

4. APRENDIZAJE INDUCTIVO NO SUPERVISADO.

El aprendizaje inductivo no supervisado estudia el aprendizaje sin la ayuda del *maestro*; es decir, se aborda el aprendizaje sin supervisión, que trata de ordenar los ejemplos en una jerarquía según las regularidades en la distribución de los pares atributo-valor sin la utilización del atributo especial *clase*. Este es el proceder de los sistemas que realizan *clustering* conceptual. Otra posibilidad contemplada para estos sistemas es la de sintetizar conocimiento cualitativo o cuantitativo, objetivo de los sistemas que llevan a cabo tareas de *descubrimiento*. Este apartado comprende estos dos tipos de aprendizaje aunque del segundo de ellos nos limitaremos a especificar su proceder en un sentido muy general, ya que suelen ser muy específicos con resultados malos fuera del tema para el que fueron creados.

4.1. EL CLUSTERING CONCEPTUAL: LA FAMILIA CLUSTER Y COBWEB.

La noción de *clustering conceptual* la presenta Michalski [MICHALSKI et al. 83] para justificar la necesidad de un *clustering* cualitativo frente al ya muy estudiado *clustering* cuantitativo, o más concretamente taxonomía numérica, basado en la vecindad entre los elementos de la población.

Sin embargo, las réplicas a la necesidad de una definición de *clustering* conceptual no tardaron. Ante los ejemplos que propone Michalski, **figura 36**, para apoyar su argumentación, en los que la población posee una distribución geométrica que pasaría inadvertida para un tratamiento sencillo de las distancias entre los elementos, Hanson propone un tratamiento basado en centroides que sí captaría la *forma* de las distribuciones. A pesar de las críticas estos sistemas proporcionan buenos resultados por lo que hoy están consolidados como métodos de aprendizaje no supervisado.

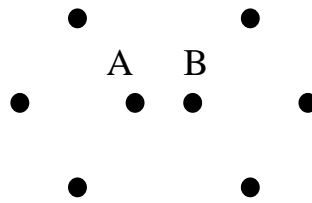


Figura 36: A y B se distribuirían en categorías distintas según se apliquen criterios basados en distancias o en distribuciones.

El sistema CLUSTER/2 de Michalski y Steep [MICHALSKI et al. 83], basado en la metodología *AQ star*, es uno de los elegidos para describir someramente como representante del *cluster conceptual*. Además de la descripción original del sistema que aparece en el texto de la referencia anterior, también hay en [BRISCOE et al., 96] un buen estudio acompañado de una discusión y de la evolución de este sistema hasta originar el nuevo sistema CLUSTER/S. La familia CLUSTER utiliza los conceptos de *semillas* y las función de valoración lexicográfica LEF heredados de la metodología *AQ star* del mismo autor.

COBWEB (Fisher 1987) es otro sistema que efectúa *clustering* conceptual, también jerárquico, aunque no con la concepción que Michalski y Steep asociaron al término. Resulta interesante por la función probabilística -frecuencias relativas- que utilizan para medir la *bondad* de las categorías. La herencia de las propiedades de una categoría por los objetos de una categoría inferior, cuando las categorías están jerarquizadas como en este caso, es la base sobre la que se asienta COBWEB.

4.1.1. CLUSTER/2

- **Generalidades**

Es éste un sistema que realiza clustering conceptual y que pertenece a la familia de los que utilizan la metodología AQ estrella. El sistema obtiene *clusters* jerarquizados y definidos por formas normales conjuntivas; esto es, clasificación de objetos o sucesos consistentes en formas normales conjuntivas de expresiones que *envuelven* relaciones o atributos. El clustering se realiza de modo que se obtienen condiciones necesarias y suficientes para la pertenencia de cada elemento a su *cluster*. Permite el mismo tipo de variables que los sistemas AQ -cualitativas, cuantitativas, estructuradas (ordenadas y desordenadas)- y las relaciones o atributos que intervienen en la descripción de los *clusters* adoptan la siguiente forma: $[x \# R_i]$, donde x es una variable, $\#$ es un operador $=, ., \neq, \leq, \geq, <, >$ o el operador rango $(..)$ - y R_i es una lista de elementos del dominio de la variable x conectados por el operador disyunción \vee -, p.e. $[\text{color} = \text{rojo} \vee \text{azul}]$; $[\text{longitud} = 3]$, $[\text{peso} = 2 \dots 6]$.

A semejanza del sistema AQ y familia, CLUSTER/2 utiliza las estrellas formadas cada una a partir de una semilla, sea ésta e_0 , y constituidas por una generalización de dicha semilla que no tenga intersección con el conjunto de ejemplos E_0 , conjunto formado por todas las otras la semillas consideradas en esa iteración. También incluye el control de la generalización de las estrellas obtenidas y la restricción de los descriptores de las estrellas en aquellos casos en que se acumulasen muchos descriptores para determinar una estrella. Se utiliza la función LEF (*Lexicographic Evaluation Function*) para evaluar, ordenar y seleccionar descripciones de *clusters* alternativas.

- **El algoritmo CLUSTER/2**

C2-1 Determina las Semillas Iniciales

Selección de k semillas de entre los sucesos que será clasificados (el k puede ser optimizado por el propio algoritmo probando iterativamente con sucesivos valores de k).

C2-2 Construye las Estrellas

Para cada semilla e , construye una estrella reducida, con el conjunto de semillas restantes como los sucesos que deberán ser excluidos de la estrella en construcción. Esto es, generaliza la descripción de cada semilla frente a todas las demás. Una estrella estará constituida por todas las descripciones maximales o *complex* que generalizan al objeto e . frente a las demás semillas. Aplica operadores de generalización tales como encerrar valores en intervalos, eliminar una condición, y ascender en el árbol de generalización.

C2-3 Optimiza el Clustering

En este punto las estrellas tendrán intersecciones invariables. Se construye un *clustering* optimizado de acuerdo a una cierta función, eligiendo un *complex* de cada estrella y modificándolos. Asegurarse que los *complex* elegidos son disjuntos para lo que se aplicará un procedimiento definido denominado NID.

C2-4 ¿Termina?

1. Si ésta es la primera iteración, almacenar las descripciones del clustering.
2. En otro caso, almacenar únicamente la descripción del clustering si mejora el rendimiento del anterior de acuerdo a LEF.
3. Si después de un cierto número de iteraciones no mejora el clustering entonces terminar.

C2-5 Seleccionar Nuevas Semillas

1. Seleccionar k nuevas semillas, una de cada *complex*.
2. Si el clustering mejora, seleccionar sucesos *centrales*, esto es, sucesos que estén más cerca de los centros geométricos de los *complex*.

3. Si el clustering no mejora, seleccionar sucesos de los *bordes*, esto es, sucesos que estén alejados de los centros.

C2-6 Repetir

Repetir desde el paso C2-2.

- **Una extensión: CLUSTER/S**

CLUSTER/S extiende al CLUSTER/2 de varias maneras. Utiliza como lenguaje de representación de los objetos el lenguaje PAC *-Annotated Predicate Calculus-* que permite descripciones estructurales y le habilita para añadir nuevos predicados dentro de un conjunto predefinido de ellos. Se utiliza un red de objetivos *-Goal Dependency Network-* para guiar la búsqueda de descriptores relevantes en la construcción de las reglas. Permite la inclusión de conocimiento aritmético y expresiones lógicas en el conocimiento de fondo *-background knowledge*.

4.1.2. COBWEB

- **Generalidades**

Una de las principales motivaciones de la categorización de un conjunto de ejemplos, que básicamente supone la formación de conceptos, es la predicción de características de las categorías que heredarán sus subcategorías. Esta conjetura es la base de COBWEB. A semejanza de los humanos, COBWEB forma los conceptos por agrupación de objetos con atributos similares. Lo hace a pesar de que exista información irrelevante o incompleta y mediante un proceso incremental. Aunque fue definido por uno de sus autores, Fisher, como un clustering conceptual realmente no se atiene a la definición que dio Michalski, padre del término; sin embargo sigue su espíritu en cuanto mide la utilidad de las descripciones de la categoría. COBWEB representa los *clusters* como una distribución de probabilidad sobre el espacio de los valores de los atributos, generando un árbol de clasificación jerárquica en el que los nodos intermedios definen subconceptos. La versión original de este algoritmo admite solamente valores simbólicos para los atributos, mientras que versiones posteriores también los admiten numéricos.

- **La utilidad de una categoría**

El *bias* de COBWEB consiste en hallar un conjunto de *clusters* que maximice la *utilidad de la categoría*, una medida probabilística inventada para explicar el *clustering* y la formación de conceptos en los humanos.

Una categoría tiene una utilidad elevada si, dado el suceso e perteneciente a la clase C_k , se puede predecir el valor V_{ij} del atributo A_i en e con elevada precisión. Una categoría también tiene elevada utilidad si, dado el valor V_{ij} del atributo A_i de un suceso e , un miembro de C_k , es previsible con cierta seguridad la pertenencia de e a C_k . Estas medidas miden la *predecibilidad* y la *previsibilidad*. La *predecibilidad* puede ser definida como la probabilidad condicional de que un suceso tenga un cierto atributo dada la clase, $P(A_i = V_{ij} / C_k)$. El mayor de estos valores, el más predecible de los valores, es el de los miembros de la clase (alta similaridad entre los elementos de la clase).

Similarmente la *previsibilidad* puede definirse como la probabilidad condicional de que un suceso sea una instancia de una cierta clase, dado el valor de una atributo particular, $P(C_k / A_i = V_{ij})$. Un valor alto indica que pocos objetos en las otras clases comparten este valor, y el valor de mayor previsibilidad es el de los miembros de la clase (baja similaridad interclase).

Estas dos medidas, combinadas mediante la Regla de Bayes, proporcionan la función que evalúa la *utilidad de una categoría* utilizada por COBWEB:

$$\frac{\sum_{k=1}^K P(C_k) \sum_i \sum_j P(A_i = V_{ij} / C_k)^2 - \sum_i \sum_j P(A_i = V_{ij})^2}{K},$$

donde K es le número de clases que se discute.

Esa función puede interpretarse que proporciona *el incremento en el número esperado de valores de atributos que pueden ser adivinados correctamente, dado un conjunto de K clases, sobre el número esperado de adivinaciones correctas sin tal conocimiento.*

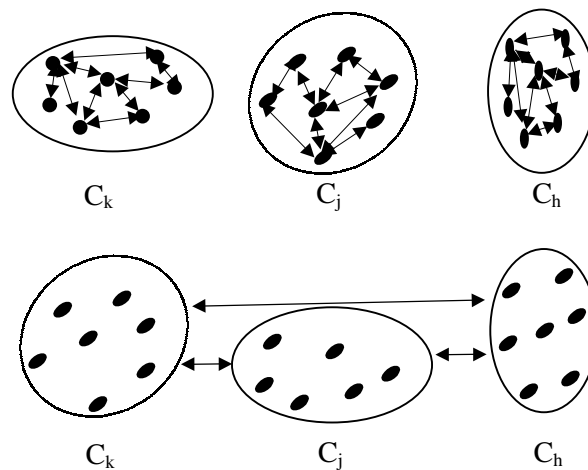


Figura 37: La *predicibilidad*, o similaridad dentro de una clase, y la *previsibilidad*, o similaridad interclases, configuran la *utilidad de una categoría*.

- **Entradas, representación de conceptos y salidas**

Las entradas de COBWEB es una colección de ejemplos o vectores de pares atributo valor, con la particularidad de la inexistencia del atributo clase empleado en el aprendizaje inductivo supervisado. En la siguiente tabla se muestra un caso de una familia de ejemplos donde el atributo nombre es meramente informativo y permitirá posteriormente identificar en la jerarquía resultante los conceptos obtenidos con alguno de estas nombres.

Nombre	Recubierto de	Nº cavidades del corazón	Temperatura del cuerpo	Fertilización
<i>Pez</i>	escamas	dos	sin regulación	externa
<i>Anfibio</i>	piel húmeda	tres	sin regulación	externa
<i>Mamífero</i>	pelo	cuatro	regulada	interna
<i>Ave</i>	plumas	cuatro	regulada	interna
<i>Reptil</i>	piel dura	cuatro imperfectas	sin regulación	interna

Figura 38: Colección de ejemplos que configuran una entrada de COBWEB

Incrementalmente se incorpora cada ejemplo a un árbol de clasificación, donde cada nodo es un concepto probabilístico que representa una clase de objetos. Los pares atributo valor del ejemplo de entrada son utilizados para calcular las diferentes probabilidades que se necesitan para calcular el valor de la función que mide la utilidad de la categoría en la que se ubicará el ejemplo de acuerdo a los diferentes supuestos que se pueden considerar y vienen reflejados en los *operadores: añadir una nueva clase, romper una ya existente*, etc. Es digno de subrayar la diferencia de los árboles de conceptos probabilísticos frente a los árboles de decisión y a las

redes de discriminación; en los primeros las etiquetas que etiquetan cada nodo describen conceptos probabilísticos.

El árbol resultante, más específicamente, cabe denominarse *organización probabilística y jerárquica de conceptos*. A continuación presentamos la jerarquía que proporciona COBWEB para el conjunto de ejemplos expuesto en la última tabla.

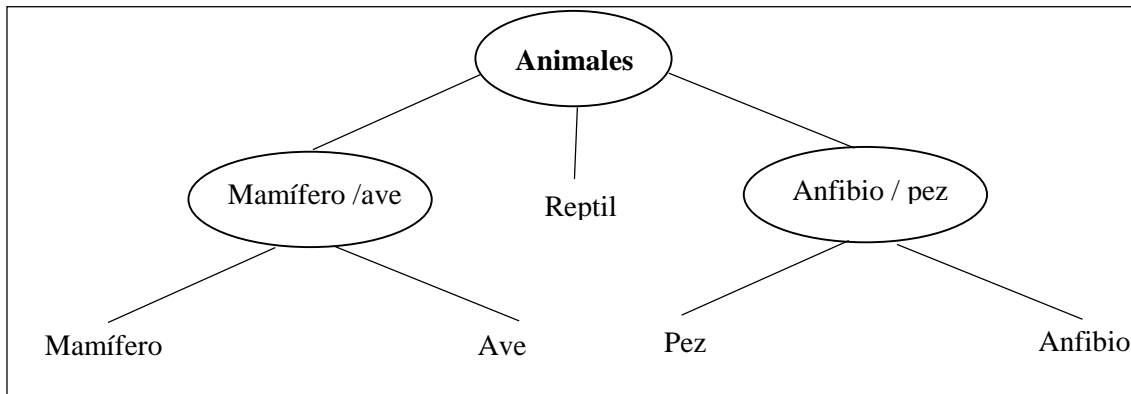


Figura 39: Árbol obtenido por COBWEB para los ejemplos descritos en la figura 38.

• Los operadores

Cuando COBWEB incorpora un nuevo ejemplo en el árbol de clasificación, desciende a lo largo del camino apropiado, actualizando las *cuentas* de cada nodo y llevando a cabo por medio de los *operadores*, una de las siguientes acciones:

- Añadir un nuevo ejemplo a un nodo ya existente (*operador de incorporación*).
- Crear una nueva clase (*operador de creación de una nueva disyunción*).
- Combinar dos clases en una sola (*operador de unión*).
- Dividir una clase existente en varias clases (*operador de división*).

La búsqueda, que se realiza en el espacio de conceptos, es por medio de un heurístico basado en el método de escalada gracias a los operadores de unión y división. La **figura 40** representa gráficamente las modificaciones del árbol en construcción según cada uno de los operadores citados.

• El algoritmo COBWEB

CW-1 Nuevo Ejemplo.

Lee un ejemplo *e*. Si no hay más ejemplos termina.

CW-2 Actualiza la raíz.

Actualiza el cálculo de la raíz.

CW-3 ¿La raíz es hoja?

Si la raíz es una hoja entonces expandir para acomodar el nuevo ejemplo y volver a CW-1.

CW-4 Avance del siguiente nivel

Aplicar la función de evaluación a varios casos para determinar el mejor lugar donde incorporar el ejemplo en el nivel actual de la jerarquía. Considerar únicamente los nodos actuales y sus hijos inmediatos permitiendo los siguientes casos:

- a) Añadir *e* a un nodo que existe (al mejor hijo)
- b) Crear un nuevo nodo conteniendo únicamente a *e*.
- c) Juntar las dos mejores clases e incorporar *e* en la nueva clase combinada.
- d) Dividir la mejor clase, reemplazando este nodo con sus hijos. Aplicar la función de evaluación para incorporar *e* en los nodos originados por la división.

CW-5 Incorporar la Instancia

Según se haya determinado en el paso anterior como mejor operación a realizar entre a),

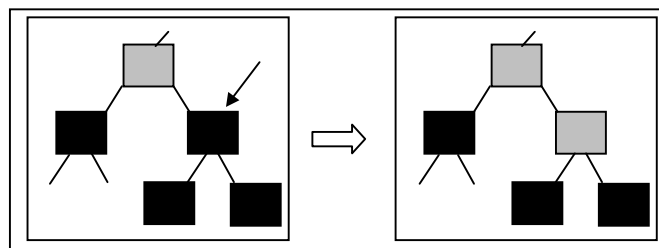
b), c) o d) entonces:

- aplicar el operador de crear nueva disyunción y volver al paso CW-1.
- Aplicar el operador incorporar. El nodo que recibe la incorporación pasa a ser la nueva raíz.
- Aplicar el operador unión. El nodo resultante es la nueva raíz.
- Aplicar el operador división. El mejor de los nodos resultantes de esa división pasa a ser la nueva raíz.

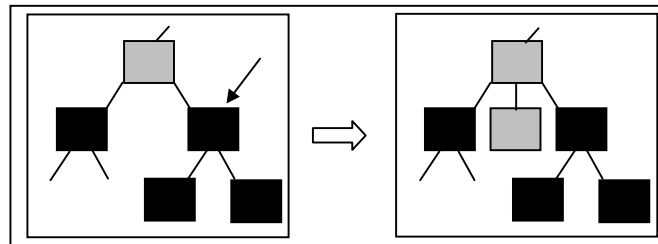
CW-6 Calculo recurrente del siguiente nivel

Repetir desde el paso CW-2 para calcular el siguiente nivel hacia abajo para incorporar el ejemplo en conceptos de mayor especificidad.

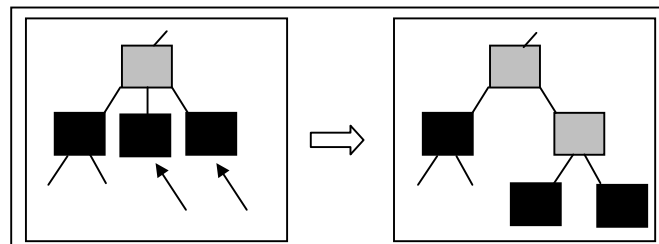
Parar cuando el concepto queda clasificado en un nodo simple en lo más bajo de la jerarquía o cuando se crea una nueva disyunción al incorporar el ejemplo en un cierto nivel.



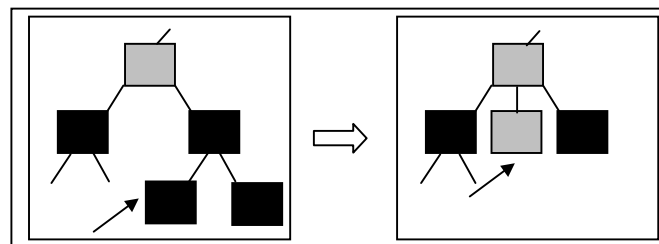
Operación *incorporación* del nuevo ejemplo a un nodo existente.



Operación *creación de una nueva disyunción* para incorporar el nuevo ejemplo al nuevo nodo.



Operación *unión* de nodos para formar un nuevo nodo en el que se incorpora el nuevo ejemplo.



Operación *división* de un nodo e incorporación del nuevo ejemplo en el mejor de los nodos resultantes

Figura 40: Las cuatro operaciones que realiza COBWEB en la construcción del árbol.

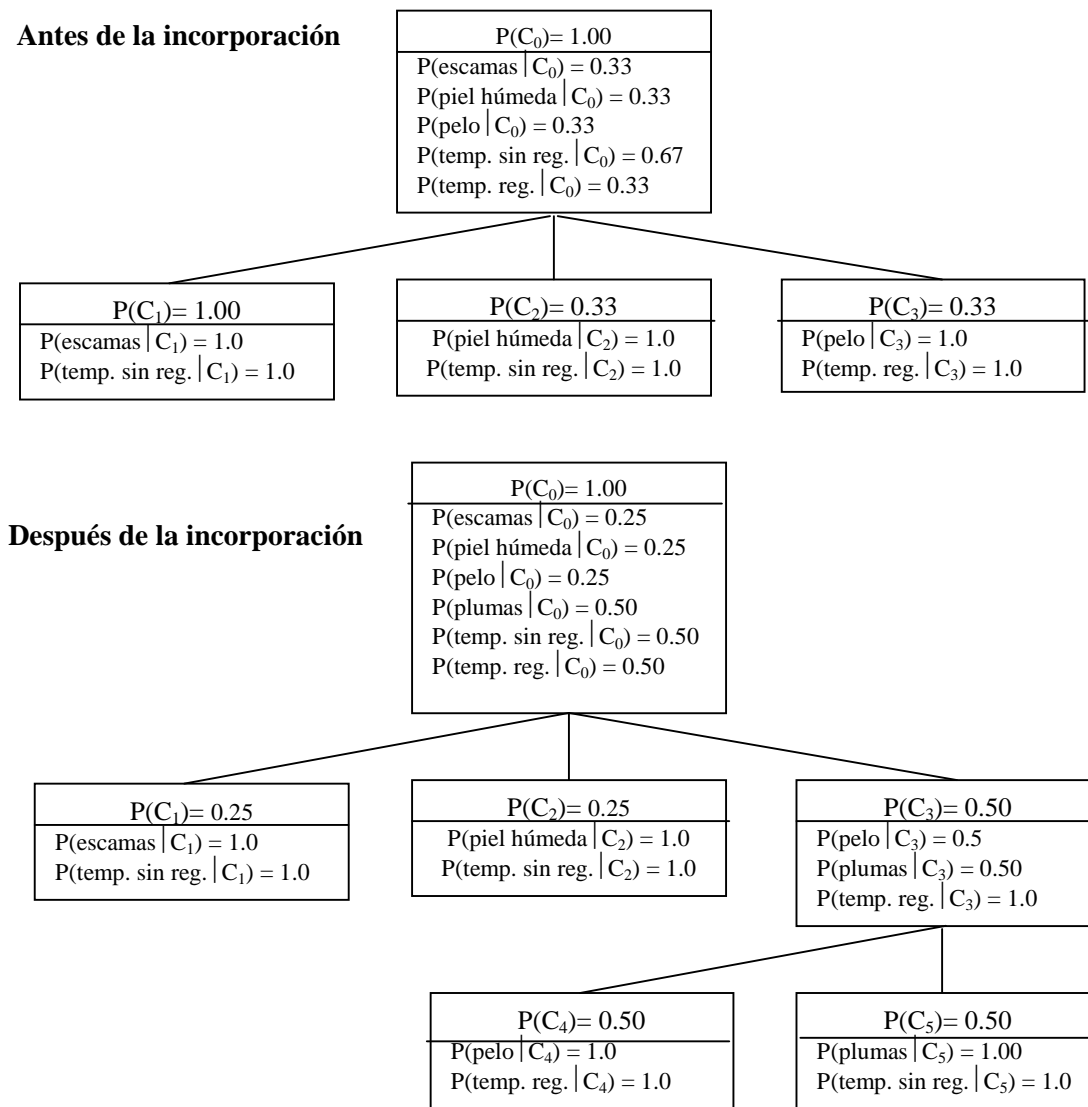


Figura 41: Incorporación del ejemplo ave al árbol *cluster*.

- **Propiedades de COBWEB**

Incremental, bastante económico y robusto. Presenta problemas frente al ruido, aunque su autor propone extensiones para mejorar ese aspecto.

4.2. APRENDIZAJE MEDIANTE DESCUBRIMIENTO.

- **Generalidades**

El Aprendizaje mediante descubrimiento utiliza un amplio rango de técnicas. En último lugar de esa escala, el descubrimiento empírico intenta hallar leyes generales que describan las observaciones, mientras que la formación de teorías más ambiciosas intentan ir más allá de la mera descripción e incluyen explicaciones detalladas para el descubrimiento de leyes. Muchos sistemas se han concentrado en el descubrimiento empírico, que utiliza algoritmos relativamente sencillos en entornos pobres de conocimiento. Trabajos recientes, sin embargo,

han intentado incluir métodos de aprendizaje de conocimiento intensivo con formas múltiples de razonamiento, tales como la inducción, la deducción y la analogía. Esto está más en la línea utilizada por los humanos para el descubrimiento que tienden a incorporar distintos razonamientos y técnicas de aprendizaje.

Entre los sistemas que existen se pueden citar como representantes a EURISCO, que descubre relaciones cualitativas, y BACON que descubre relaciones cuantitativas.

EURISCO extiende a AM (*Automathed Mathematician*) (Lenat 1989), y añade al *redescubrimiento* de conceptos de la teoría de conjuntos y de la teoría de números, el descubrimiento de heurísticos.

BACON (1987) *redescubrió* leyes cuantitativas famosas como la Ley de Ohm, la Ley de Boyle de los gases, las Leyes de Kepler sobre el movimiento de los planetas y la Ley de Galileo sobre la aceleración uniforme.

III. APRENDIZAJE DEDUCTIVO. SISTEMAS INTEGRADOS

En el aprendizaje inductivo, como ya se vio, trata de seleccionar en un espacio de hipótesis H y a partir del conocimiento de bajo nivel proporcionado por un conjunto de ejemplos de entrenamiento, una hipótesis h de H que sea consistente con los ejemplos de entrenamiento. El aprendizaje deductivo incluye asimismo el conjunto de hipótesis H de partida, y el conjunto de ejemplos de entrenamiento, más una entrada adicional que consiste en una teoría del dominio T (reglas). La salida que se busca es una hipótesis h de H que sea consistente con los ejemplos de entrenamiento y con la teoría del dominio.

- **Consideraremos en este capítulo dos tipos de aprendizaje basados en el razonamiento deductivo:**
 - Por un lado aquel que se basa en el razonamiento deductivo sobre un conocimiento a priori correcto y completo (aunque la carencia de estas propiedades los sistemas más evolucionados tratan de subsanarlas de diferentes maneras), y que trata de aumentar el conocimiento proporcionado por conjunto de ejemplos de partida. Se identifica este aprendizaje con el aprendizaje deductivo *puro*, aunque también se suele denominar *aprendizaje analítico*,
 - Por otro lado se tiene el aprendizaje que combina las inducción y la deducción. Los sistemas que siguen ese modelo en su versión más completa se denomina sistemas integrados. El conocimiento de partida no tiene ahora por qué ser ni correcto ni completo.

1. APRENDIZAJE DEDUCTIVO.

Un método más específico de los conocidos como aprendizajes analíticos y que aquí hemos identificado con los deductivos *puros*, es el aprendizaje basado en la explicación. Es una manera de construir descripciones de un concepto desde un ejemplo típico con la ayuda de un conocimiento previo (teoría del dominio). El tema central de este aprendizaje para construir descripciones generalizadas es sencillo:

Dada una instancia del concepto objetivo, se utiliza la teoría del dominio para explicar cómo esta instancia satisface al concepto. Esta explicación puede analizarse para que sea aplicable no solo a la instancia de partida sino a también a instancias similares, y puede utilizarse como parte añadida de la descripción del concepto para futuros reconocimientos de otras instancias del concepto.

Un ejemplo intuitivo de este tipo de aprendizaje que aparece en la literatura es el de un jugador de cartas que conoce varios juegos de cartas y sus reglas ante el aprendizaje de un juego de cartas nuevo. Aplicará todo su conocimiento de los otros juegos (teoría del dominio) al que está

aprendiendo, y refinará su conocimiento de él conforme va jugando partidas (instancias). El primer paso del jugador será explicar una jugada con el conocimiento previo acumulado de otros juegos de cartas. El segundo paso será analizar la explicación encontrada para intentar generalizarla.

- **Existen dos técnicas principales en este aprendizaje que pueden verse como complementarias:**

- La de los *macro-operadores*, que trata de combinar pequeñas piezas de conocimiento en otras más grandes lo que permite resolver el problema de dar mayores pasos en la búsqueda a través del espacio de hipótesis.
- La del *conocimiento para el control de la búsqueda*, que trata de captar conocimiento para decidir qué regla o paso debe darse en el instante siguiente.

- **Las teorías del dominio, para ciertos dominios de aplicación de estos sistemas, puede resultar:**

- *Incompletas*; es decir que el conocimiento es insuficiente para lograr probar que un ejemplo es miembro del concepto. A pesar de ello puede ser posible formar explicaciones *plausibles* que establezcan vínculos entre la teoría del dominio y los ejemplos de entrenamiento.
- *Intratables*; es decir computacionalmente prohibitivas (p.e. reglas completas del ajedrez). La solución es construir un teoría del dominio tratable y aproximada pero que constituya una base para y aprender conceptos desde explicaciones.
- *Inconsistentes*; es decir la teoría del dominio contiene conocimiento contradictorio. Desde ellas es posible construir explicaciones contradictorias. La solución consiste en ir depurando esa inconsistencia.

Seguidamente se describe someramente el sistema EBG (*Explanation-Based Generalization*) (Mitchell et al. 1986) como un caso representativo del aprendizaje basado en la explicación con la técnica de los macro-operadores. Básicamente su funcionamiento es el siguiente:

- **Dados:**

- Una definición de un concepto objetivo; o sea, una definición del concepto que se trata de aprender, con el supuesto de que esta definición no es *operativa* como definición para manejar.
- Ejemplo de entrenamiento; o sea, un ejemplo del concepto objetivo.
- Teoría del dominio, o sea, un conjunto de reglas y hechos que se utilizarán para explicar cómo el ejemplo de entrenamiento es un ejemplo del concepto objetivo.
- Criterio de operabilidad; o sea un predicado referente a las definiciones del concepto objetivo y que indique como la forma en que debe de ser expresada la definición del concepto aprendido.

- **Hallar:**

- Una generalización del ejemplo de entrenamiento que sea una descripción suficiente del concepto objetivo y que satisfaga el criterio de operabilidad.

Ejemplo

Cosidérese el siguiente conjunto de reglas como una teoría de un cierto dominio:

$$\text{Odia}(x,y), \text{posee}(x,z), \text{Es}(x, \text{arma}) \longrightarrow \text{Mata}(y,y).$$

$$\text{Deprimido}(x) \longrightarrow \text{Odia}(x,x).$$

$$\text{Compra}(x,y) \longrightarrow \text{Posee}(x,y).$$

$$\text{Es}(x,\text{pistola}) \longrightarrow \text{Es}(x,\text{arma}).$$

$$\text{Deprimido}(\text{John}).$$

$$\text{Compra}(\text{John}, \text{Objeto1}).$$

$$\text{Es}(\text{objeto1}, \text{pistola}).$$

Se trata de probar como objetivo, partiendo de esta teoría de dominio, que John perpetrará un suicidio; es decir que: $\text{mata}(\text{John}, \text{John})$.

En un primer estadio EBG construye el árbol para la explicación del ejemplo mediante la teoría del dominio, figura 42.

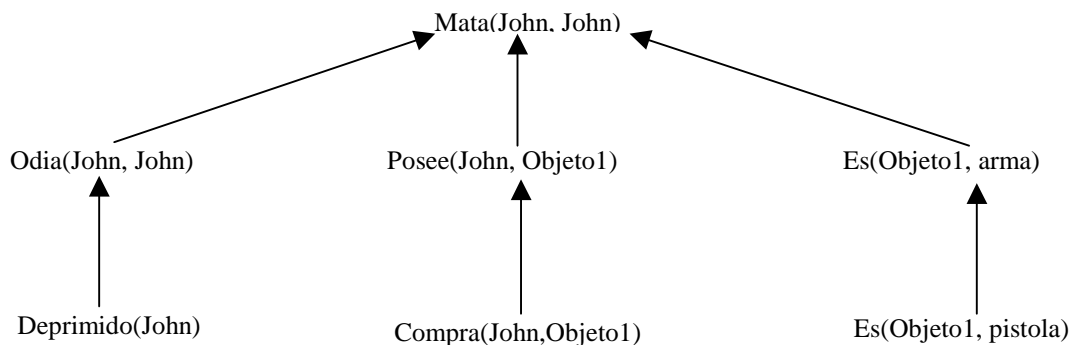


Figura 42: Árbol para la explicación del ejemplo del suicidio

A continuación EBG busca una descripción para ello retrocede a la fórmula del concepto objetivo a través del árbol de la explicación para obtener una descripción del concepto operativa y generalizada, figura 43.

Por último se realiza un proceso de reconstrucción de la demostración para asegurarse que la generalización aprendida es correcta. En este caso la macro-regla aprendida es:

$$\text{Deprimido}(y), \text{Compra}(y,c), \text{Es}(c, \text{pistola}) \longrightarrow \text{Mata}(y,y)$$

Sin este proceso de reconstrucción de la demostración que no incorporaba el sistema EBG original se podían perder restricciones que daban lugar a reglas incorrectas del tipo:

$$\text{Deprimido}(y), \text{Compra}(y,c), \text{Es}(c, \text{pistola}) \longrightarrow \text{Mata}(x,y)$$

De forma resumida la forma de operar del algoritmo es la siguiente:

- **El algoritmo**

EBG-1 Explicación

Construir una explicación utilizando la teoría del dominio que prueba como el concepto objetivo es satisfecho por el ejemplo de entrenamiento. Cada rama de la estructura de la explicación concluirá en una expresión que satisface el criterio de operabilidad

EBG-2 Propagación

Se regresa al concepto objetivo a través de la estructura de la explicación para determinar un conjunto suficiente de condiciones bajo las cuales la explicación sigue siendo verdadera. También estas condiciones deben de verificar el criterio de operabilidad. Se obtiene el conjunto de antecedentes generalizados.

EBG-3 Rehacer la Demostración

Se comienza con los antecedentes generalizados y se rehace la demostración para obtener el concepto objetivo generalizado.

Históricamente ese sistema marcó un hito pues supuso la unificación de los elementos del aprendizaje basado en la explicación, sin embargo fue posteriormente criticado y superado por sistemas posteriores.

Por último, citar dentro de los sistemas de aprendizaje analítico mediante la técnica del conocimiento para el control de la búsqueda a los siguientes:

- LEX2 (Mitchell et al. 1983) y su evolución METALEX (Keller 1987)
- PRODIGY (Minto et al. 1988)
- SOAR, (Laird et al. 1986)

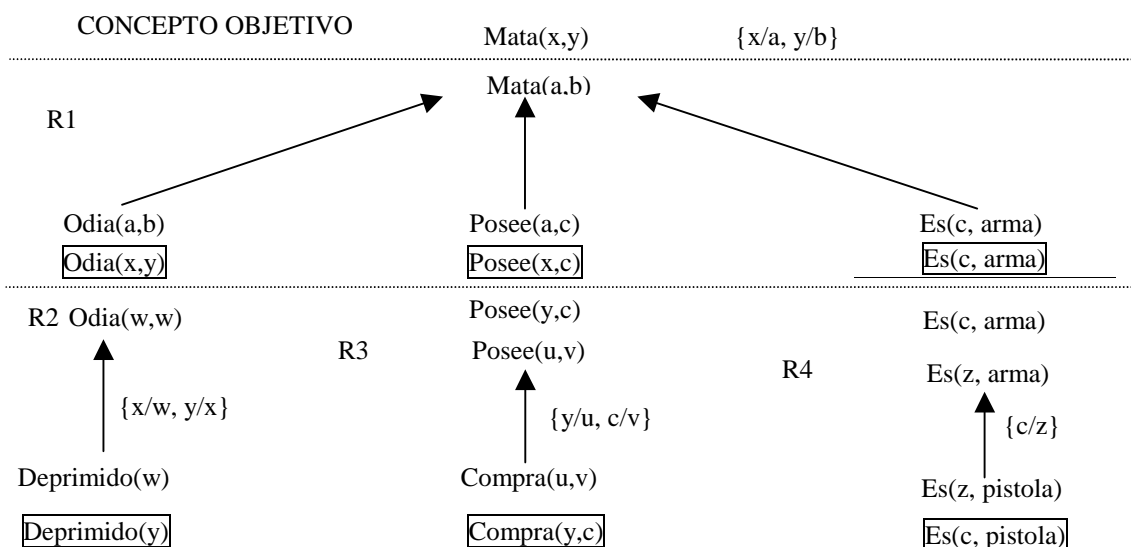


Figura 43: Proceso de generalización en el ejemplo del suicidio

2. SISTEMAS INTEGRADOS: FOCL.

La teoría del dominio que se maneja en un problema de aprendizaje deductivo puede ser imperfecta por demasiado particular, y el sistema tratará de generalizarla, demasiado general, y el sistema trata de particularizarla, y por último, puede ser ambas cosas. Los *sistemas integrados*, también conocidos como *sistemas para la revisión de teorías generales*, son los que tratan con teorías imperfectas por ambos motivos. Existen sistemas que trabajan exclusivamente con teorías sobre-generalizadas o sobre-particularizadas, pero aquí comentaremos un sistema integrado que puede decirse que integra la labor de los sistemas de los otros dos tipos.

- **El Sistema FOCL**

El sistema FOCL (Pazzani et al. 1991) es una extensión de sistema puramente inductivo FOIL, que le añade la capacidad de aprendizaje basado en la explicación y como él aprende reglas en forma de cláusulas Horn, mediante recubrimiento de los ejemplos positivos y exclusión de los negativos. La diferencia principal con el sistema FOIL radica en los literales candidatos para incorporar a una regla en el proceso de su construcción; además de contemplar los que contemplaba aquel, FOCL contempla literales que provienen del conocimiento que posee del dominio y proporcionado inicialmente como entrada. La elección del literal que se incorporará en la regla para particularizarla se realiza mediante la función heurística de la ganancia-Foil.

El tipo de problema que resuelve FOCL es el siguiente:

- **Dados:**
 - El nombre de un predicado de aridad conocida.
 - Un conjunto de tuplas positivas.
 - Un conjunto de tuplas negativas.
 - Un conjunto de definiciones extensionales de predicados.
 - Un conjunto de definiciones intensionales de predicados (opcional).
 - Un conjunto de restricciones sobre los predicados intensionales y extensionales (opcional).
 - Una regla inicial para el predicado (opcional).
- **Hallar:**
 - Una regla para el predicado en términos de los predicados extensionales tales que ninguna cláusula cubre ejemplos negativos y algunas cláusulas cubren todos los ejemplos positivos.

En cada momento de su búsqueda de lo general a lo específico, FOCL, expande la hipótesis actual utilizando dos operadores; el primero semejante al de FOIL y el segundo específico de él; utiliza el concepto de *operacional* que bien puede ser un solo literal o un conjunto de ellos. Los métodos de especialización de una regla en construcción son los siguientes:

1. Para cada literal *operacional* que no es parte ya de la regla en construcción, crear una especialización añadiendo un literal sencillo a las precondiciones, figura 44.
2. Crear un *operacional* que se corresponda con una condición lógicamente suficiente para el concepto objetivo de acuerdo con la teoría del dominio. Añadir los literales que constituyen el *operacional* a las actuales precondiciones de la regla en construcción. Finalmente podar de la regla los literales que supongan redundancia de acuerdo con el conjunto de entrenamiento.

FOCL además tien otras novedades:

Utiliza cierto tipo de variables con restricciones que reducen el espacio de búsqueda.

Puede hacer que predicados especiales como $\text{igual}(x,x)$ pueda no considerarse como trivialmente verdadero.

Acepta una regla incorrecta como aproximación al predicado que debe de ser aprendido.

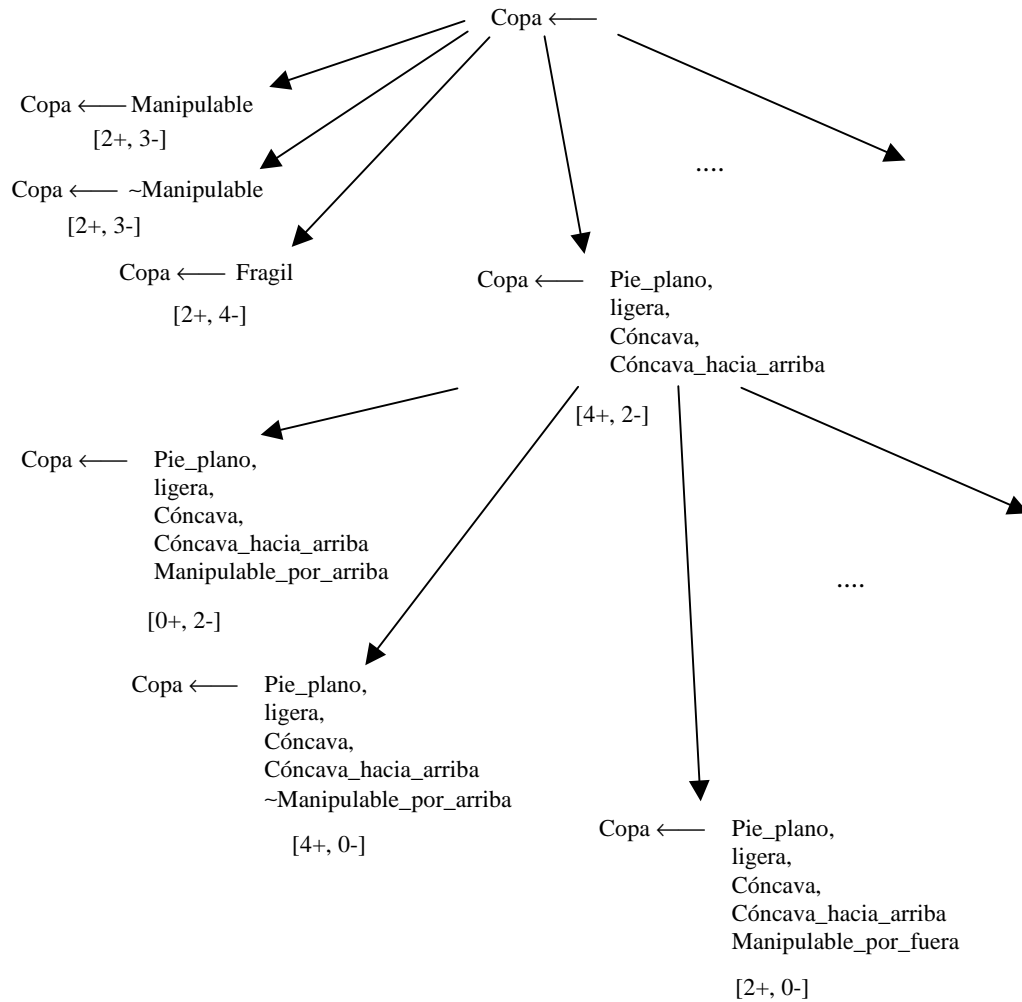


Figura 44: Trazo de un proceso construcción de una regla con la incorporación de literales de la teoría del dominio.

IV. APRENDIZAJE BASADO EN INSTANCIAS Y POR ANALOGÍA.

Se desarrollan en este tema tres nuevos métodos de aprendizaje que son el *basado en ejemplares (o instancias)*, el *razonamiento basado en casos*, y el *aprendizaje por analogía* los dos primeros con un principio subyacente común: el almacenamiento masivo de conocimiento con un nivel bajo de refinamiento mediante ejemplos o instancias. A partir del conocimiento almacenado y por comparación se determinará en el futuro nuevo conocimiento sobre los casos venideros. De alguna forma se deciden las características de un caso por su *parecido o analogía* con los casos retenidos en memoria. La mención que acabamos de hacer a la analogía no ha sido casual; nos permite justificar la asociación de estos dos tipos de aprendizaje con el *aprendizaje por analogía* y desarrollarlos en un mismo bloque temático aunque para el último la aplicación de la analogía es mucho más elaborada que la simple comparación. No está totalmente justificado un nexo mayor entre estos tipos de aprendizaje que el principio común de utilizar la analogía, aunque sea con diferente grado en su aplicación. Sin embargo, no puede decirse que esta asociación sea original ya que abunda en la literatura.

1. APRENDIZAJE BASADO EN INSTANCIAS: MÉTODO DE LOS K-VECINOS MÁS PRÓXIMOS, NGE.

El *aprendizaje basado en ejemplares o instancias* en sus múltiples variantes tiene como principio de funcionamiento el almacenamiento de ejemplos: en unos casos todos los ejemplos de entrenamiento, en otros solo los más representativos, en otros los incorrectamente clasificados cuando se clasifican por primera vez, etc. La clasificación posterior se realiza por medio de una función que mide la *proximidad* o parecido. Dado un ejemplo para clasificar se le clasifica de acuerdo al ejemplo o ejemplos más próximos. El *bias* que rige este método es la proximidad; es decir, la generalización se guía por la proximidad de un ejemplo a otros. Algunos autores consideran este *bias* más apropiado para el aprendizaje de conceptos naturales que el correspondiente al proceso inductivo (Bareiss et al. en [KODRATOFF et al. 90]), por otra parte también se ha estudiado la relación entre este método y los que generan reglas (Clark, 1990).

Se han enumerado ventajas e inconvenientes del aprendizaje basado en ejemplares [BRISCOE et al. 96], pero se suele considerar no adecuado para el tratamiento de atributos no numéricos y valores desconocidos. Las mismas medidas de proximidad sobre atributos simbólicos suelen proporcionar resultados muy dispares en problemas diferentes.

- **El método de los k-vecinos más próximos**

Consideramos como un buen representante de este tipo de aprendizaje, y que es de gran sencillez conceptual, el *método de los k-vecinos más próximos* [MITCHELL 97]. Se suele denominar método porque es el esqueleto de un algoritmo que admite el intercambio de la *función de proximidad* dando lugar a múltiples variantes. La función de proximidad puede

decidir la clasificación de un nuevo ejemplo atendiendo a la clasificación del ejemplo o de la mayoría de los k ejemplos más cercanos. Admite también funciones de proximidad que consideren el peso o coste de los atributos que intervienen, lo que permite, entre otras cosas, eliminar los atributos irrelevantes. Una función de proximidad clásica entre dos instancias x_i y x_j , si suponemos que un ejemplo viene representado por una n -tupla de la forma $(a_1(x), a_2(x), \dots, a_n(x))$ en la que $a_r(x)$ es el valor de la instancia para el atributo a_r , es la siguiente:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^n (a_r(x_i) - a_r(x_j))^2}$$

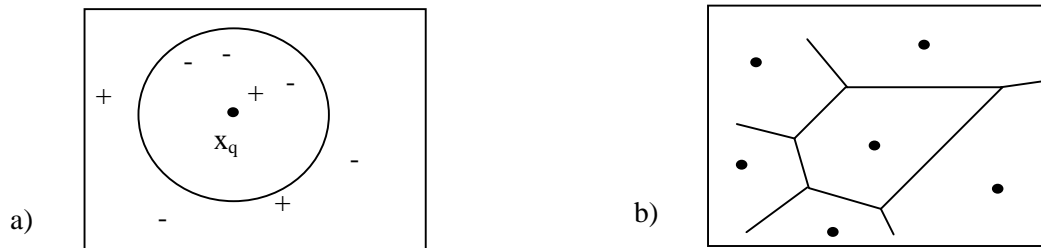


Figura 45: a) Según el valor de k la instancia x_q se clasificará de maneras diferentes. b) Partición de la superficie de acuerdo con un caso típico de clasificación mediante un algoritmo de los k vecinos más próximos.

- **El algoritmo**

Se expone para el caso del aprendizaje o estimación de funciones f con valores en un espacio discreto $V = \{v_1, v_2, \dots, v_n\}$, es decir $f: \mathbb{R}^n \longrightarrow V$. El valor estimado para una instancia que se clasifica es el valor más común entre los k vecinos más próximos.

Algoritmo de entrenamiento

Comenzar

Para cada ejemplo e = añadirlo a la lista de Ejemplos_de_Entrenamiento *fin_para*

Fin

Algoritmo de clasificación

Entradas:

x_q : una instancia o ejemplo para clasificar.

$\{x_1, x_2, \dots, x_k\}$: conjunto de las k instancias o ejemplos de entrenamiento más próximos a x_q .

Comenzar

$$\text{Devolver } f'(x_q) = \arg.\max_{v \in V} \left\{ \sum_{i=1}^k \delta(v, f(x_i)) \right\}, \quad \text{donde} \quad \delta(a,b) = \begin{cases} 1 & \text{si } a = b \\ 0 & \text{en otro caso} \end{cases}$$

Fin.

- **Otra función de proximidad**

El algoritmo anterior se puede adaptar de forma sencilla al aprendizaje o estimación de funciones con imagen en un espacio continuo. Bastará sustituir la expresión que produce el valor que devolverá por la siguiente:

$$\text{Devolver } f'(x_q) = \frac{\sum_{i=1}^k f(x_i)}{k}$$

En este caso el valor estimado para la instancia que se calcula es el valor medio de los valores de los k vecinos más próximos.

- **Vecino más próximo con distancia ponderada**

Para los casos anteriores si se introduce una ponderación de la distancia a cada ejemplo de entrenamiento resultan las expresiones siguientes:

$$\text{Devolver } f(x_q) = \arg.\max_{v \in V} \left\{ \sum_{i=1}^k w_i \delta(v, f(x_i)) \right\} \quad \text{donde} \quad w_i = \frac{1}{d(x_q, x_i)}$$

$$\text{Devolver } f(x_q) = \frac{\sum_{i=1}^k w_i f(x_i)}{\sum_{i=1}^k w_i}$$

Por último, comentar que con este método se han incorporado funciones de proximidad más complejas, incluso auxiliadas por redes neuronales.

- **El sistema NGE de Salzberg**

Un sistema que por su manera novedosa de almacenar los ejemplos merece una reseña es *Nested Generalized Exemplar* (NGE) de Salzberg (1990), sistema incremental que produce reglas y una métrica basada en los pesos asignados por el sistema a ejemplos y atributos.

2. RAZONAMIENTO BASADO EN CASOS.

El *razonamiento basado en casos* puede considerarse que maneja el mismo principio que el anterior pero en un nivel superior. Los *ejemplos* en esta ocasión son sustituidos por descripciones simbólicas muy ricas que se denominan *casos*, almacenados por el sistema para utilizarlos al estilo de los ejemplos en el método anterior. Una novedad importante es la medida de la proximidad entre casos. En el método anterior estaba basada en una representación de los ejemplos en un espacio euclídeo mientras que ahora la complejidad de las descripciones de los casos inutiliza esa representación y hay que auxiliarse de otras métricas. Debido a la especificidad de las aplicaciones de los sistemas de aprendizaje que siguen este método, simplemente se hace referencia a algunos de ellos: CADET (Sycara et al 1992) [MITCHELL 97], o JUDGE (Schank 1987) CHEF (Hammond 1989) [BRISCOE et al. 96], etc., por si el lector estuviera interesado en el estudio de algunas de las métricas utilizadas. (Para un estudio más detallado de estos sistemas cabe citar la obra de Janet Kolodner titulada *Case-Based Reasoning* de Morgan Kaufmann Publishers, USA 1993).

3. APRENDIZAJE POR ANALOGÍA.

El razonamiento por analogía es un proceso que conlleva inferencias inductivas, deductivas y con *abducción*. Se parte de un conocimiento sobre relaciones de un cierto dominio conocido que se aplica a las relaciones de un dominio objetivo. Mientras que en el proceso inductivo se genera conocimiento desde niveles muy bajos, en la analogía se transfiere conocimiento desde el dominio de partida hacia el dominio objetivo. En general, el proceso de aprendizaje por analogía no está actualmente bien comprendido y quedan muchas cuestiones por resolver. En este apartado, dedicado a los sistemas de aprendizaje que trabajan con la analogía, esbozamos los principios de funcionamiento de aquellos que la aplican de la manera hoy considerada como standard. El esquema general de aplicación de la analogía al aprendizaje queda resumido en el diagrama de la figura 46.

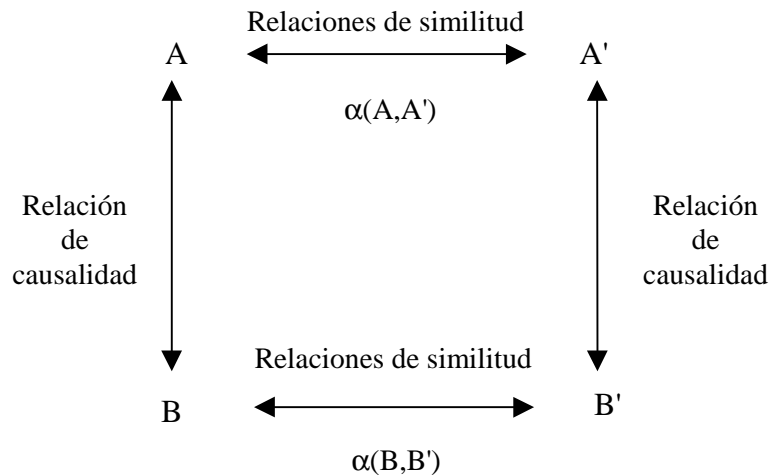


Figura 46: Esquema general de la analogía

La relación de causalidad β -no necesariamente en sentido estricto- de dos elementos en el dominio de partida se transmite a sendos elementos homólogos del dominio objetivo como relación de causalidad β' , siempre que se satisfaga una relación de similitud α entre los elementos homólogos del primer dominio y del segundo.

Se consideran cuatro fases en el proceso de aplicación de la analogía al aprendizaje:

- *búsqueda del dominio de partida,*
- *búsqueda de una aplicación entre los dominios,* normalmente biyectiva
- *evaluación de la aplicación entre dominios elegida,* incorporada por pocos sistemas, y
- *aprendizaje,* que puede ser más o menos refinado e incluso aprender de las situaciones de analogía que fallen.

Los modos que se reconocen en la aplicación de la analogía son tres, y todos se acomodan al esquema de la figura 46 con ligeras variantes:

- **Analogía proporcional.** La que concierne con relaciones *proporcionales* entre los elementos homólogos de los dos dominios; al estilo de ciertas cuestiones de test: *A es a B como A' es a B'*.
- **Analogía predictiva.** También conocida como inferencia analógica y que trata de *alcanzar* relaciones entre objetos del dominio objetivo justificadas a partir de las relaciones sobre los objetos homólogos del dominio de partida. Es en este proceso donde se aplica la abducción, figura 47; es decir, se busca el antecedente de un condicional cuyo consecuente conocemos, de forma que por aplicación de la regla *modus ponens* al condicional construido junto con el antecedente como hecho, obtengamos, como conclusión, el consecuente que fue el dato de partida. Obviamente puede haber varias soluciones al problema de la abducción, que habrá que valorar; además, no queda garantizado el *mantenimiento de la verdad*.
- **Analogía interpretativa** se centra en la búsqueda de similitudes entre los objetos de los dominios, que no viene especificada a priori; llevada al extremo de la exageración, y quizás fuera de contexto pero muy clarificador en el aula, es la base de los manidos chistes sobre parecidos: "*¿A qué se parece...*"

Llegados a este punto, una vez que damos por suficiente la introducción de la analogía, solamente resta exponer brevemente las líneas de funcionamiento de un sistema de aprendizaje basado en la analogía. Como sistemas que aplican la analogía están COPYCAT (Hofstadter y Mitchell 1988) y ANALOGY (Winston1983), un clásico en las referencias a sistemas que aprenden por analogía, del que se detalla siempre el proceder de obtención de las consecuencias sobre una cierta pareja de cónyuges basado en la analogía que tiene con el prototipo de pareja formada por Macbecht, hombre de carácter débil, y su esposa, mujer avariciosa que arrastra a su marido a la maldad. Los sistemas citados y otros más los describe someramente Briscoe et al [BRISCOE et al. 96]. Para una mayor profusión en los detalles desde un punto de vista teórico sobre aplicación de la analogía al aprendizaje tenemos los artículos de Carbonell cuyas referencias aparecen en la bibliografía básica [MICHALSKI et al. 83] [MICHALSKI et al. 86].

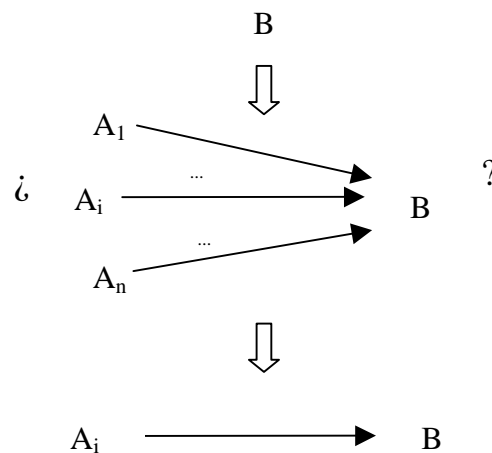


Figura 47: Esquema del proceso abductivo

BIBLIOGRAFÍA BÁSICA

[BRISCOE 96]

Briscoe, G., Caelli, T. *A Compendium of Machine Learning*. Vol. 1: *Symbolic Machine Learning*. Ablex Publishing Corporation, New Jersey, 1996.

[LANGLEY 96]

Langley, P. *Elements of Machine Learning*. Morgan Kaufmann Publishers, San Francisco, California, 1996.

[MITCHELL 97]

Mitchell, T.M. *Machine Learning*. McGraw-Hill, USA, 1997.

[MUGGLETON 90]

Muggleton, S. *Inductive Acquisition of Expert Knowledge*. Turing Press in association with Addison-Wesley Publishing Company, Avon (Great Britain), 1990.

[NILSSON 96]

Nilsson, N. J. *Introduction to Machine Learning*, (borrador para evaluación). Stanford University, California, 1996.

[QUINLAN 93]

Quinlan, J. R. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo (California), 1993.

BIBLIOGRAFÍA GENERAL

REFERENCIAS

V. OTROS ENFOQUES DEL APRENDIZAJE.

1 ENFOQUE CONEXIONISTA: INTRODUCCIÓN A LAS REDES NEURONALES.

1.1. EL PERCEPTRON Y REGLAS DE ENTRENAMIENTO.

1.2. REDES MULTICAPA Y EL ALGORITMO DE BACKPROPAGATION.

2. ENFOQUE EVOLUTIVO: INTRODUCCIÓN A LOS ALGORITMOS GENÉTICOS

3. EL APRENDIZAJE POR REFUERZO.