
 - P: (*precondiciones*) conjunto de literales que describen cómo debe ser el estado del sistema para poder aplicar el operador
 - B: (*lista borrar*) lista de literales a eliminar de la representación del estado del sistema una vez ejecutado el operador
 - A: (*lista añadir*) lista de literales a añadir en la representación del estado del sistema una vez ejecutado el operador

Si el estado actual verifica P, el nuevo estado será el resultado de eliminar los átomos del conjunto B y añadir los del conjunto A.

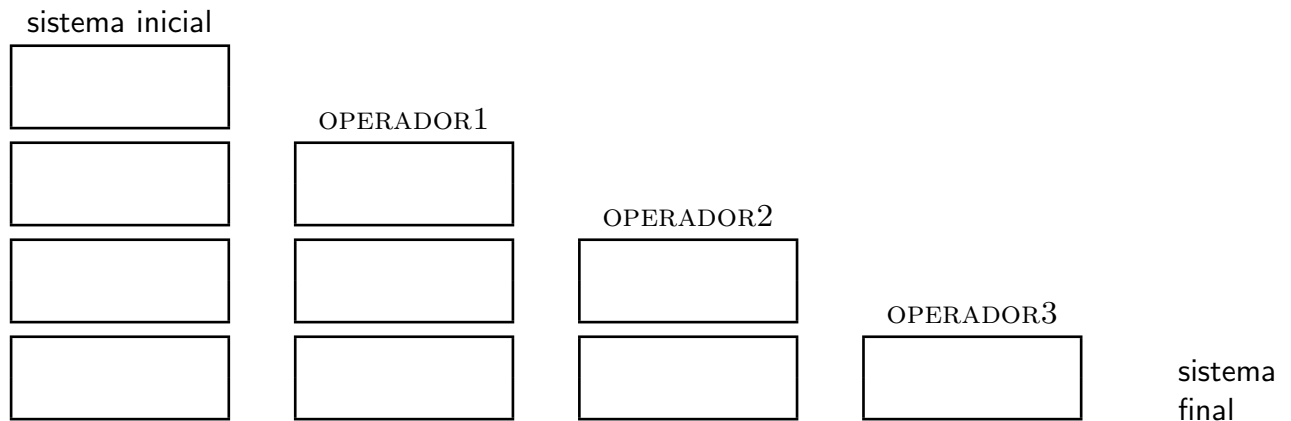
- ENTRADA STRIPS: descripción de la situación inicial del problema + descripción de las condiciones a cumplir por la situación objetivo
- SALIDA STRIPS: secuencia ordenada de acciones/operadores, que dan lugar a un estado ESTADO_FINAL verificando
“ESTADO_FINAL \models SISTEMA_OBJETIVO”

(a) Aprendizaje de macro-operadores en STRIPS

- Después de realizada la búsqueda
- Guardarla de forma generalizada (macro-operador)

Proceso

- Crear **tablas triangulares**: resumen del plan resultado de la búsqueda



- Parametrización de la tabla triangular
 - Sustituir algunas constantes por variables
 - Nuevo macro-operador aprendido
 - P: conjunto de literales de la primera columna
 - A: conjunto de literales de la última fila
 - B: idéntico a P

(b) Ejemplo de aprendizaje de macro-operadores

Minimundo de los bloques

- bloques etiquetados (a, b, c, \dots) que se pueden apilar unos sobre otros
- mesa sobre la que poner los bloques
- brazo para mover los bloques

Operadores:

op1. QUITAR(X, Y) $\left[\begin{array}{l} P: encima(X, Y), libre(X), brazo_libre \\ A: sujeto(X), libre(Y) \end{array} \right]$

op2. DEJAR(X) $\left[\begin{array}{l} P: sujeto(X) \\ A: en_mesa(X), libre(X), brazo_libre \end{array} \right]$

op3. LEVANTAR(X) $\left[\begin{array}{l} P: en_mesa(X), libre(X), brazo_libre \\ A: sujeto(X) \end{array} \right]$

op4. PONER(X, Y) $\left[\begin{array}{l} P: sujeto(X), libre(Y) \\ A: libre(X), encima(X, Y), brazo_libre \end{array} \right]$

Ejemplo: intercambio de los bloques (a y b) de la cima de una pila

situación inicial	situación final
encima(a, b)	encima(b, a)
encima(b, c)	encima(a, c)
libre(a)	libre(b)
en_mesa(c)	en_mesa(c)
brazo_libre	brazo_libre

Plan resultante: QUITAR(a, b), DEJAR(a), QUITAR(b, c), DEJAR(b),
LEVANTAR(a), PONER(a, c), LEVANTAR(b), PONER(b, a)