Capítulo 4

Las reglas de asociación y su uso

1. Extensiones del modelo básico

1.1 Jerarquías de conceptos

El uso de taxonomías o jerarquías de conceptos puede ser muy útil a la hora de descubrir reglas de asociación en grandes bases de datos. Tales jerarquías pueden verse simplemente como grafos dirigidos acíclicos. Los arcos entre pares de nodos describen relaciones del tipo ES-UN (*IS-A* en inglés). Por lo tanto, los nodos hoja del grafo representan los conceptos más específicos del universo de "items". El uso de jerarquías de conceptos puede justificarse por las siguientes razones:

- Reglas que relacionen conceptos muy específicos pueden no llegar al umbral de relevancia mínima. Por ejemplo, si en un supermercado hay muchas variedades de cereales la regla de asociación que relaciona la compra de cereales con la compra de leche (*"Cereal⇒Leche"*) podría no llegar a obtenerse nunca ya que reglas concretas como *"Kellogs⇒Puleva"* quizá no alcancen los umbrales mínimos de relevancia.

- Se permite la poda de reglas redundantes. Siguiendo el ejemplo anterior, muchas reglas que asocien la compra de marcas concretas de cereales con la adquisición de leche de cualquier tipo pueden resumirse en una sola regla del tipo *"Cereal⇒Leche"*.

- Se puede acelerar el proceso de análisis de los datos. Inicialmente, el usuario puede obtener reglas de asociación generales que asocien conceptos en niveles altos de la jerarquía. Posteriormente, uno se puede centrar en áreas concretas que sean de su interés. Como resulta evidente, analizar un conjunto reducido de itemsets en la base de datos reduce el trabajo realizado por los algoritmos de extracción de reglas, lo que permite un funcionamiento interactivo de las apliaciones de análisis al acelerar el proceso de data mining.

- Se puede realizar un análisis más detallado de los datos, centrándo éste en cada momento en las partes de la base de datos que más nos interesen.

Hemos de resaltar que de las reglas de asociación que se obtienen cuando se utilizan jerarquías de conceptos (denominadas reglas de asociación generalizadas) se han de eliminar aquéllas que contengan literales en el consecuente que correspondan a generalizaciones de literales en el antecedente). Así mismo, una regla será de interés sólo si revela información que no podría obtenerse de reglas más generales. En el ejemplo del supermercado, reglas relativas a marcas concretas de cereales serían redundantes si también se ha obtenido una regla más general del tipo *"Cereal⇒Leche"* (salvo que la relevancia y fiabilidad de la regla concreta se desvíen notablemente de las de la regla más general).

1.2 Manejo de atributos numéricos

La obtención de reglas de asociación clásica se realiza en bases de datos transaccionales. En ellas, cada transacción contiene o no contiene un ítem determinado. Por lo tanto, los atributos de las transacciones pueden considerarse booleanos [*BARP: Boolean Association Rules Problem*]. Un algoritmo típico para resolver este problema es *Apriori*. Sin embargo, cuando el proceso de Data Mining se aplica a otros tipos de bases de datos, los atributos pueden ser categóricos o numéricos [*QARP: Quantitative Association Rules Problem*].

Los atributos exclusivamente booleanos de BARP pueden considerarse un caso particular de los atributos categóricos que nos encontramos en otros dominios, por lo que no suponen ningún problema para los algoritmos clásicos como *Apriori*.

Sin embargo, a la hora de trabajar con atributos continuos (atributos numéricos con valores comprendidos en un determinado intervalo), éstos han de discretizarse de algún modo para poder aplicar cualquier algoritmo clásico de extracción de reglas de asociación. La técnica más usual es dividir los valores de cada atributo numérico en una serie de intervalos. De esta forma los atributos numéricos se pueden tratan exactamente igual que los demás atributos (categóricos).

Al definir los intervalos que nos sirven para discretizar los atributos numéricos nos encontramos con dos problemas:

- PROBLEMA 1 [*MinSup problem*]

  Por un lado, si definimos un número excesivamente grande de intervalos, la relevancia [*support*] asociada a ellos puede ser demasiado baja, posiblemente por debajo del umbral previamente establecido por el usuario.

- PROBLEMA 2 [*MinConf problem*]

  Si establecemos un pequeño número de intervalos, la fiabilidad de las reglas de asociación que los incluyan se verá afectada negativamente (estas reglas tal vez no lleguen a alcanzar la fiabilidad mínima requerida para ser consideradas interesantes, con la consiguiente pérdida de información).

En resumen:

Si los intervalos son demasiado grandes las reglas no alcanzarán la fiabilidad mínima. Por el contrario, si dichos intervalos son demasiado pequeños, las reglas no llegarán a tener la relevancia mínima.

El problema asociado con la relevancia mínima puede solucionarse fusionando intervalos adyacentes o reduciendo el umbral *MinSupport*. El asociado con la fiabilidad mínima se arregla incrementando el número de intervalos. Desgraciadamente, incrementar el número de intervalos provoca un aumento del tiempo de ejecución de los algoritmos y del número de reglas espúreas generadas.

Para evitar que al solucionar el primer problema estemos provocando el segundo, se podría establecen un límite superior para la relevancia de los intervalos considerados [*MaxSupport*] que determine hasta qué punto se han de seguir combinando intervalos adyacentes.

Existe una gran variedad de métodos de discretización binaria (umbralización en Visión Artificial) basados en la selección de un umbral. Las técnicas utilizadas para seleccionar este umbral suelen ser de tipo estadístico (la mayor parte de ellas utiliza medidas sobre el histograma) y pueden aplicarse a distintos tipos de problemas.

Fayyad e Irani discuten distintas técnicas ampliamente utilizadas en la construcción de clasificadores para la discretización de atributos numéricos o continuos (ya sean enteros, reales o un conjunto de valores ordenados). Se pretende generalizar la discretización binaria que se realiza en algoritmos de construcción de árboles de decisión como CART, ya que ésta suele resultar más eficiente que otras no binarias basadas, por ejemplo, en medidas de entropía (vg: ID3). Sin embargo, la discretización binaria se comporta peor cuando existen más de dos clases diferentes. Fayyad e Irani proponen realizar discretizaciones binarias sucesivas utilizando un criterio de decisión basado en el principio MDL de Rissanen.

El método de partición equitativa [*equi-depth partition*] propuesto por Skirant y Agrawal consiste en distribuir uniformemente los casos de los que dispongamos. Es decir, si tenemos N casos y M intervalos, a cada intervalo han de corresponderle N/M casos aproximadamente. El método es simple e intuitivo pero no siempre funciona bien (valores adyacentes que estén relacionados pueden quedar en intervalos diferentes). Para obtener las reglas de asociación, Skirant y Agrawal exponen una variante del algoritmo Apriori que hace uso de una medida de interés para podar los conjuntos de candidatos *C[k]*

Fukuda y sus compañeros del centro de investigación de IBM en Tokyo proponen un algoritmo bastante más sofisticado para encontrar reglas de asociación en las que los atributos numéricos discretizan. Su algoritmo intenta conseguir la partición de los valores de un atributo numérico que maximice tanto la relevancia como la fiabilidad de las reglas obtenidas. Para ello convierte el problema en un problema geométrico que puede resolverse utilizando algoritmos eficientes bien conocidos en geometría computacional.

Z. Zhang, Y. Lu y B. Zhang proponen utilizar técnicas de clustering para discretizar en intervalos los atributos numéricos. Inicialmente se divide el conjunto de valores de cada atributo numérico en una serie de intervalos. Si dos intervalos adyacentes están relacionados (aparecen en reglas de asociación similares) entonces se combinan para formar un único intervalo (simplificando así el conjunto de reglas). Algunos intervalos pueden descartarse en el proceso de clustering si no se consideran interesantes.

1.2.1 MAQA [Mining Association among Quantitative Attributes]

Para cada atributo numérico (e incluso para los atributos categóricos que tengan un número elevado de valores diferentes), Zhang y sus colaboradores de la Universidad de Tsinghua en Beijing (China) proponen el algoritmo de clustering que a continuación exponemos.

Para cada máximo local Max_i del histograma del atributo se encuentran los mínimos locales $MinL_i$ y $MinR_i$ situados a su izquierda y a su derecha respectivamente. A continuación se obtiene el número de casos incluidos entre $MinL_i$ y $MinR_i$, valor al que denotaremos por Sum_i. Finalmente, se obtiene S_{ave} como la media de los valores Sum_i.

La amplitud de los intervalos obtenidos de este modo puede ser diferente. Si el valor Sum_i está por encima de la media S_{ave}, entonces el intervalo entre $MinL_i$ y $MinR_i$ se considera interesante.

Para que el algoritmo sea efectivo el valor S_{ave} debe ser representativo (no debería aproximarse ni a $min\{Sum_i\}$ ni a $max\{Sum_i\}$). Esto puede solucionarse si al calcular S_{ave} no incluimos ni el máximo ni el mínimo de los valores Sum_i.

Por otro lado, si el número de casos cubierto por cada intervalo es similar (todos los valores Sum_i próximos a S_{ave}), es difícil determinar qué intervalos son más interesantes que otros. Para solucionar este problema se utiliza otra medida de interés: el intervalo entre $MinL_i$ y $MinR_i$ se considera interesante si $Sum_i/(MinR_i-MinL_i) > Sum/(Max-Min)$, donde Sum es la suma de los valores que toma el atributo, Min es su mínimo y Max su máximo.

Además, descartar todos los intervalos que no alcancen S_{ave} puede suponer una pérdida importante de información. Aunque un intervalo contenga pocos casos, podría ser interesante su combinación con intervalos adyacentes. Si la amplitud de un intervalo es pequeña y el mínimo local que lo une a su vecino es cercano a su máximo local, entonces se fusionan los dos intervalos en uno solo.

1.3 Medidas alternativas de relevancia

A la hora de aplicar un algoritmo de extracción de reglas de asociación, utilizar simplemente la frecuencia de aparición de los distintos itemsets en la base de datos puede producir resultados no deseados, tales como la aparición de itemsets espúreos o la omisión de reglas que podrían resultar interesantes al usuario.

Por ejemplo, supongamos que hemos recogido datos acerca de las costumbres de los estudiantes universitarios a través de una encuesta realizada a 5000 alumnos. De ellos, 3000 practican algún tipo de deporte habitualmente, 3750 beben alcohol cuando salen los fines de semana y 2000 realizan ambas actividades. Un algoritmo clásico obtendría que la regla de asociación *"deporte ⇒ alcohol"* se cumple con una confianza [*confidence*] del 66.6% cuando el 75% de los entrevistados consume alcohol. La regla obtenida resulta engañosa ya que el consumo de alcohol entre los deportistas es menor que en el conjunto global de los estudiantes. De hecho, la práctica de algún deporte influye 'negativamente' en el consumo de alcohol. Por tanto, la regla *"deporte ⇒ ¬alcohol"* sería más precisa a pesar de que el itemset *{deporte, ¬alcohol}* es menos frecuente que *{deporte, alcohol}* y la regla de asociación tendría menos fiabilidad (33.3%).

Si bajamos lo suficiente los umbrales de relevancia (*support* en este caso) y fiabilidad se pueden obtener reglas de asociación contradictorias (en este ejemplo *"deporte ⇒ alcohol"* y *"deporte ⇒ ¬alcohol"*). Por otro lado, un valor excesivamente bajo de los umbrales podría conducir a la generación de un conjunto enorme e inmanejable de reglas de asociación. Además, si prefijamos un valor alto para estos umbrales, sólo se generaría la regla *"deporte ⇒ alcohol"* que, obviamente, resulta menos significativa. En otras palabras, ninguna combinación de los umbrales utilizados en los algoritmos clásicos de reglas de asociación obtendría la asociación 'correcta', que en el ejemplo corresponde a *"deporte ⇒ ¬alcohol"*.

Tras observar efectos como el mostrado en el ejemplo anterior, Aggarwal y Yu propusieron el uso de una medida de interés alternativa al calcular los itemsets relevantes que después se utilizan para construir las reglas de asociación. A esta medida alternativa la denominaron 'fuerza colectiva' [*collective strength*].

La fuerza colectiva de un itemset es un número real mayor o igual que cero. Un valor de 0 indica una correlación negativa perfecta (cuando la presencia de un ítem excluye la presencia de los demás items del itemset). Un valor de fuerza colectiva igual a 1 corresponde a cuando la presencia del itemset es la esperada teniendo en cuenta las probabilidades de aparición de los items individuales. La fuerza colectiva $C(I)$ de un itemset I se define como

$$C(I) = \frac{1 - v(I)}{1 - E\{v(I)\}} \cdot \frac{E\{v(I)\}}{v(I)}$$

donde $v(I)$ es la proporción del número de transacciones en que aparecen algunos de los items del itemset pero no todos [*violation ratio*]. Nótese que un valor elevado de $1\text{-}v(I)$ ocurre cuando la correlación entre la presencia de distintos items en un itemset es elevada.

Aggarwal y Yu definieron que un itemset *I* es fuertemente colectivo a un nivel K cuando su fuerza colectiva *C(I)* y la de todos sus subconjuntos es, al menos, igual a K. De esta forma (al forzar la propiedad de monotonía) se puede utilizar la fuerza colectiva de los itemsets en cualquier algoritmo clásico de extracción de reglas de asociación como un parámetro más, cuya utilización es además ortogonal al uso de la frecuencia de aparición de los itemsets en la base de datos [*support*].

1.4 Obtención de reglas de asociación con restricciones

En la práctica, los usuarios que utilizan herramientas de Data Mining suelen estar interesados en analizar únicamente un pequeño subconjunto de la base de datos completa. Por ejemplo, un distribuidor podría estar interesado en obtener información acerca de los clientes que tiene en determinada área geográfica, restringiendo de esa forma el conjunto de datos acerca de los que desea obtener información.

Obviamente, las restricciones indicadas por el usuario pueden utilizarse en una etapa de post-procesamiento del conjunto completo de reglas de asociación. Sin embargo, la integración de dichas restricciones en el proceso de obtención de las propias reglas puede reducir notablemente el tiempo de respuesta del sistema ante las peticiones del usuario.

Skirant, Vu y Agrawal propusieron en KDD'97 distintos algoritmos de obtención de reglas de asociación que permiten incorporar restricciones definidas mediante expresiones booleanas sobre la presencia o ausencia de items en los itemsets. Variantes de estos algoritmos también pueden utilizarse para obtener reglas de asociación incorporando taxonomías (véase el apartado correspondiente a las jerarquías de conceptos).

Ng, Lakshmanan, Han y Pang proponen otro algoritmo, al que denominan CAP, que permite especificar restricciones de una forma más flexible utilizando una sintaxis similar a la de SQL.

1.5 Mantenimiento de bases de datos de reglas de asociación

Otro problema que ha recibido bastante atención es el mantenimiento de un conjunto de reglas de asociación actualizándolo conforme se modifica la base de datos. Mantener la integridad del conjunto de reglas no es algo trivial y los algoritmos propuestos suelen limitarse al caso de bases de datos en las que sólo se realizan inserciones (por ejemplo, las que registran las transacciones comerciales realizadas por una compañía).

Entre los algoritmos propuestos destacan FUP, FUP* y MLUp, tres métodos ideados por David W. Cheung y sus colaboradores. Feldman y su equipo también han realizado diferentes propuestas, entre las que destaca su algoritmo Delta.

2. Post-procesamiento de las reglas de asociación

Los algoritmos de extracción de reglas de asociación pueden producir un número enorme de reglas. Esto podría llegar incluso a interpretarse como un fenónemo de sobreaprendizaje [*overfitting*] al generarse todos los patrones que satisfacen determinadas restricciones (efecto de Bonferroni). Paradójicamente, un algoritmo de data mining puede producir una cantidad tan grande de datos que necesite el uso de alguna técnica de data mining (lo que se conoce como un problema de data mining de segundo orden).

Por ejemplo, Megiddo y Skirant proponen ordenar la presentación de las reglas ante el usuario utilizando un test de significancia estadística de forma que se distingan las reglas de asociación verdaderamente predictivas de las que no lo son. Por ejemplo, dada una regla de asociación $S \Rightarrow T$ donde S y T son itemsets deberíamos comprobar si $p(S \wedge T) > p(S) \cdot p(T)$ para eliminar falsos descubrimientos (reglas obtenidas que no incrementan nuestro conocimiento del problema).

Determinar qué reglas pueden interesar realmente al usuario es un problema bastante complejo. Mika Klemettinen y sus colaboradores de la Universidad de Helsinki (Finlandia) proponen la utilización de patrones [*templates*] que especifiquen el formato de las reglas y los atributos involucrados en ellas de forma que el usuario sólo visualiza un subconjunto del conjunto total de reglas obtenidas. Además, se propone el uso del producto de la relevancia y la fiabilidad de la regla (medida a la que se denomina *commonness*) como un parámetro adicional cuyo umbral puede fijar el usuario (para descartar reglas que simultáneamente son poco relevantes y poco fiables).

Lent, Swami y Widom, por su parte, proponen reducir el conjunto de reglas de asociación obtenidas agrupando reglas similares para obtener reglas más generales. Inicialmente se obtienen reglas del tipo $X \wedge Y \Rightarrow G$, donde X e Y son atributos numéricos y G es un atributo categórico (por ejemplo, el objetivo de una clasificación). Los valores de los atributos numéricos se agrupan en intervalos de amplitud fija y se construye una imagen en la que el valor de cada píxel (x,y) corresponde al valor del consecuente G de la regla correspondiente a los intervalos de X e Y representados por x e y, respectivamente. De esta forma, intervalos consecutivos resultantes de la discretización de los atributos numéricos quedan representados en píxeles adyacentes en la imagen. Para reducir el efecto del ruido y los outliers, la imagen se pasa por un filtro paso-bajo (utilizando alguna técnica de suavizado de las muchas existentes en procesamiento digital de señales). Una vez suavizada la imagen, al agrupar píxeles adyacentes etiquetados de la misma forma obtendremos un conjunto reducido de reglas de asociación generalizadas. Lent, Swami y Widom utilizan un método iterativo en el que pretenden obtener el mejor conjunto reducido de reglas según un criterio heurístico basado en el principio MDL de Rissanen. Los mejores resultados de este método se obtienen comenzando con un umbral de relevancia bajo que posteriormente se va incrementando paulatinamente en cada iteración.

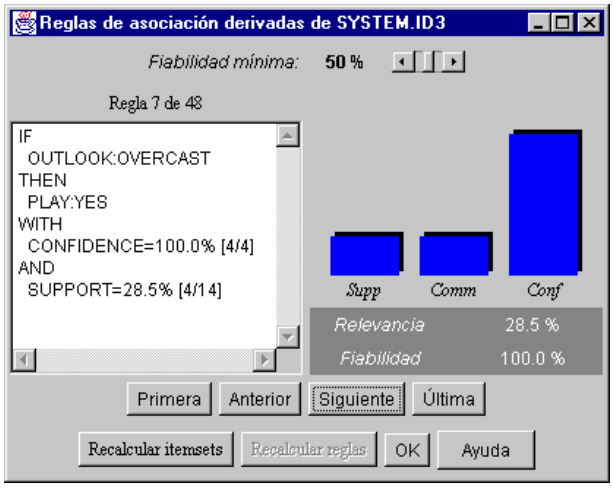
2.1 Técnicas de visualización

El uso de técnicas de visualización puede ayudar en el manejo de grandes conjuntos de reglas. En el artículo de Klemettinen y sus colaboradores se propone la utilización de diagramas de barras que muestran la relevancia y la fiabilidad de cada regla de asociación, así como el producto de ambas [*commonness*].

Knobbe y Adriaans, de Syllogic, utilizan también diagramas de barras tridimensionales para mostrar conjuntos de asociaciones binarias. En este tipo de diagramas se pueden mostrar distintas medidas de similaridad o correlación entre los valores de pares de atributos (algunas basadas en la Teoría de la Probabilidad, como la probabilidad condicionada, y otras en la Teoría de la Información, como la información mutua).

Por otro lado, también se ha propuesto la utilización de grafos que muestren simultáneamente distintas reglas de asociación. En estos grafos, los nodos corresponden a los distintos items y las aristas a las asociaciones entre conjuntos de items. Por ejemplo, Klemettinen et al. emplean el grosor y el color de las aristas para representar la fiabilidad y relevancia de las distintas reglas mientras que Knobbe y Adriaans prefieren obtener un árbol generador minimal del grafo dirigido completo del conjunto de asociaciones utilizando algoritmos como el de Kruskal o el de Prim.

Sin embargo, dado que las reglas de asociación no son transitivas ($S \Rightarrow T$ y $T \Rightarrow U$ no implica necesariamente $S \Rightarrow U$), la utilización de grafos como método de visualización no parece demasiado adecuada para la representación simultánea de varias reglas de asociación.



Visualización de una regla complementada con el uso
de un diagrama de barras tal como propone Klemettinen

3. Otras Aplicaciones

3.1 Análisis de secuencias temporales

El descubrimiento de eventos que suelen ocurrir conjuntamente (esto es, en un determinado intervalo de tiempo o ventana temporal) puede realizarse utilizando algoritmos clásicos de extracción de reglas de asociación. Basta con considerar transacciones aquellas secuencias de eventos consecutivos incluidos en una ventana. Sin embargo, aunque esta forma de solucionar el problema es muy simple, no tiene en cuenta una característica esencial de las secuencias temporales: la ordenación parcial de los sucesos (algunos ocurren siempre antes o después de otros, o bien pueden ocurrir en paralelo).

Para poder tratar adecuadamente la ordenación de los eventos, se define un episodio como un conjunto de conjuntos de eventos parcialmente ordenados. Un episodio puede estar formado por sub-episodios dada su naturaleza recursiva. Obviamente, un episodio sólo será frecuente cuando lo sean todos sus sub-episodios (como sucede con los itemsets).

En KDD'95 se puede encontrar un algoritmo de Mannila, Toivonen y Verkamo que permite analizar secuencias temporales siguiendo la filosofía del algoritmo Apriori.

3.2 Análisis de imágenes

Los algoritmos de extracción de reglas de asociación también pueden utilizarse en bases de datos de imágenes. Entre las posibles aplicaciones de este tipo de técnicas se encuentran la predicción meteorológica, el reconocimiento aéreo (con fines militares o económicos), la investigación médica e, incluso, la criminal.

En principio se podrían identificar los objetos que aparecen en una imagen mediante las diferencias de color y textura en distintas regiones de las imágenes. A partir de estos objetos, sin etiquetar inicialmente, podría accederse a objetos similares contenidos en otras imágenes mediante un sistema de CBIR [*Content-Based Image Retrieval*]. En esta situación un algoritmo clásico de data mining podría utilizarse para determinar si algunos objetos suelen aparecer conjuntamente. Las reglas de asociación así obtenidas podrían utilizarse, por ejemplo, para clasificar automáticamente los objetos contenidos en las imágenes.

Ordonez y Omiecinski proponen dividir el análisis de las imágenes en distintas etapas. En primer lugar se ha de realizar una segmentación de las imágenes. Es decir, cada imagen ha de dividirse en regiones, para lo cual es necesaria la utilización de técnicas propias de Visión Artificial. Idealmente, cada región corresponderá a un objeto representado en la imagen. Posteriormente, los objetos de cada imagen han de ser identificados comparándolos con los objetos de las demás imágenes. Esta etapa se puede realizar utilizando algún tipo de algoritmo de agrupamiento [*clustering*]. Finalmente, se deben obtener las reglas de asociación existentes entre los objetos identificados utilizando algoritmos clásicos de Data Mining.

3.3 Clasificación con reglas de asociación

El aprendizaje supervisado o clasificación es un problema de gran interés para los expertos en Inteligencia Artificial. Los datos de entrada (denominados conjunto de entrenamiento) son instancias de las clases que se desean modelar e incluyen una serie de atributos o características. El objetivo de la clasificación es obtener una descripción precisa para cada clase utilizando los atributos de los datos de entrada. El modelo así obtenido puede servir para clasificar casos cuyas clases se desconozcan o, simplemente, para comprender mejor la información de la que disponemos.

Muchas veces no podemos construir modelos completos que nos permitan una clasificación perfecta de todos los casos con los que nos podemos encontrar. A veces tendremos que conformarnos con descubrir modelos aproximados que contemplen algunas características de las distintas clases sin que el modelo abarque todas las clases posibles ni todos los casos particulares de una clase determinada.

Una clasificación completa puede que no sea factible cuando hemos de tratar con una gran cantidad de atributos, cuando muchos valores son desconocidos, unos atributos deben modelarse en función de otros o el número de casos de entrenamiento es excesivamente elevado. La clasificación parcial intenta descubrir características comunes a los distintos casos de cada clase sin formar un modelo predictivo completo.

La extracción de reglas de asociación puede ser útil para resolver problemas de clasificación parcial donde las técnicas de clasificación clásicas no son efectivas. Los árboles de decisión no son muy adecuados para tratar con información incompleta (valores desconocidos en atributos de los casos de entrenamiento) y resultan problemáticos cuando unos atributos son función de otros. Las redes neuronales tampoco son apropiadas cuando tenemos información incompleta y su entrenamiento puede llegar a consumir demasiado tiempo. Por su parte, las técnicas empleadas en ILP (*Inductive Logic Programming*) son mucho menos eficientes que los algoritmos de extracción de reglas de asociación.

El problema de la clasificación parcial se puede resolver usando reglas de asociación de dos formas diferentes:

① *Dividiendo el conjunto de casos de entrenamiento*

Para los casos de entrenamiento de cada clase C se aplica un proceso de obtención de itemsets relevantes A. Posteriormente, se comparan los patrones obtenidos con los presentes en el conjunto de entrenamiento de forma que se eliminan aquellos patrones que, aun siendo frecuentes, no resultan predictivos (usando un test chi-cuadrado).

$$r(A \Rightarrow C) = \frac{P(C|A)}{P(C|\neg A)}$$

Cuando este cociente (denominado "riesgo relativo" [*relative risk*]) es elevado, podemos considerar interesante la regla $A \Rightarrow C$. Por ejemplo, carecería de interés clasificar una enfermedad atendiendo a síntomas que no siempre se manifiestan asociados a esa enfermedad y sería trascendental identificar síntomas específicos de una enfermedad (aunque éstos sean muy poco frecuentes).

② *Considerando la clase como un atributo más*

Al tratar la clase como un atributo más, hemos de aplicar el proceso de obtención de reglas de asociación al conjunto completo de casos de entrenamiento. Es decir, se aplican algoritmos típicos de obtención de reglas de asociación (como *Apriori* o *DHP*) al conjunto completo de datos y, posteriormente, se procesa el conjunto de reglas obtenidas (que puede ser enorme).

Una vez generado el conjunto completo de reglas, para cada regla de asociación A⇒C obtenida ha de calcularse su riesgo relativo utilizando la siguiente expresión:

$$r(A \Rightarrow C) = \frac{P(C|A)}{P(C|\neg A)} = \frac{\frac{Support(C \cup A)}{Support(A)}}{\frac{Support(C) - Support(C \cup A)}{1 - Support(A)}}$$

Para no perder información importante, el umbral de relevancia mínima ha de ser muy pequeño, lo que implica un mayor coste computacional a la hora de obtener los itemsets relevantes. Para que el proceso de obtención de reglas sea eficiente hemos de incorporar al algoritmo de extracción de reglas de asociación la restricción de que sólo nos interesan las reglas que contengan como consecuente la clase del caso de entrenamiento (con lo que se reduce el tiempo consumido en el proceso de *Data Mining*).

Empíricamente se ha comprobado que los resultados obtenidos de esta forma son similares a los encontrados dividiendo el conjunto de casos de entrenamiento, por lo que se suele optar por la primera estrategia.

Otra alternativa para construir clasificadores a partir de reglas de asociación consiste en ir obteniendo paulatinamente reglas de la forma $A \Rightarrow \{C: c_j\}$, donde A es un itemset y $\{C: c_j\}$ es un ítem que nos indica la clase que se deberá asignar a los ejemplos que verifiquen A. Obsérvese que el itemset A no es más que un conjunto de pares *atributo:valor* y, por tanto, el antecedente de la regla $A \Rightarrow \{C: c_j\}$ es una conjunción de descriptores (selectores según la terminología utilizada en la metodología STAR).

Inicialmente, partimos del conjunto completo de datos y buscamos aquellos itemsets de la forma $\{A_i:a_i, C:c_j\}$ que nos permitan derivar reglas del tipo $IF\ A_i=a_i\ THEN\ C=c_j$. Si no encontrásemos itemsets que nos permitiesen obtener reglas con una capacidad de predicción mínima (dada por el parámetro *MinConfidence* típico de los algoritmos clásicos de obtención de reglas de asociación), continuaríamos la búsqueda a partir de itemsets de mayor tamaño. Es decir, cuando un atributo no nos sirve para discriminar adecuadamente entre las distintas clases, se utilizan combinaciones de varios atributos. Una vez agotados los itemsets $\{A_i:a_i, C:c_j\}$, proseguiríamos con itemsets de la forma $\{A_{i1}:a_{i1}, A_{i2}:a_{i2}, C:c_j\}$ con los que derivar reglas del tipo $IF\ A_{i1}=a_{i1}\ AND\ A_{i2}=a_{i2}\ THEN\ C=c_j$ y así sucesivamente.

La idea básica de ART es utilizar en cada instante todos los itemsets relevantes $A \cup \{C: c_j\}$ en los que A esté formado siempre por descriptores correspondientes a los mismos atributos. De estos itemsets relevantes se deriva un conjunto de reglas de la forma $IF\ A\ THEN\ C=c_j$.

Por ejemplo, como atributos recogidos en A se pueden seleccionar aquéllos que permiten la obtención de un conjunto de reglas $IF\ A\ THEN\ C=c_j$ que maximice el número de casos cubiertos del conjunto de entrenamiento. De esta forma se minimiza el número de casos que requieren reglas *IF-THEN* más complejas.

Los casos que no quedan cubiertos por los itemsets relevantes seleccionados $A \cup \{C: c_j\}$ se pueden agrupan para formar reglas de la forma $IF\ \neg A\ AND\ ...\ THEN\ C=c_j$. El antecedente de estas reglas se construye a partir de los itemsets relevantes dentro de este conjunto de casos no cubiertos por las reglas anteriores de la misma forma, teniendo en cuenta que son un subconjunto del conjunto inicial con valores distintos para los atributos de A.

Este proceso, aparentemente complejo, puede considerarse completamente equivalente a la construcción de árboles de decisión en los que se permite la existencia de ramas ELSE, salvo que en ocasiones se emplean varios atributos simultáneamente para ramificar. Esta analogía justifica la denominación de esta técnica: ART, acrónimo inglés correspondiente a "árbol de reglas de asociación".

Se ha comprobado experimentalmente que ART construye buenos clasificadores, aunque si los vemos como árboles de decisión, hay que decir que dichos árboles están muy poco balanceados. En cada instante se busca siempre obtener reglas directas que permitan clasificar la mayor cantidad posible de casos (en cierto modo, el proceso es similar al de construcción de un árbol Huffman).

Utilicemos como ejemplo la conocida tabla propuesta por Quinlan en la que se determina si se juega o no al golf en función de las condiciones meteorológicas. El método ART obtiene lo siguiente cuando *MinSupport* es igual a dos tuplas:

```
OUTLOOK:{ 'overcast' }
   PLAY : { 'P' } (4)
En otro caso
   TEMPERATURE:{ 'hot' }
      PLAY : { 'N' } (2)
   En otro caso
      OUTLOOK:{ 'rain' },WINDY:{ 'false' }
         PLAY : { 'P' } (3)
      OUTLOOK:{ 'rain' },WINDY:{ 'true' }
         PLAY : { 'N' } (2)
      En otro caso
         HUMIDITY:{ 'normal' }
            PLAY : { 'P' } (2)
         En otro caso
            PLAY : { 'N' } (1)
```

Recordemos que un algoritmo clásico de construcción de árboles de decisión (tal como ID3) habría obtenido lo siguiente:

```
OUTLOOK = overcast
     PLAY = P (4.0)
OUTLOOK = rain
     WINDY = false
          PLAY = P (3.0)
     WINDY = true
          PLAY = N (2.0)
OUTLOOK = sunny
     HUMIDITY = high
          PLAY = N (3.0)
     HUMIDITY = normal
          PLAY = P (2.0)
```

Por su parte un algoritmo como AQ o CN2 obtendría la siguiente descripción para cada una de las clases del problema:

```
PLAY = N                        PLAY = P
--------                        --------

OUTLOOK = sunny                 OUTLOOK = overcast
AND
TEMPERATURE = hot

OUTLOOK = rain                  OUTLOOK = rain
AND                             AND
WINDY = true                    WINDY = false

OUTLOOK = sunny                 OUTLOOK = sunny
AND                             AND
HUMIDITY = high                 HUMIDITY = normal
```

Obsérvese la similaridad existente entre los modelos de clasificación conseguidos con los diferentes métodos.

Para otros conjuntos de datos más complejos (como *MUSHROOM*, de 8124 tuplas), ART también consigue construir buenos clasificadores:

```
ODOR:{ 'a' }
   EDIBLEPOISONOUS : { 'e' } (400)
ODOR:{ 'l' }
   EDIBLEPOISONOUS : { 'e' } (400)
ODOR:{ 'c' }
   EDIBLEPOISONOUS : { 'p' } (192)
ODOR:{ 'f' }
   EDIBLEPOISONOUS : { 'p' } (2160)
ODOR:{ 'm' }
   EDIBLEPOISONOUS : { 'p' } (36)
ODOR:{ 'p' }
   EDIBLEPOISONOUS : { 'p' } (256)
ODOR:{ 's' }
   EDIBLEPOISONOUS : { 'p' } (576)
ODOR:{ 'y' }
   EDIBLEPOISONOUS : { 'p' } (576)
En otro caso
   SPOREPRINTCOLOR:{ 'b' }
      EDIBLEPOISONOUS : { 'e' } (48)
   SPOREPRINTCOLOR:{ 'h' }
      EDIBLEPOISONOUS : { 'e' } (48)
   SPOREPRINTCOLOR:{ 'k' }
      EDIBLEPOISONOUS : { 'e' } (1296)
   SPOREPRINTCOLOR:{ 'n' }
      EDIBLEPOISONOUS : { 'e' } (1344)
   SPOREPRINTCOLOR:{ 'o' }
      EDIBLEPOISONOUS : { 'e' } (48)
   SPOREPRINTCOLOR:{ 'y' }
      EDIBLEPOISONOUS : { 'e' } (48)
   SPOREPRINTCOLOR:{ 'r' }
      EDIBLEPOISONOUS : { 'p' } (72)
   En otro caso
      GILLSIZE:{ 'b' }
         EDIBLEPOISONOUS : { 'e' } (528)
      En otro caso
         STALKSURFACEABOVERING:{ 'f' }
            EDIBLEPOISONOUS : { 'e' } (24)
         STALKSURFACEABOVERING:{ 'k' }
            EDIBLEPOISONOUS : { 'p' } (32)
         STALKSURFACEABOVERING:{ 'y' }
            EDIBLEPOISONOUS : { 'p' } (8)
         En otro caso
            CAPCOLOR:{ 'c' }
               EDIBLEPOISONOUS : { 'e' } (12)
            CAPCOLOR:{ 'n' }
               EDIBLEPOISONOUS : { 'e' } (12)
            CAPCOLOR:{ 'w' }
               EDIBLEPOISONOUS : { 'p' } (8)
```

El principal inconveniente de ART se encuentra en que es bastante más rápido construir un árbol de decisión con un algoritmo clásico y después podarlo que construir el clasificador ART. Además, los resultados obtenidos con ambos métodos son similares y resulta más fácil interpretar un árbol en el que no existan ramas ELSE.

NB: Una ventaja de ART se encuentra en el uso de un valor adecuado del parámetro *MinConfidence*, que le permite tratar con facilidad y naturalidad información con ruido (algo para lo que los algoritmos TDIDT de construcción de árboles de decisión requieren el uso posterior de técnicas de poda, AQ no permite y CN2 pretende conseguir con más o menos éxito).

4. Referencias bibliográficas

Jerarquías de conceptos

- Jiawei Han & Yongjian Fu
  "Discovery of multiple-level association rules from large databases"
  21st Intl. Conference on Very Large Databases, Zürich, 1995.

- Ramakrishnan Skirant & Rakesh Agrawal
  "Mining generalized association rules"
  21st Intl. Conference on Very Large Databases, Zürich, 1995.

Manejo de atributos numéricos

- Usama M. Fayyad & Keki B. Irani
  "Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning". © Machine Learning

- Takeshi Fukuda, Yasuhiko Morimoto, Shinichi Morishita & Takeshi Tokuyama:
  "Mining Optimized Association Rules for Numeric Attributes"
  PODS'96, Montreal, Quebec, Canadá, 1996

- Ramakrishnan Skirant & Rakesh Agrawal:
  "Mining Quantitative Association Rules in Large Relational Tables"
  SIGMOD'96.

- Zhaohui Zhang, Yuchang Lu & Bo Zhang:
  "An effective partitioning-combining algorithm for discovering quantitative association rules" En "KDD. Techniques and Applications", H. Lu et al. eds., World Scientific, 1997.

Medidas alternativas de relevancia

- Charu C. Aggarwal & Philip S. Yu
  "A New Framework for Itemset Generation"
  ACM SIGMOD PODS'97

Reglas de asociación con restricciones

- Raymond T. Ng, Laks V.S. Lakshmanan, Jiawei Han & Alex Pang
  "Exploratory Mining and Pruning Optimizations of Constrained Association Rules"
  ACM SIGMOD'98.

- Ramakrishnan Skirant, Quoc Vu & Rakesh Agrawal
  "Mining Association Rules with Item Constraints"
  KDD'97.

Mantenimiento de bases de datos de reglas de asociación

- David W. Cheung, Jiawei Han, Vincent T. Ng & C.Y. Wong
  "Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique". 1996 Intl. Conference on Data Engineering, New Orleans, 1996.

- David W. Cheung, Vincent T. Ng & Benjamin W. Tam
  "Maintenance of Discovered Knowledge: A Case in Multi-level Association Rules"
  KDD'96, Oregon, USA, August 1996

- R. Feldman, A. Amir, Y. Auman, A. Zilberstien & H. Hirsh
  "Incremental Algorithms for Association Generation"
  En "KDD: Techniques and Applications", World Scientific, H.Lu et al. eds., 1997

POST-PROCESAMIENTO

- M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen & A.I. Verkamo
  "Finding Interesting Rules from Large Sets of Discovered Association Rules"
  CIKM'94, Gaitherburg MD (USA), Nov. 1994

- Arno J. Knobbe & Pieter W. Adriaans
  "Analysing Binary Associations"
  KDD'96, Oregon (USA), 1996

- Brian Lent, Arun Swami & Jennifer Widom
  "Clustering Association Rules"
  Stanford University

- Nimrod Megiddo & Ramakrishnan Skirant
  "Discovering Predictive Association Rules"
  © AAAI 1998

Análisis de secuencias temporales

- Heikki Mannila, Hannu Toivonen & A. Inkeri Verkamo
  "Discovering frecuent episodes in sequences"
  KDD'95

Análisis de imágenes

- Carlos Ordonez & Edward Omiecinski
  "Image Mining: A New Approach for Data Mining"
  Georgia Institute of Technology, 1998

Clasificación con reglas de asociación

- Kamal Ali, Stefanos Manganaris & Ramakrishnan Skirant
  "Partial Classification Using Association Rules"
  KDD'97 , California (USA), 1997