UNIVERSIDAD DE GRANADA

E.T.S. DE INGENIERÍA INFORMÁTICA

Departamento de Ciencias de la Computación
e Inteligencia Artificial

FUSIÓN DE CONOCIMIENTO EN BASES DE DATOS RELACIONALES: MEDIDAS DE AGREGACIÓN Y RESUMEN

TESIS DOCTORAL

José María Serrano Chica

Granada, Septiembre de 2003

Editor: Editorial de la Universidad de Granada
Autor: María Victoria Martínez Jiménez
D.L.: GR 522-2013
ISBN: 978-84-9028-372-1

FUSIÓN DE CONOCIMIENTO EN BASES DE DATOS RELACIONALES: MEDIDAS DE AGREGACIÓN Y RESUMEN

JOSÉ MARÍA SERRANO CHICA

FUSIÓN DE CONOCIMIENTO EN BASES DE DATOS RELACIONALES: MEDIDAS DE AGREGACIÓN Y RESUMEN

MEMORIA QUE PRESENTA

JOSÉ MARÍA SERRANO CHICA

PARA OPTAR AL GRADO DE DOCTOR EN INFORMÁTICA

SEPTIEMBRE 2003

DIRECTORES

MARIA AMPARO VILA MIRANDA
DANIEL SÁNCHEZ FERNÁNDEZ

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
E INTELIGENCIA ARTIFICIAL

E.T.S. de INGENIERÍA INFORMÁTICA UNIVERSIDAD DE GRANADA

La memoria titulada **Fusión de conocimiento en bases de datos relacionales: Medidas de agregación y resumen**, que presenta D. José María Serrano Chica para optar al grado de Doctor, ha sido realizada en el Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada bajo la dirección de los doctores Dña. Maria Amparo Vila Miranda y D. Daniel Sánchez Fernández.

Granada, Septiembre de 2003

El doctorando | Los directores

Fdo. J. Mª Serrano Chica | Fdo. A. Vila Miranda | Fdo. D. Sánchez Fernández

Agradecimientos

Esta memoria no hubiera sido posible sin la ayuda de varias personas, siendo destacable la inestimable labor de mis directores de tesis, responsables directos de mi interés hacia la investigación. Gracias a Dani por dirigirnos a Juanra y a mí nuestro Proyecto Final de Carrera, mi primera toma de contacto con la Minería de Datos. Y gracias a Amparo por dirigir mi atención hacia las Bases de Datos y su extensión difusa. Gracias también a Gabriel y Julio, del Dpto. de Edafología y Química Agrícola de la Universidad de Granada. Sin ellos esta memoria carecería de significado práctico.

Dedico esta memoria a mi novia Teresa y a mi familia, a mis padres y a mis hermanas, en especial a mi hermana Paqui Loli, "Francis", quien, junto con Teresa, ha tenido a bien aguantarme durante todo este tiempo. Gracias a las dos por vuestra paciencia y comprensión.

Han pasado cuatro años ya desde que me incorporara al Departamento de Ciencias de la Computación y, dentro de él, al grupo de Bases de Datos y Sistemas de Información Inteligentes (IdBIS). Comencé como becario adscrito a un proyecto de investigación de la Universidad de Granada, con el que inicié mi formación como investigador. Dicha formación continuó con una beca predoctoral de la Junta de Andalucía. Gracias a ambas entidades por permitir llevar a buen fin esta memoria, lo cual tampoco hubiera sido posible sin el excelente ambiente de trabajo dentro del departamento. Gracias a todos, especialmente a mis camaradas de IdBIS. Mención aparte se merece Nacho, que siempre ha estado ahí cuando lo he necesitado, desde el mismo principio.

Por último, no sería justo olvidar a la facción joven de Ciencias de la Computación. Los "Meceneros", habitantes contra viento y marea del Módulo Mecenas, entre los que me siento orgulloso de incluirme. Ésta no es la primera memoria que se redacta entre sus paredes, como seguro tampoco será la última. Ánimo y gracias a todos.

Granada, Septiembre de 2003

Índice general

Índice de Figuras

Índice de Algoritmos

Índice de Tablas

1. Introducción

De las múltiples disciplinas que abarcan las Ciencias de la Computación, podría decirse que la Representación del Conocimiento, adquirido del mundo real para su posterior uso, es una de las más generales e importantes. Hoy en día, gran cantidad de esfuerzo se destina a procurar un almacenamiento lo más eficiente posible, al tiempo que se mantenga la máxima fidelidad con respecto a la información alusiva al problema del mundo real que se pretende abarcar. Pero el interés no se centra únicamente en almacenar un tipo concreto de conocimiento, sino en darle una posterior utilidad. Este hecho, unido al de que en la actual y autodenominada sociedad de la información se precisa de ingentes cantidades de datos sobre los más diversos caracteres y para las más diversas situaciones, propicia el que se planteen algunos problemas surgidos al respecto. Entre ellos podría destacarse el de la necesidad de extraer información de un posible interés de entre los datos recogidos. Aparte de las herramientas típicas de consulta con las que cuenta cualquier sistema de almacenamiento y manejo de datos, en ocasiones es necesario el uso de

herramientas más específicas para obtener un conocimiento más complejo a partir de la información almacenada.

En la actualidad se cuenta con múltiples modelos para la representación de información, con distintas características en función de la semántica que se le quiera dar a los datos. Existen modelos pensados para el almacenamiento de características, otros más adecuados para la representación de acciones, e incluso combinaciones de varios de estos modelos. De entre los primeros, cabe destacar el modelo relacional de bases de datos, posiblemente uno de los más extendidos por su popularidad e inherente simplicidad, sobre el que nos extenderemos más adelante en esta memoria.

Por otro lado, durante los últimos años, se han estudiado y desarrollado una serie de técnicas y metodologías, englobadas bajo los términos anglosajones *Data Mining and Knowledge Discovery in Databases* (traducibles por Minería de Datos y Extracción de Conocimiento en Bases de Datos), para abordar el problema de analizar grandes cantidades de información en busca de conocimiento hasta entonces desconocido y de un posible interés. Aunque actualmente existe toda una pléyade de metodologías de este tipo, en todas ellas el objetivo final continúa siendo básicamente el mismo: extraer el máximo posible de información potencialmente útil de una forma eficiente, y proporcionar unos resultados a partir del análisis de los datos que el usuario sea capaz de comprender y aplicar para su propio beneficio.

Otro de los problemas que nos pueden generar las actuales bases de datos estriba en la necesidad de representar y manejar la información de la forma más cercana posible al modo en que lo hace el ser humano. El lenguaje natural suele estar altamente afectado por imprecisión o vaguedad cuando, por ejemplo, en una conversación informal usamos expresiones tan comunes para nosotros como *"Juan es alto, pero no tanto como Antonio"*, o *"Parece que hoy hace un poco de frío"*. Casi con total seguridad, nuestro interlocutor será capaz de captar la semántica del mensaje, pero no así un ordenador, al menos en principio. La riqueza del lenguaje humano se pierde cuando hay que introducir valores en una máquina tales como *"Juan mide 1 metro 80"*, *"Antonio mide 1 metro 87"*, o *"Hoy la temperatura es de 10º C"*. Hoy en día, se ha avanzado bastante en relación a esta materia. Existen bastantes resultados teóricos y prácticos que, aplicados a la representación del conocimiento, nos permiten,

en cierta forma, modelar la imprecisión o incertidumbre que puedan venir asociadas a unos datos dentro de un ordenador, y permitirnos, en una etapa posterior, trabajar con ellos. La Teoría de los Subconjuntos Difusos, propuesta por L.A. Zadeh en 1965, es un buen ejemplo de ello. La Teoría de las Bases de Datos, y especialmente los modelos relacionales, han extendido su ámbito de actuación con el objeto de, no sólo poder recuperar información en un modo que sea fácilmente comprensible por un usuario, mediante lo que se denominan consultas flexibles, sino también permitir que dicha información pueda ser recogida del mundo real y posteriormente almacenada sin que por ello pierda sus características de vaguedad, incertidumbre o imprecisión. Pero estos procesos introducen una complejidad adicional a la hora de analizar los datos y extraer conocimiento más complejo a partir de los mismos.

Este trabajo se encuadra en dicha línea de investigación, como se expondrá a continuación en los objetivos de esta memoria. Estudiaremos algunos modelos existentes de representación de información afectada de imprecisión o incertidumbre con objeto de proponer nuevas herramientas de extracción de conocimiento que permitan ser aplicadas con éxito sobre dicha información, procurando que nos proporcionen resultados interesantes y comprensibles haciendo un uso eficiente de los recursos disponibles.

Antecedentes

En concreto, partiremos de un modelo de representación y almacenamiento del conocimiento, como es el Modelo Relacional de Bases de Datos (introducido inicialmente por E.F. Codd en 1970) y sus posteriores extensiones. Más adelante en la memoria, describiremos el modelo GEFRED [Medina et al, 1994], extensión del modelo relacional de Codd para manejar imprecisión e incertidumbre conjuntamente. Siguiendo este modelo, ha sido posible implementar FIRST [Medina et al, 1995], una arquitectura para mantener bases de datos difusas sobre un sistema de gestión de bases de datos actual.

A continuación, estudiaremos algunas de las herramientas existentes para el análisis y obtención de información útil a partir de los datos almacenados, dentro de lo que hemos definido anteriormente como Minería de Datos y Extracción de Conocimiento. Recordemos que estas técnicas tienen como objetivo

la búsqueda de nueva información, no trivial y de una posible utilidad, a partir de datos previamente almacenados en una base de datos. Especialmente, nos vamos a centrar en el estudio de las posibles relaciones entre atributos. De entre las variadas herramientas con las que cuenta la Minería de Datos, nos vamos a centrar sobre un grupo de las mismas, las llamadas reglas de asociación y dependencias aproximadas.

Las Reglas de Asociación [Agrawal et al., 1993] miden la posible relación que pueda existir entre ítems en un conjunto de transacciones. Más adelante veremos cómo es posible extender dichas reglas para establecer relaciones entre valores de atributos dentro de un conjunto de objetos, que normalmente representaremos en una base de datos relacional. Además, entran en nuestro campo de investigación las Dependencias Aproximadas en bases de datos relacionales, que también pueden aportar información sobre relaciones existentes en nuestra base de datos. De manera informal, se puede definir una dependencia aproximada como una dependencia funcional con "excepciones". Sobre estos conceptos, existen distintos enfoques para la valoración de las reglas de asociación (como los discutidos en [Berzal et al., 2001a]), aunque como se explicará más adelante, usaremos el basado en el factor de certeza de Shortliffe and Buchanan [Shortliffe y Buchanan, 1975]. Volveremos a estos puntos en posteriores apartados de esta memoria.

Más en profundidad, abordaremos las extensiones existentes de estas dos metodologías para manejar información que pueda estar afectada por algún grado de incertidumbre, imprecisión o vaguedad. El porqué de esta apreciación estriba en el hecho anteriormente comentado de que uno de los objetivos buscados es el de obtener conocimiento, en términos de dependencias entre atributos, o a un nivel más local, de asociaciones entre valores de atributos, acerca de un problema del mundo real de la forma más eficiente posible. Esto conlleva, en la gran mayoría de los casos, el hecho de tener que afrontar el que algunas de nuestras fuentes de información no sean todo lo precisas que deseáramos y, en consecuencia, hemos de actuar adecuadamente, admitiendo más flexibilidad en las estructuras mediante las cuales representaremos el conocimiento a extraer. Veremos cómo, en el apartado de esta memoria dedicado a la definición de conceptos previos, ya contamos con potentes herramientas, como por ejemplo, la ya citada Teoría de Subconjuntos Difusos, para tratar con este tipo de

inconvenientes.

También merecen ser destacados los estudios realizados sobre la extensión al caso difuso de algunas herramientas de minería de datos, tales como la definición se que hace en [Cubero et al., 1994a] para las dependencias funcionales difusas, o el establecimiento de una metodología para la extracción de reglas de asociación en bases de datos difusas [Delgado et al., 2003a]. Esta última herramienta, las Reglas de Asociación Difusas, constituirá uno de los pilares sobre los que basaremos los objetivos de esta memoria, que se exponen a continuación.

Objetivos

Todo lo anteriormente expuesto nos lleva a describir los objetivos que se pretenden abordar en esta memoria, y éstos son los siguientes:

- **Definición de Dependencia Aproximada Difusa.** El primer objetivo que nos proponemos es el de definir y desarrollar el concepto de Dependencia Aproximada Difusa, como extensión al caso difuso del concepto ya existente de Dependencia Aproximada. Para ello, partiremos del enfoque propuesto en [Blanco et al., 2000] para el caso "crisp". Además estudiaremos el problema de la implementación eficiente, con objeto de optimizar en la medida de lo posible los requerimientos necesarios en tiempo de ejecución y espacio de memoria.

  Dentro de este mismo apartado, describiremos un conjunto de situaciones tipo en las que aplicar esta metodología, para a continuación mostrar dos ejemplos concretos en los que aplicamos nuestros resultados en problemas reales que, de otra forma, serían mucho más difíciles de abordar.

- **Minería de Datos y Análisis de Correspondencias.** Como un segundo objetivo, propondremos el uso de Dependencias Aproximadas para el Análisis de Correspondencias entre Particiones. Estudiaremos el problema de concordancia de particiones difusas, que puede verse como una extensión del análisis de correspondencias clásico. Veremos que dicho problema puede formularse como un caso particular de la búsqueda de dependencias aproximadas difusas.

Plantearemos un ejemplo real y práctico, dentro de un entorno agrícola, en el que estudiaremos las relaciones que puedan existir entre distintas apreciaciones sobre un mismo concepto como es el suelo de cultivo.

- **Desarrollo de una aplicación software.** Por último, y para que estos resultados sean realmente eficaces, será necesario contar con una herramienta software que lleve a la práctica los resultados teóricos previamente obtenidos. Es por ello que otro de los objetivos presentados en esta memoria es el desarrollo de una aplicación que englobe la implementación, no sólo de los resultados presentados en esta memoria, sino también de otros necesarios para una buena gestión de un Sistema de Bases de Datos Relacionales Difusas.

Contenidos

La memoria está organizada como se presenta a continuación. En primer lugar, tras esta introducción, en la que se han expuesto de forma general los problemas que intentamos resolver y los objetivos que nos hemos propuesto realizar, será necesario definir algunos conceptos y resultados previos, necesarios para la comprensión del total de este trabajo, en lo que constituirá el segundo capítulo de la memoria.

En este segundo capítulo recordaremos conceptos tales como, por ejemplo, lo que entendemos por Base de Datos Relacional, con las definiciones que lleva asociadas, y por Minería de Datos y Extracción de Conocimiento en Bases de Datos, recordando igualmente algunos conceptos que nos serán muy necesarios con posterioridad. Este apartado está dedicado básicamente a la formulación y notación de los conceptos previamente expuestos.

En el tercer capítulo revisaremos algunos de los avances que han tenido lugar para extender al caso difuso el modelo de Bases de Datos Relacionales y las técnicas de Extracción de Conocimiento. Previamente, introduciremos algunos conceptos relevantes sobre la Teoría de los Subconjuntos Difusos, necesarios para comprender las definiciones posteriores.

En el siguiente capítulo, procederemos a definir el concepto de Dependencia Aproximada Difusa, enumerando sus propiedades y ventajas, y discutiendo

cómo superar los posibles inconvenientes. Describiremos las situaciones en las que este concepto nos puede resultar útil y expondremos algunos ejemplos prácticos de su aplicación. Presentaremos, asimismo, una colección de algoritmos con los pasos a seguir para obtenerlas de una forma lo más eficiente posible.

En el capítulo quinto introduciremos el análisis de correspondencias basado en el uso de reglas de asociación y dependencias aproximadas. Recordaremos los conceptos asociados al análisis de correspondencias clásico para después formular, en términos de herramientas de extracción de conocimiento, una alternativa a dicho análisis que, adicionalmente, permite ser aplicada a casos en los que la separación entre clases no es del todo precisa. Mostraremos y discutiremos algunos resultados obtenidos en un caso real.

Seguidamente, nuestra atención se dirigirá a la implementación, en forma de una aplicación software, de los resultados teóricos mostrados en la memoria. La aplicación se presenta como un asistente para la extracción de conocimiento, afectado o no de imprecisión, en bases de datos. Dicha herramienta se compone de varios módulos pensados para diversas tareas, con la idea de poder ser fácilmente extensible para abordar nuevos problemas,

Este capítulo se estructurará presentando en primer lugar un resumen histórico del ciclo de vida de nuestra aplicación, para pasar después a describir su funcionamiento mediante un sencillo manual de usuario, que incluimos como apéndice.

Por último, terminaremos la memoria con un apartado en el que, junto a las conclusiones sobre el trabajo realizado, presentaremos una serie de propuestas para las futuras líneas a seguir en nuestra investigación.

2. Preliminares (I): Conceptos generales

En el contexto de la representación y procesamiento de la información en un ordenador, el principal problema que debemos abordar es cómo representar dicha información, mediante qué tipo de estructuras, de forma que el acceso a la misma sea lo más cómodo y eficiente posible. Por otro lado, se han de facilitar técnicas y herramientas para poder procesar la información recogida con el objeto de extraer nuevo conocimiento que pueda resultar útil al usuario.

Este capítulo pretende poner al lector en situación, recordando algunos conceptos que nos serán de utilidad más adelante, a la hora de exponer los resultados de la memoria. Comenzaremos enumerando los conceptos más importantes asociados al modelo de Bases de Datos Relacionales, para, seguidamente, continuar con las definiciones más relevantes para nuestro trabajo relativas a la Minería de Datos y la Extracción de Conocimiento.

2.1. Modelo relacional de bases de datos

Entenderemos por Base de Datos un repositorio o conjunto de datos almacenados, normalmente en dispositivos electrónicos de un ordenador, y susceptibles de ser manejados para su consulta o actualización por un Sistema Gestor de Bases de Datos (SGBD). Éste se encarga de procesar las solicitudes de acceso a los datos por parte de los usuarios, ocupándose de ocultar detalles sobre la organización interna de la información. Adicionalmente, el SGBD se encarga de tareas como la privacidad de los datos o la eficiencia en el acceso a los mimos.

Las bases de datos comenzaron a popularizarse en los años 1970s y 1980s y en la actualidad su uso está enormemente extendido, tanto al nivel de gran empresa como al de usuario.

En todas las bases de datos se almacena información sobre determinadas entidades u objetos, y sobre las posibles relaciones existentes entre ellos. En la figura 2.1, podemos ver un sencillo esquema que ilustra de una forma muy básica cómo se pueden representar diversas entidades del mundo real en una base de datos, modelando previamente dichos objetos.

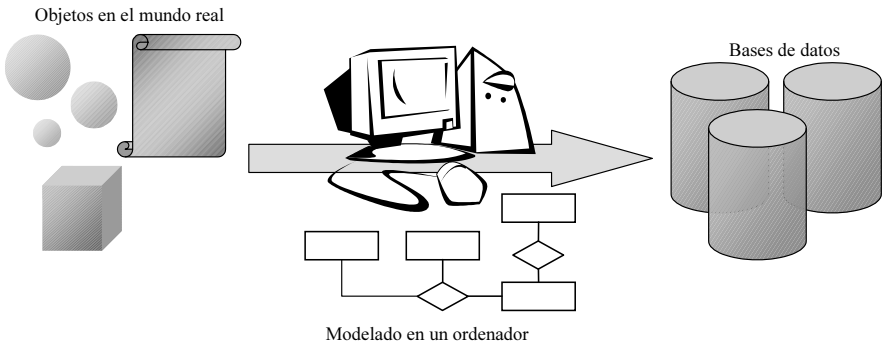


Figura 2.1: Representación del conocimiento en bases de datos

Existen varios modelos de bases de datos, pero los más conocidos y extendidos son los siguientes:

- **Modelo jerárquico:** Los datos se representan a través de una estructura de árbol, en la que los datos colocados en la zona inferior dependen o

están incluidos en los datos distribuidos en las zonas más altas del árbol. La limitación de este modelo radica en el hecho de que sólo permite representar directamente relaciones de uno a muchos.

- **Modelo en red:** Los datos se pueden ver como en el modelo anterior, el jerárquico, pero ya sin limitación en lo que se refiere a su organización, de manera que los datos de las zonas inferiores pueden estar también relacionados con algunos de los superiores. De esta forma, se permite modelar mejor una correspondencia de muchos a muchos.

- **Modelo relacional:** En este modelo, todos los datos son vistos en forma de tabla o relación. Es el modelo más extendido por su facilidad de comprensión y representación y será en el que nos centraremos a continuación.

Para un estudio más detallado de las materias aquí nombradas, recomendamos al lector la consulta de bibliografía especializada como, por ejemplo, [Ullman, 1989], [Date, 1990] o [Korth y Silberschatz, 1993].

En este apartado se recogerán aquellos aspectos más relevantes en lo concerniente al modelo de Bases de Datos Relacionales, y más concretamente, al presentado por Codd en 1970 [Codd, 1970]. Dicho modelo ha sido, y continúa siendo, una pieza clave dentro del ámbito de las Bases de Datos. La gran mayoría de las bases de datos existentes se ajustan en mayor o menor medida al modelo relacional. Éste ha ido evolucionando a lo largo de estos años, y se le han ido añadiendo nuevas características conforme ha ido siendo necesario, tales como la posibilidad de trabajar con objetos o su aplicación, como modelo multidimensional, en los sistemas de ayuda a la decisión.

Un **Sistema de Bases de Datos Relacional** (SBDR, en adelante) responde a dos características básicas:

(I) Los datos son percibidos por el usuario como tablas o relaciones.
(II) Los operadores de los que dispone el usuario (p.e., para consulta) generan como resultado de su aplicación nuevas tablas a partir de las originales. Por ejemplo, podemos considerar un operador que nos extraiga un subconjunto de filas de una tabla dada, mientras que otro operador distinto sea usado para obtener un subconjunto de columnas de dicha tabla. La

combinación de ambos subconjuntos puede verse, claramente, como una nueva tabla.

2.1.1. Estructuras de datos en un SBDR

La estructura de los datos viene determinada por el concepto de relación. Para llegar a entender dicho concepto, es necesario introducir al lector en algunos términos que también forman parte de la definición de la estructura del Modelo Relacional:

- Llamaremos **Atributo** a cualquier elemento de información del "mundo" que vamos a representar susceptible de tomar valores. También puede definirse como una de las características que describen a una entidad u objeto del que guardamos información. Se corresponde con lo que habitualmente denominamos **Campo** o **Columna** en una base de datos. Son ejemplos de atributo elementos tales como nombre, sueldo, edad de una persona, o potencia, número de plazas y matrícula de un coche.

- Llamaremos **Dominio**, notado como D_i, al conjunto de valores que puede tomar un determinado atributo A_i. En la bibliografía, también se suele notar el dominio de un atributo A_i como $dom(A_i)$, reforzando la relación existente entre dominio y atributo. El dominio de un atributo se suele considerar finito (en la práctica, en la teoría, por ejemplo, el dominio de un atributo numérico puede ser todo el conjunto de los números reales, $\Re$). Dos atributos pueden tener el mismo dominio, o estar incluido el dominio de uno de ellos en el dominio del otro.

  Es importante tener en cuenta que el dominio de un atributo no coincide con el tipo de dato de dicho atributo cuando se implementa una base de datos relacional. La mayoría de los sistemas de bases de datos relacionales no soportan correctamente el concepto de dominio y sólo ofrecen tipos de datos. Sólo en caso de sistemas muy elaborados se soporta este concepto, proporcionando una definición de dominio y permitiendo especificar que dos atributos tengan el mismo dominio. No obstante, el concepto está claro desde el punto de vista teórico y es posible fijar el dominio de un atributo a través de la integridad.

Una característica inicial que se le exigía a los valores del dominio era la atomicidad, en el sentido de que no exista una descomposición de los mismos que aporte significado. No obstante, esta premisa se ha ido relajando con la evolución del modelo, y actualmente se pueden encontrar en la literatura muchas matizaciones sobre este aspecto. Por último, cabe destacar que un dominio puede llevar asociado un conjunto de operadores específicos del mismo. Por ejemplo, un dominio esencialmente numérico contaría con operadores aritméticos y de comparación.

- Consideremos los atributos A_i, $i \in \{1, \ldots, n\}$ con dominios asociados D_i (no necesariamente distintos). Definimos la **Relación** asociada a los atributos $A_1, \ldots, A_n$, $R[A_1, \ldots, A_n]$ como cualquier subconjunto finito del producto cartesiano $D_1 \times \ldots \times D_n$.

  En una relación hay que considerar siempre dos aspectos diferentes:

  - *Esquema*: Es el conjunto de atributos $[A_1, \ldots, A_n]$ junto con sus dominios
  - *Instancia*: Es el conjunto de tuplas $r = \{t_1, \ldots t_m\}$ tal que $t_i = (x_1^i, \ldots, x_n^i)$, $\forall i \in \{1, 2, \ldots, m\}$ con $x_j^i \in D_j$.

  Es obvio que el esquema de una relación no varía habitualmente con el tiempo, pero una instancia de una relación es siempre variable, pudiendo estar vacía en una etapa concreta del desarrollo de la base de datos.

  Los valores m y n se denominan **cardinalidad** y **grado** de la instancia r, respectivamente. Como acabamos de decir en otras palabras, mientras que el grado se mantiene constante con el tiempo, la cardinalidad es susceptible de variar con éste.

- Una **Tupla** se define como cada una de las filas o registros de la instancia. Notaremos el valor de un atributo A_j para una determinada tupla t_i por $t_i[A_j]$.

- Una **Clave** (o **Llave**) **Primaria** se define como un subconjunto de atributos que identifican unívocamente a cada tupla de una determinada relación. En una relación, pueden existir varios de estos conjuntos, pero

únicamente se puede seleccionar uno de ellos como **Clave Primaria**. Al resto se les considera **Claves Candidatas** o **Alternativas**.

2.1.2. Propiedades de una relación

Algunas propiedades deseables en una relación son las siguientes:

1 No puede haber tuplas duplicadas. Siempre deberá existir una clave primaria. En el caso particular de las bases de datos transaccionales, representadas por medio del modelo relacional y que definiremos más adelante, se puede añadir un atributo en particular para que no se corresponda con uno de los ítems, sino que identifique unívocamente a cada transacción.
2 Los atributos no están ordenados. Esta propiedad proviene del hecho de que la cabecera de una relación es un conjunto.
3 De igual forma, las tuplas no se encuentran ordenadas. Esta propiedad también se apoya en que hayamos definido el cuerpo de una relación como un conjunto.
4 Los valores de los atributos son atómicos, en el sentido que utilizábamos cuando definíamos el concepto de dominio. Una relación que cumpla esta propiedad se dice que está Normalizada. Aunque a primera vista esta propiedad pueda suponer una fuerte restricción a la representación de información imprecisa, se pueden realizar lecturas de la misma que posibiliten la representación de este tipo de conocimiento sin ocasionar por ello una pérdida de validez del modelo. Este aspecto se tratará más adelante con más detalle.

2.1.3. Dependencia funcional

El diseño de una base de datos tiene como principales objetivos, no sólo almacenar información, sino también poder recuperar dicha información cuando sea requerido. Ambos objetivos han de cumplirse de la forma más eficiente posible. Pero para ello, es necesario que la distribución de los datos en tablas verifique una serie de restricciones, llamadas genéricamente restricciones de integridad. Dichas restricciones se han de determinar en la fase de análisis del problema real de acuerdo a las características de éste, y se tendrán en cuenta en la subsiguiente fase de diseño de la base de datos para, a partir de los

atributos, determinar las tablas o relaciones en las que se van a distribuir. Por tanto, estas restricciones son propiedades que se verifican a priori, y cuyo cumplimiento queremos mantener mediante un apropiado diseño de la base de datos.

Entre las diversas restricciones de integridad, podemos destacar las llamadas **dependencias funcionales**, definidas como sigue:

Definición 2.1.1 *Sea RE un esquema relacional, y sean $V, W \subseteq RE$. Decimos entonces que la dependencia funcional $V \rightarrow W$ se verifica en el esquema RE si y sólo si para cualquier instancia r de RE se cumple*

$$\forall t_1, t_2 \in r \text{ si } t_1[V] = t_2[V] \text{ entonces } t_1[W] = t_2[W] \tag{2.1}$$

Las dependencias funcionales que aparecen dentro de un esquema relacional suelen corresponderse con restricciones que se dan en el mundo real que estamos tratando de modelar.

- En ocasiones, se suele emplear la denominación de dependencia funcional en el sentido de dependencia inducida por los datos, para indicar con ello que una cierta dependencia se verifica en una relación concreta. Pero al no asegurarse entonces que la dependencia se cumpla en todas las relaciones de su mismo esquema relacional, no puede hablarse en estos casos de restricciones del mundo real.

- Sin embargo, si la tabla en cuestión contiene una gran cantidad de datos y éstos son muy representativos del dominio del mundo real que estamos modelando, se puede considerar que, mediante técnicas de extracción de conocimiento en bases de datos como las que definiremos más adelante, se ha descubierto una restricción hasta ahora desconocida.

Pero la obtención de dependencias funcionales en bases de datos es bastante difícil, debido a que las condiciones de existencia de estas dependencias son bastante restrictivas. De hecho, una sola excepción a la regla general (ecuación 2.1) hace que la dependencia ya no se cumpla. Por este motivo, y con diversos objetivos, la definición de dependencia funcional se puede extender mediante la introducción de algún tipo de imprecisión, incertidumbre o gradualidad, dando

lugar a diversos tipos de dependencias "suavizadas", que serán descritos con posterioridad en la memoria.

2.2. Bases de datos transaccionales

Al margen del modelo objeto-propiedades, en el que se basan la gran mayoría de las bases de datos existentes, y en particular los modelos que distinguíamos al inicio del capítulo, existen otras formas no estructuradas de representar información. Una de ellas es la llamada representación mediante "conjuntos de cosas". Existen varios casos presentes en el mundo real donde este tipo de representación resulta muy apropiada. Uno de ellos es el problema del carro de la compra. Un carro de la compra puede reducirse a un conjunto de elementos o ítems, en este caso artículos de un supermercado. Otro ejemplo muy común es el referido a la representación de los términos contenidos en un documento.

El estudio y análisis de colecciones de este tipo de conjuntos plantea algunos problemas como, por ejemplo, la extracción de asociaciones entre la presencia o no de dichos elementos, y que serán comentados más adelante.

Los dos ejemplos citados pueden formularse mediante un modelo matemático que se compone de los siguientes elementos o conceptos:

- En primer lugar, denominamos **ítem** al elemento básico del modelo. Sea $I = \{i_1, \ldots, i_n\}$ el conjunto total de ítems que estemos considerando. En los ejemplos anteriores, un ítem es uno cualquiera de los artículos que se pueden adquirir en un supermercado, o uno de los términos que pueden aparecer en un documento determinado.

- Una **transacción** t_i es todo conjunto de ítems relacionados entre sí por su presencia. Se cumple que $t_i \subseteq I$. En nuestros ejemplos, cada carro de la compra es una transacción, como también lo es cada documento.

- Por último, debemos introducir el concepto de **conjunto de transacciones**, como el objeto principal de estudio en este tipo de problemas. La tabla 2.1 nos muestra un conjunto de transacciones determinado.

Tabla 2.1: Un conjunto de transacciones

t#	ítems
t_1	{leche, huevos, harina, pescado}
t_2	{huevos, patatas, lechuga}
t_3	{huevos, harina, azúcar, levadura}
...	...

2.2.1. Representación de un conjunto de transacciones en una base de datos relacional

Como ya decíamos al inicio de este apartado, los conjuntos de transacciones constituyen una representación no estructurada de información, lo cual se traduce en que su almacenamiento en un ordenador puede resultar poco eficiente. Habida cuenta, además, de que el modelo relacional está muy extendido y es el más utilizado en las bases de datos actuales, resulta razonable pensar en una forma eficaz de representar conjuntos de transacciones en una base de datos relacional. Para ello, contamos con dos posibilidades:

- Por un lado, podemos representar cada una de las posibles ocurrencias <*transacción, ítem*> de nuestro conjunto como una tupla perteneciente a una relación de dos columnas, como la mostrada en la tabla 2.2.

- En segundo lugar, podemos optar por otra representación tabular en la que cada fila corresponde a una transacción y la tabla o relación tiene tantas columnas como ítems posibles. En dicho caso, el dominio para cada atributo es el conjunto $\{0, 1\}$, en el sentido de que se indica la presencia o ausencia del ítem asociado. Tenemos un ejemplo de lo anteriormente expuesto en la tabla 2.3. "Grosso modo", una **Base de Datos Transaccional** podría considerarse como un caso particular de una base de datos relacional en el que cada atributo representara el caso de presencia o ausencia de un ítem de la transacción.

Ambas representaciones son igualmente válidas y manejables, si bien la segunda puede resultar más difícil de llevar a la práctica en casos en los que el

Tabla 2.2: Representación de las ocurrencias <*transacción*, *ítem*> en un conjunto de transacciones

$t\#$	ítem
t_1	i_1
t_1	i_2
t_1	i_3
t_1	i_4
t_2	i_2
t_2	i_5
t_2	i_6
t_3	i_2
t_3	i_3
t_3	i_7
t_3	i_8

Tabla 2.3: Representación de un conjunto de transacciones en una tabla relacional

t#	leche	huevos	harina	pescado	patatas	lechuga	azúcar	levadura
t_1	1	1	1	1	0	0	0	0
t_2	0	1	0	0	1	1	0	0
t_3	0	1	1	0	0	0	1	1

número de ítems considerados sea muy elevado, por las limitaciones que pueda tener el SGBD en cuanto a número máximo de columnas.

2.2.2. Representación de una base de datos relacional mediante un conjunto de transacciones

El caso contrario es aquél en el que podemos representar una relación clásica por medio de una relación transaccional. El proceso es muy sencillo, pero puede llegar a resultar costoso en espacio, en función del tamaño de los dominios. Básicamente, consiste en definir una columna para la relación transaccional por cada posible par $< Atributo, valor >$ en la relación original, y rellenar la segunda relación en consecuencia. Detallamos dicho procedimiento en el algoritmo B.1 (página 236), y a continuación mostramos un ejemplo en el que transformamos una relación normal y corriente (tabla 2.4) en una relación

transaccional (tabla 2.5).

Como vemos, ambas relaciones contienen el mismo número de tuplas (cardinalidad), pero no ocurre así con el número de columnas (grado), que puede dispararse en aquellos casos en los que los dominios de los atributos sean muy extensos o no finitos (por ejemplo, el conjunto de los números reales $\Re$). Más adelante, veremos cómo se pueden afrontar este tipo de inconvenientes en problemas que hagan uso de relaciones transaccionales, como es el caso de las reglas de asociación sobre atributos cuantitativos.

Tabla 2.4: Una relación r

Nombre	Edad
Juan	23
Teresa	23
David	26

Tabla 2.5: La relación transaccional T resultante de transformar la relación de la tabla 2.4

$Nombre = Juan$	$Nombre = Teresa$	$Nombre = David$	$Edad = 23$	$Edad = 26$
1	0	0	1	0
0	1	0	1	0
0	0	1	0	1

2.3. Minería de datos y Extracción de conocimiento

Entendemos por minería de datos (o *data mining*) un proceso complejo para la extracción de información implícita, desconocida anteriormente y posiblemente útil (tal como reglas, restricciones, etc.) a partir de una base de datos.

Mediante la minería de datos puede extraerse e investigarse conocimiento de interés, patrones regulares o información de alto nivel de los conjuntos de datos adecuados. Podemos ver las bases de datos como fuentes seguras

y abundantes de generación y verificación de conocimiento. Muchos investigadores han considerado a la extracción de información y conocimiento de grandes bases de datos como un tema clave en sistemas de bases de datos y en aprendizaje automático (*machine learning*), y muchas compañías industriales lo ven como un área importante de cara a obtener mayores beneficios. El conocimiento extraído puede aplicarse en la gestión de información, procesamiento de demandas, toma de decisiones, control de procesos y otras muchas aplicaciones. Los investigadores de muchos campos, tales como sistemas de bases de datos, sistemas basados en el conocimiento, inteligencia artificial, aprendizaje de máquinas, estadística y visualización de datos, han mostrado un gran interés en el *data mining*. Más aún, algunas aplicaciones que suministran información, como los servicios en línea, también hacen uso de varias técnicas de *data mining* para comprender mejor el comportamiento del usuario, para mejorar el servicio que prestan y para incrementar las oportunidades de negocios.

Para un estudio más amplio sobre estas metodologías, recomendamos al lector algunos trabajos como [Imielinski y Mannila, 1996], [Fayyad et al, 1995], [Frawley et al, 1991] o [Wright, 1998], por citar algunos de los más conocidos.

Otra opción bastante recomendable es la de visitar el sitio web de KD-Net (`http://www.kdnet.org/control/`), la Red de Excelencia Europea sobre Extracción de Conocimiento, donde el lector podrá encontrar sobrada información y recursos sobre Extracción de Conocimiento, Minería de Datos y Aprendizaje Automático en general.

La primera definición para Extracción de conocimiento que encontramos proviene de [Frawley et al, 1991], donde la Extracción de conocimiento es *"Un proceso no trivial de identificación de patrones en los datos válidos, novedosos, potencialmente útiles y comprensibles"*. La figura 2.2 muestra el proceso general de la extracción de conocimiento en bases de datos.

El término patrones hace referencia a información capaz de ser comprendida y utilizada directamente por el usuario, o bien por otra aplicación, como por ejemplo, un sistema experto o un analizador semántico.

Los datos han de ser válidos y justificables y han de resultar interesantes. Es decir, no debían conocerse antes y no debían resultar obvios. Aparte, si esos

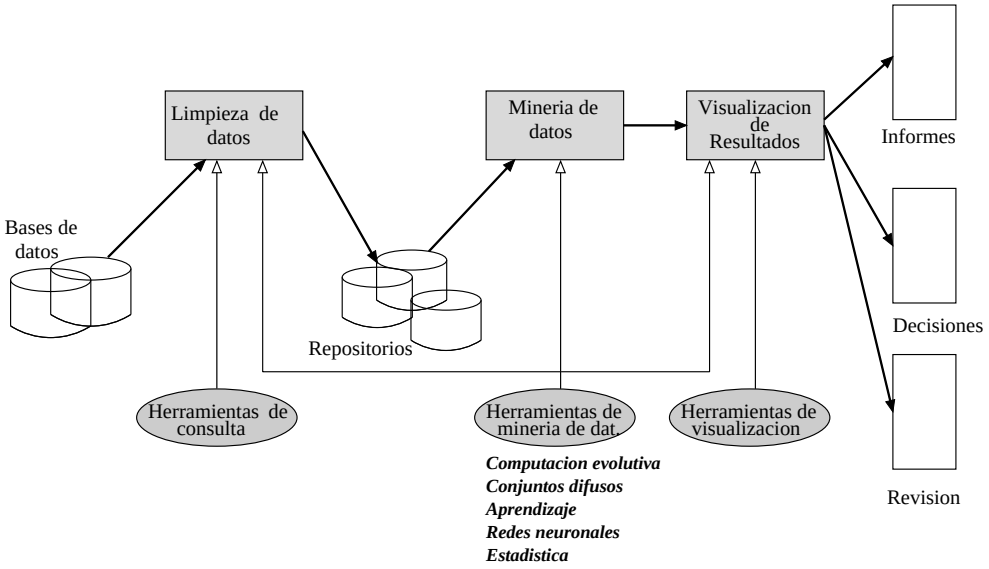


Figura 2.2: Esquema general del proceso de extracción de conocimiento en bases de datos

datos van destinados a ser presentados a un usuario, han de ser comprensibles por él. Para ello, deberían de poder expresarse mediante algún lenguaje de alto nivel.

Un problema asociado a la extracción de nuevo conocimiento es que, al trabajar con datos masivos, el coste en tiempo (además del coste en espacio que conlleva) puede ser inviable. Por esta razón, los métodos de extracción de conocimiento han de resultar eficientes y proporcionar una buena respuesta en un tiempo aceptable.

2.3.1. Niveles en la extracción de conocimiento

Podemos apreciar tres niveles o tipos en la Extracción de conocimiento:

- **No dirigida o pura:** No se aplica ningún tipo de restricción ni consideración especial sobre el resultado que se espera. No se busca nada en concreto. Sencillamente se analizan los datos en bruto, como un primer paso para posteriores refinados.
- **Dirigida:** Una vez que se tiene información sobre el tipo de datos

que pueden extraerse, el proceso puede redirigirse hacia un dominio de búsqueda concreto. En este nivel se pretenden encontrar relaciones de posible interés entre los atributos.

- **De contraste y refinamiento de hipótesis:** Es el nivel de mayor concreción. El dominio de búsqueda es mucho más específico. El objetivo es ahora comprobar, normalmente de forma estadística, la veracidad de una hipótesis expuesta sobre los datos.

2.3.2. Factores que intervienen en la extracción de conocimiento

Una segunda definición para Extracción de conocimiento nos viene dada en [Fayyad et al, 1995], de la siguiente forma: *"Proceso de uso de una base de datos para cualquier consulta que se requiera; incluyendo: preprocesamiento, muestreo y transformaciones, aplicación de técnicas de minería de datos para obtener patrones y la evaluación de los resultados de dicha minería para identificar qué patrones se consideran conocimiento"*.

Los métodos de Extracción de conocimiento suelen apoyarse en los siguientes factores:

- La Extracción de conocimiento se aplica frecuentemente sobre grandes bases de datos, y requiere de técnicas eficientes para el tratamiento de grandes volúmenes de datos.
- La Extracción de conocimiento está vinculada con otras muchas metodologías, ya sea basándose en ellas o al contrario, constituyendo una herramienta para aplicar. Por ejemplo, Estadística, Aprendizaje máquina, Análisis de regresión, etc.
- La Extracción de conocimiento está fundamentada en la interacción hombre-máquina. A pesar de la capacidad de cálculo de los ordenadores actuales, todavía se necesita de un experto con capacidad para supervisar el proceso sobre los datos.

Un elemento determinante de cara a extraer conocimiento es la base de datos. En función de las característica de ésta, el proceso de Extracción de

conocimiento se verá favorecido o no. Algunos de los factores que influyen en esto son:

- **Datos dinámicos:** Por regla general, una base de datos se encuentra en continua actualización. El proceso de Extracción de conocimiento ha de poder ampliar el conocimiento descubierto con la nueva información que sea introducida en la base de datos.

- **Campos irrelevantes:** En función del contexto, se puede pretender extraer un tipo específico de conocimiento. En tales casos, en la base de datos existirán campos prescindibles, redundantes o poco significativos, que poco o nada podrán aportar a la respuesta esperada del sistema. Por tanto, podrán omitirse, agilizando el proceso ahora que no tiene que tenerlos en cuenta.

- **Valores perdidos:** La presencia o ausencia de determinados valores puede afectar substancialmente al proceso de descubrimiento, en función de la importancia dada a esos valores, de acuerdo al punto anterior.

- **Ruido e incertidumbre:** También la presencia de valores erróneos o vagos puede influir en el proceso de extracción de conocimiento. En tales casos, puede ser apropiado aplicar técnicas estadísticas para el estudio en conjunto de los datos, con objeto de atenuar el impacto de los errores.

- **Campos perdidos:** Al igual que los campos irrelevantes pueden entorpecer la extracción de conocimiento, también puede hacerlo la no presencia de campos erróneamente omitidos.

El conocimiento obtenido a la salida de uno de estos métodos cuenta con varias características:

- **Formato:** Podemos clasificar el conocimiento según el formato que presente. Podemos distinguir entre patrones dentro de un mismo campo, dentro de un mismo registro, etc. Otra categorización puede hacerse de acuerdo a la capacidad descriptiva del conocimiento. Así, puede ser de tipo cuantitativo o cualitativo.

- **Representación:** El conocimiento obtenido ha de poder ser representado de acuerdo al usuario final al que va destinado. Si el destinatario es otro módulo o programa, se han de seguir unas normas de formato compatibles entre unos programas y otros. Si por el contrario, el destinatario es un usuario, la información ha de serle presentada en una forma que pueda comprender. El uso del lenguaje natural es muy deseable en estos casos. E, incluso, la representación visual puede ser de gran ayuda en determinados ámbitos.

- **Incertidumbre:** A menudo, ocurrirá que los datos obtenidos aparezcan con cierto grado de vaguedad. Por ejemplo, si la base de datos no es completa o aparece ruido dentro de la información. Se ha de tratar esa incertidumbre de manera apropiada. En referencia al apartado anterior, el lenguaje natural lleva asociado muchas veces un alto grado de imprecisión. La teoría de conjuntos difusos [Zadeh, 1979a] o la teoría de conjuntos "rugosos" (*rough sets*) [Pawlak, 1984] proporcionan herramientas para manejar este tipo de problemas.

- **Muestreo de datos:** Relacionado con el punto anterior, también puede ocurrir que el tamaño de la base de datos impida trabajar con todos ellos. Para solucionarlo, debemos aplicar algún método que nos permita reducir el número de datos con los que trabajar. Técnicas de muestreo y herramientas estadísticas son de gran utilidad en estos casos.

Actualmente existe una amplia variedad de técnicas para la Extracción de conocimiento. Entre ellas podemos destacar algunos ejemplos:

- **Agrupamiento.** Su objetivo es encontrar grupos (*clusters*) entre los datos, de acuerdo a sus características. Se englobaría en el primer tipo de Minería de datos, la no dirigida.

- **Modelos de regresión.** Aplican el análisis de regresión clásico sobre un conjunto de datos con el objetivo de describir esos datos mediante una función.

- **Clasificación.** Partiendo de un conjunto de clases, clasificar nuevos

datos en las clases existentes, es decir, obtener los criterios que determinan a dichas clases.

- **Resumen.** Se trata de caracterizar al conjunto de datos mediante el menor número posible de atributos o características.

- **Análisis de enlaces.** Intenta encontrar relaciones y dependencias entre los datos, de acuerdo a sus características. En determinados casos, puede resultar de gran interés encontrar si existe algún grado de correlación entre los atributos de los datos.

- **Análisis de secuencias.** Esta técnica está indicada para problemas de modelado de datos secuenciales.

2.3.3. Campos de aplicación

La Extracción de conocimiento está estrechamente relacionada con otros campos:

- **Gestión de Bases de datos:** Un sistema de gestión de bases de datos proporciona procedimientos para el almacenamiento, acceso y modificación de la información. Aunque esas operaciones son muy básicas de por sí, constituyen una base sobre la que se apoya la Extracción de conocimiento para trabajar con los datos a un nivel superior.

- **Sistemas expertos:** Un sistema experto intenta simular el comportamiento de un experto real ante un problema específico. Para ello, necesita trabajar con una base de conocimiento, en la que la información esté almacenada de una forma concreta, permitiendo trabajar con ella eficientemente, para poder obtener unas respuestas compresibles de cara al usuario.

- **Sistemas estadísticos:** Tradicionalmente, la Estadística ha constituido una buena herramienta teórica para el análisis de datos. Pero en nuestro problema, esto no es suficiente. Aun así, la Extracción de conocimiento se apoya fuertemente en estas herramientas.

- **Descubrimiento científico:** Aunque en realidad, son dos tipos de descubrimientos distintos, la Extracción de conocimiento todavía tiene algunos puntos en común con el descubrimiento científico. El descubrimiento científico está más enfocado al tratamiento de datos empíricos y al estudio de cómo influye en los datos la variación de parámetros. Mediante técnicas de Extracción de conocimiento se podrían descubrir ese tipo de relaciones.

Las aplicaciones de las técnicas comentadas comprenden un elevado conjunto de áreas, en las que el manejo de enormes cantidades de información está a la orden del día. Así, podemos destacar de entre todas las siguientes: Medicina, Finanzas, Agricultura, Política y Demografía, Marketing y ventas, Compañías de Seguros, Ingeniería, Ciencias Físicas y Químicas, Aplicaciones militares, Ciencia Aeroespacial e incluso Prensa y Publicidad.

Como hemos visto, La minería de datos presenta muchos temas de investigación, y existen multitud de estudios dedicados a inventar nuevos métodos de data mininig o desarrollar técnicas integradas para una extracción de conocimiento eficiente y efectiva. En el contexto de esta memoria, nos vamos a centrar en dos conocidas técnicas de minería de datos: **Reglas de Asociación** y **Dependencias Aproximadas**.

2.3.4. Reglas de asociación

En [Agrawal et al., 1993] se introduce por primera vez el concepto de Regla de Asociación. En líneas generales, se trata de buscar las posibles relaciones existentes entre la presencia o no de ítems en determinados conjuntos de transacciones. Se trata por tanto de un concepto muy arraigado a su vez en el concepto de base de datos transaccional que definíamos en el apartado 2.2.

Formalmente, sea $I = i_1, i_2, \ldots, i_m$ un conjunto de ítems. Sea T un T-set (usando la terminología inglesa, si bien también podemos referirnos al mismo como hemos hecho hasta ahora, conjunto de transacciones). Sea X un **ítemset** (conjunto de ítems) de I, esto es, $X \subseteq I$. Decimos que una transacción $t \in T$ satisface X si todos los ítems i_k de X aparecen en t, o lo que es lo mismo, $X \subseteq t$. Partiendo de estos conceptos, podemos definir a continuación lo que se entiende por regla de asociación.

Definición 2.3.1 ([Agrawal et al., 1993]) *Una **Regla de Asociación** es una implicación de la forma $X \Rightarrow Y$, donde $X, Y \subset I$ y $X \cap Y = \emptyset$, entendiendo con ello que toda transacción que satisface X satisface también Y.*

De manera informal, podemos definir las reglas de asociación como "implicaciones" que establecen una relación entre los ítemsets que aparecen en un determinado número de transacciones del T-set. Se dice entonces que una regla de asociación se cumple en el T-set. El ejemplo clásico al que se suele hacer referencia es el de los clientes de un supermercado, donde consideramos que los ítems son artículos que podemos adquirir en el supermercado, y cada transacción se corresponde con el carro de la compra de un cliente determinado. En este contexto, las reglas de asociación nos informan sobre la presencia de ítems dentro de un mismo carro, lo que podemos expresar, por ejemplo, como "todo cliente que compra mantequilla y huevos, compra también harina". Esta sentencia en lenguaje natural puede ser formulada mediante la siguiente notación: $mantequilla, huevos \Rightarrow harina$. Siguiendo esta notación, denominamos **antecedente** a la parte izquierda de la regla, y **consecuente** a la parte derecha.

2.3.4.1. Valoración de las reglas de asociación

En problemas del mundo real, es muy difícil hallar implicaciones de este tipo que se cumplan con total certeza en un conjunto de transacciones. Es por ello que, puesto que aquellas reglas con un alto nivel de certeza nos siguen resultando interesantes, nos planteamos el medir de alguna forma la certeza de las reglas que podamos obtener.

Más aún, no podemos regirnos sólo por la certeza en que se cumple una regla. Incluso si ésta es muy alta, puede ocurrir que el conjunto de transacciones donde esto ocurra sea tan pequeño que la regla deje de resultarnos interesante. De aquí que tengamos que establecer al menos dos medidas para una regla de asociación, una asociada a su certeza, y otra relativa a su interés de cara al usuario final. Dicho interés puede venir asociado al porcentaje de transacciones en las que la regla se cumple.

Con este propósito en mente, en [Agrawal et al., 1993] se propusieron las medidas de **confianza** y **soporte**, obtenidas ambas a partir del soporte de

un ítemset. Entenderemos por soporte de un ítemset el porcentaje de transacciones en las que dicho ítemset aparece (aunque también se pueda expresar mediante un valor entre 0 y 1).

Definición 2.3.2 ([Agrawal et al., 1993]) *Dado un conjunto de ítems I y un conjunto de transacciones T sobre I, el **soporte** de un **ítemset** $I_0 \subseteq I$ se expresa como,*

$$supp(I_0) = \frac{|\{\tau \in T | I_0 \subseteq \tau\}|}{|T|} \tag{2.2}$$

o, lo que es lo mismo, la probabilidad de que el ítemset aparezca en una transacción de T, entendiendo $|\cdot|$ como el cardinal o número de elementos del conjunto.

Definición 2.3.3 ([Agrawal et al., 1993]) *El **soporte** de una regla de asociación $I_0 \Rightarrow I_1$ en T equivale a la probabilidad de la aparición conjunta de I_0 e I_1, $I_0 \cup I_1$, y se calcula por tanto como,*

$$Supp(I_0 \Rightarrow I_1) = supp(I_0 \cup I_1) = \frac{|\{\tau \in T | I_0 \cup I_1 \subseteq \tau\}|}{|T|} \tag{2.3}$$

Es decir, el soporte de una regla se define como la proporción de tuplas (o transacciones) en las que la regla se cumple.

La medida clásica para la certeza o precisión de una regla es la confianza, que se define como la probabilidad de aparición del ítemset condicionada a la aparición del antecedente.

Definición 2.3.4 ([Agrawal et al., 1993]) *Dada una regla de asociación $I_0 \Rightarrow I_1$, calculamos su **confianza** como*

$$Conf(I_0 \Rightarrow I_1) = \frac{supp(I_0 \cup I_1)}{supp(I_0)} = \frac{Supp(I_0 \Rightarrow I_1)}{supp(I_0)} \tag{2.4}$$

Las medidas de soporte y confianza evalúan el interés y el grado de cumplimiento de las reglas de asociación desde un punto de vista meramente estadístico. El soporte mide el número de casos en los que la regla se verifica en el conjunto de transacciones y por tanto puede verse como una medida de la importancia

de la regla. La confianza mide la probabilidad del consecuente condicionada al antecedente, y es por tanto una medida de implicación.

El uso del soporte está bastante generalizado y se acepta como la mejor opción para medir la importancia, ya que reúne una serie de ventajas:

- Es adecuado para la tarea de medir la relevancia estadística de una regla.
- Es una medida con significado intuitivo, y por tanto la interpretación de los valores de soporte es relativamente sencilla para el usuario.
- El uso del soporte contribuye al diseño de algoritmos eficientes para la búsqueda de reglas de asociación.

Sin embargo, la confianza como medida del grado de asociación, implicación o dependencia entre antecedente y consecuente ha recibido diversas críticas. Los argumentos en contra de esta medida son los siguientes:

- La confianza no mide adecuadamente el grado de independencia estadística entre el antecedente y el consecuente.
- La confianza no refleja la dependencia negativa entre antecedente y consecuente.
- Por último, la confianza es una medida de probabilidad conficionada. La probabilidad condicionada no es intuitiva, y por esta razón resulta difícil para un usuario no experto establecer umbrales mínimos de confianza semánticamente significativos a la hora de obtener reglas de asociación.

Por los motivos expuestos, algunos autores han propuesto medidas alternativas la confianza, tales como:

- El **interés** [Silverstein et al., 1998] trata de medir el grado de dependencia/independencia entre el antecedente y el consecuente de la regla. Un valor de la medida de interés de 1 indica total independencia. Un valor superior a 1 indica dependencia positiva. Un valor inferior a 1 indica dependencia negativa. Se define para una regla $X \Rightarrow Y$ como

$$Inter(X,Y) = \frac{p(A \wedge B)}{p(A)p(B)} \tag{2.5}$$

- Otra medida, que estudia los grados de dependencia negativa entre antecedente y consecuente, es la **convicción**, que se define en [Brin et al., 1997] como

$$Conv(X \Rightarrow Y) = \frac{p(A)p(\overline{B})}{p(A \wedge \overline{B})} \tag{2.6}$$

En [Sánchez, 1999, Berzal et al., 2001a] se discute con más detalle el uso de estas y otras medidas para abordar los problemas asociados al uso del soporte y la confianza. Concretamente, en dichos trabajos se propone el **Factor de Certeza** de Shortliffe y Buchanan [Shortliffe y Buchanan, 1975] como una alternativa a la confianza, y que será la medida que usemos para valorar las reglas de asociación en lo sucesivo.

Definición 2.3.5 ([Berzal et al., 2001a]) *El factor de certeza de una regla de asociación $I_0 \Rightarrow I_1$ se define como,*

$$CF(I_0 \Rightarrow I_1) = \frac{(Conf(I_0 \Rightarrow I_1)) - supp(I_1)}{1 - supp(I_1)} \tag{2.7}$$

si $Conf(I_0 \Rightarrow I_1) > supp(I_1)$, y

$$CF(I_0 \Rightarrow I_1) = \frac{(Conf(I_0 \Rightarrow I_1)) - supp(I_1)}{supp(I_1)} \tag{2.8}$$

si $Conf(I_0 \Rightarrow I_1) < supp(I_1)$, o 0 en otro caso.

El factor de certeza toma valores en $[-1, 1]$, indicando el grado en el que nuestra creencia de que el consecuente es cierto varía cuando el antecedente es también cierto. Se mueve desde 1, el máximo incremento (esto es, si I_0 es cierto, I_1 también lo es), hasta -1, que indica el máximo decremento.

2.3.4.2. Extracción de reglas de asociación

El problema de extraer reglas de asociación puede dividirse en dos subproblemas:

1 Generar todas las combinaciones de ítems, entendiendo éstas como ítemsets, con un soporte por encima de cierto umbral, previamente definido,

Tabla 2.6: Conjunto de transacciones o T-set, T

	vino	*cerveza*	*jamón*	*calamares*	*pincho*
t_1	0	1	0	0	1
t_2	0	1	0	0	1
t_3	1	0	1	0	0
t_4	0	1	0	0	1
t_5	1	0	1	0	0
t_6	0	1	0	0	1
t_7	0	1	0	0	1
t_8	0	1	0	0	1
t_9	1	0	1	0	0
t_{10}	0	1	0	0	1
t_{11}	0	1	0	1	0
t_{12}	0	1	0	0	1
t_{13}	0	1	0	1	0
t_{14}	0	1	0	0	1
t_{15}	0	1	0	0	1
t_{16}	0	1	0	0	1

al que llamaremos soporte mínimo (*minsupp*). Esas combinaciones suelen encontrarse en la literatura con el nombre de **ítemsets frecuentes**.

2 Dado un ítemset frecuente $Y = i_1, i_2, \ldots i_k, k \geq 2$, generar todas las reglas que contengan todos los ítems de ese ítemset. Para ello, se toman todos los subconjuntos no vacíos X de Y y se generan las reglas $X \Rightarrow Y - X$ que cumplan que su confianza es mayor que cierto umbral al que llamaremos confianza mínima (*minconf*). El valor de la confianza viene dado por

$$\frac{soporte(Y)}{soporte(X)}$$

Como Y es frecuente, cualquier subconjunto suyo también será frecuente ([Agrawal et al., 1993]), así que tendremos almacenado su soporte. Las reglas que se deriven de Y satisfarán la restricción del soporte, pues Y así lo hace.

En nuestro caso, y como acabamos de decir, hemos de substituir este criterio de mínima confianza por el de mínimo factor de certeza, aunque al fin y al cabo éste se calcula a partir del valor de confianza.

2.3.4.3. Un ejemplo sencillo

Consideremos el conjunto de transacciones definido por la tabla 2.6, donde se nos muestra, por ejemplo, las preferencias de los clientes de un bar en cuanto a la relación bebida/tapa. Una propuesta interesante de cara al propietario sería conocer algo más sobre las preferencias que puedan tener sus clientes habituales, de cara a proporcionar un mejor servicio a sus futuros clientes.

Por medio de un sencillo algoritmo como el que se describe más adelante, nos sería posible extraer todas las reglas que aparecen en la tabla 2.7. Podemos aventurar a dar una interpretación de estos resultados como que, por ejemplo, los amantes del buen vino son fieles al montadito de jamón, y viceversa, mientras que los clientes que toman cerveza prefieren en su mayoría acompañar ésta con un pincho moruno.

Tabla 2.7: Ejemplo de reglas de asociación

$[jamón] \Rightarrow [vino], supp$ 18,75 %, $conf$ 100,0 %, CF 1,0
$[vino] \Rightarrow [jamón], supp$ 18,75 %, $conf$ 100,0 %, CF 1,0
$[calamares] \Rightarrow [cerveza], supp$ 12,5 %, $conf$ 100,0 %, CF 1,0
$[cerveza] \Rightarrow [calamares], supp$ 12,5 %, $conf$ 15,38 %, CF 0,03
$[pincho] \Rightarrow [cerveza], supp$ 68,75 %, $conf$ 100,0 %, CF 1,0
$[cerveza] \Rightarrow [pincho], supp$ 68,75 %, $conf$ 84,61 %, CF 0,51

Observando la tabla de transacciones original y las reglas de asociación obtenidas, se aprecia una total certeza ($CF = 1$) entre la relación bidireccional existente entre vino y jamón, o la que existe entre las tapas de calamares o pincho y la cerveza. Sin embargo, en los casos en los que la certeza no es absoluta, se puede ver que es bastante alta la diferencia entre la información que nos aporta la confianza y la que nos proporciona el factor de certeza. En el caso de la regla $[cerveza] \Rightarrow [calamares]$, sin ir más lejos, debido al bajo soporte del consecuente, el factor de certeza es muy cercano al 0, frente

al valor de confianza, por lo que de seguir el criterio del factor de certeza, descartaríamos con más seguridad dicha regla.

2.3.4.4. Mejoras al problema original

El principal inconveniente que presenta la extracción de reglas de asociación estriba en que, en bases de datos lo suficientemente voluminosas, los costes tanto en tiempo como en espacio pueden resultar inviables. Por un lado, para lograr nuestro objetivo hemos de trabajar con todos los ítemsets posibles. Si tenemos m ítems, esto quiere decir que habremos de considerar 2^m posibles ítemsets. Afortunadamente, los algoritmos existentes, que se comentarán más adelante, aplican técnicas heurísticas para reducir en la medida de lo posible el número de ítemsets que se considerarán, de acuerdo a la estimación de si podrán o no ser frecuentes.

Aún contando con esa capacidad de los algoritmos para disminuir los requerimientos en tiempo de proceso y en espacio de memoria, mejorando en definitiva la eficiencia del procedimiento, todavía nos podemos encontrar con otros problemas, ésta vez asociados a la aplicación que les pueda dar el usuario final. El conocimiento obtenido es muy dependiente del contexto al que pertenece la información contenida en la base de datos original. Por este motivo, suele ser conveniente y necesaria la intervención de un experto humano que pueda dar una interpretación de las reglas obtenidas, indicando cuáles de ellas son potencialmente útiles y cuáles no, debido por ejemplo a su trivialidad.

Pero la labor del experto humano puede verse entorpecida si el conjunto de reglas obtenido es demasiado amplio. Es por eso que, de cara a optimizar la obtención y posterior interpretación de reglas de asociación dentro de una base de datos se pueden establecer ciertas restricciones:

- **Restricciones sintácticas**. Estas restricciones limitan los ítems que pueden aparecer en una regla. Por ejemplo, podemos estar interesados sólo en las reglas que tengan un ítem específico en el consecuente o en el antecedente. También se pueden realizar combinaciones de las restricciones anteriores (como especificar que las reglas tengan ciertos ítems en el antecedente y otros en el consecuente).

- **Restricciones de soporte**. Podemos estar interesados sólo en las reglas

cuyos ítems aparezcan en un porcentaje de las tuplas de T por encima de un soporte mínimo. Esto quiere decir que para que la información que nos da la regla tenga cierto peso es necesario que aparezca con cierta frecuencia en la base de datos.

- **Restricciones de cumplimiento**. El factor de certeza nos da una medida de la fuerza de una regla. Nos informa sobre la dependencia entre la aparición del consecuente si aparece el antecedente. En general, nos interesa que el factor de certeza supere un mínimo.

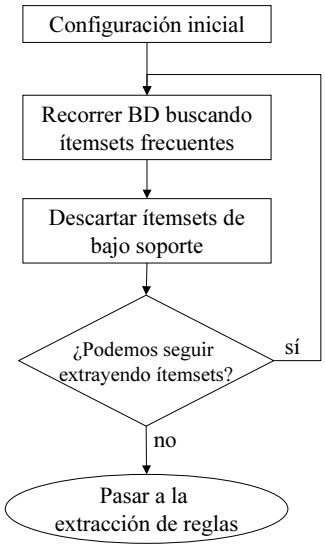
2.3.4.5. Algoritmos

Como hemos apuntado con anterioridad, el proceso de extracción de reglas de asociación comprende dos partes principales. En una primera se han de obtener los ítemsets frecuentes o interesantes a partir de la base de datos. Ésta primera fase suele ser la más estudiada con objeto de optimizar este cálculo cuanto sea posible, ya que, al ser la fase durante la cual se accede a la base de datos, es la que más recursos consume, tanto en tiempo como en espacio de memoria.

En la literatura, encontramos diversos algoritmos para la extracción de reglas de asociación. Los más básicos son SETM [Houtsma y Swami, 1993] y AIS [Agrawal et al., 1993]. Éste último fue adaptado posteriormente y dio lugar al algoritmo Apriori (presentado junto con una optimización del mismo, Apriori-TID, en [Agrawal y Srikant, 1995]), siendo el más conocido por su versatilidad y simplicidad. La mayoría de los enfoques siguientes parten del funcionamiento básico de éste, y así encontramos los algoritmos OCD [Mannila et al., 1994] y DHP [Park et al., 1995]. Otros algoritmos más recientes son, por citar algunos de los más interesantes, DIC [Brin et al., 1997], CARMA [Hidber, 1999], TBAR [Berzal et al., 2001b] y FP-Growth [Han et al, 2000]. Para más información, en [Hipp et al., 2000] se puede encontrar una interesante recopilación de algoritmos de este tipo.

El algoritmo Apriori (algoritmo B.2, página 237) es uno de los más básicos, pero nos muestra los pasos necesarios para obtener el conjunto de ítemsets frecuentes, que además pueden verse en la figura 2.3. En este algoritmo, los ítemsets se consideran ordenados por tamaño. Primero se calculan los 1-ítem-

Figura 2.3: Ordinograma básico de cálculo de ítemsets frecuentes



sets, después los 2-ítemsets, etc. En cada iteración, se recorre la base de datos y se comprueba si los ítemsets aparecen en ella con suficiente soporte (para lo cual fijamos un umbral mínimo al que se suele llamar *minsupp*). Si no es así, se descartan. La función $CrearNivel(i, L)$ se usa para generar el siguiente conjunto de ítemsets candidatos a ser frecuentes, siguiendo un resultado presentado en [Agrawal y Srikant, 1995] según el cual todos los subconjuntos propios de un ítemset frecuente han de ser también frecuentes. De esta forma, se utiliza una especie de poda para eliminar de antemano aquellos candidatos que no van a ser frecuentes, ahorrándonos espacio de memoria y tiempo de proceso.

El conjunto de ítemsets resultante se usa como entrada para el algoritmo de generación de las reglas de asociación en sí. En las sucesivas extensiones y optimizaciones para la extracción de reglas de asociación o medidas similares, esta parte del algoritmo ha permanecido normalmente invariante. Podemos ver un esquema del mismo en el algoritmo B.3, que se halla en el apéndice B, página 238.

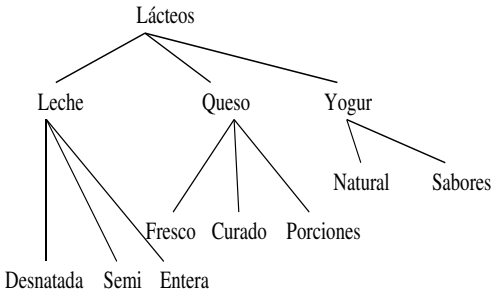
2.3.5. Otros tipos de reglas de asociación

A continuación comentaremos brevemente algunas extensiones al problema general de extracción de reglas de asociación, como son las reglas de asociación generalizadas y las cuantitativas.

2.3.5.1. Reglas de asociación generalizadas

En la mayoría de los casos, es posible construir una taxonomía (conjunto de jerarquías del tipo *es-un*) sobre los ítems. Por ejemplo, en unos grandes almacenes pueden tener 50 marcas distintas de queso (por nombrar un producto en concreto). En la base de datos transaccional se almacenará, si se compra una marca concreta, esa marca de queso como ítem. Es posible que esa marca (o alguna de las otras) por separado no tenga soporte mínimo en la base de datos, pero que el número global de compras de queso sí lo tenga. Si no somos capaces de definir una jerarquía para darnos cuenta de que el número de compras de queso puede tener importancia, estaremos perdiendo una información que puede resultar útil. Por lo tanto, sería interesante poder extraer reglas cuyos ítems se encuentren en cualquier nivel de la jerarquía, y serían más generales cuanto más cerca tuvieran algún ítem de la raíz.

Figura 2.4: Ejemplo de taxonomía definida sobre un conjunto de productos lácteos



Hay, por tanto, dos motivos para encontrar reglas en niveles distintos de una taxonomía:

- Las reglas de niveles más bajos pueden no tener soporte mínimo, como ya hemos visto en el ejemplo anterior.

- Se pueden utilizar las taxonomías para podar reglas redundantes o sin interés. Para ello habremos de definir una medida de interés.

Distintos autores han abordado este problema, proponiendo diversas soluciones al mismo. En este punto, destacaremos los diversos algoritmos presentados en [Srikant y Agrawal, 1995], a través de los cuales se define la taxonomía como un grafo dirigido acíclico, antes de aplicar un algoritmo de extracción de reglas de asociación modificado convenientemente.

2.3.5.2. Reglas de asociación cuantitativas

Lo dicho hasta ahora sobre reglas de asociación es en realidad un caso particular, que se suele referir en la literatura como el problema de las Reglas de Asociación Binarias (BARP). Cuando se representa un conjunto de transacciones mediante una base de datos relacional, tal y como veíamos en el apartado 2.2.1, se interpreta cada ítem con un atributo cuyo dominio es el conjunto $\{0, 1\}$, indicando presencia/ausencia de los ítems respectivos. Sin embargo, en la mayoría de las bases de datos existentes, los atributos tienen tipos de datos más ricos en información. Pueden ser cuantitativos (por ejemplo: edad, sueldo, ...) o bien categóricos (como pueden ser color, artículo, sexo, ...). Resulta muy sencillo extender este caso para la extracción de reglas de asociación en bases de datos relacionales, simplemente asociando cada ítem a un par $< atributo, valor >$, lo que también puede notarse como $[Atributo = valor]$.

Si todos los atributos de la base de datos son categóricos o los atributos cuantitativos tienen pocos valores (sus correspondientes dominios son finitos y reducidos), la transformación es directa y poco problemática. Pero si el dominio de los valores es grande, estamos ante un problema similar al caso de las Reglas de Asociación Generalizadas. Una posible solución sería dividir el dominio en intervalos y luego transformar cada par $[Atributo \in intervalo]$ en un atributo booleano o ítem. Una vez hecho esto, podemos aplicar cualquier algoritmo como los anteriormente comentados para la extracción de reglas de asociación booleanas, pero para extraer, en este caso, reglas de asociación cuantitativas.

Existen dos problemas cuando aplicamos esto a atributos cuantitativos:

- *"Minsupp"*. Si el número de intervalos para un atributo cuantitativo (o de valores, si optamos por no dividir el dominio del atributo) es grande, el soporte para cada intervalo en solitario puede ser bajo. Por lo tanto, si no se utilizan intervalos mayores, es posible que no se encuentren reglas que involucren este atributo por causa de no alcanzar el soporte mínimo necesario.

- *"Minconf"*. Existe cierta pérdida de información cuando dividimos los dominios en intervalos. Algunas reglas pueden tener confianza mínima (en nuestro caso, factor de certeza mínimo) sólo cuando un ítem del antecedente consiste en un sólo valor o en un intervalo pequeño. Esta pérdida de información se incrementa conforme crece el tamaño de los intervalos.

Para solucionar ese "tira y afloja" entre los dos problemas anteriores podemos considerar todos los posibles rangos continuos sobre los valores del atributo cuantitativo o sobre los intervalos. El problema *"Minsupp"* desaparece, ya que podemos combinar intervalos o valores adyacentes. El problema *"Minconf"* sigue presente, pero la pérdida de información puede reducirse incrementando el número de intervalos, sin encontrarnos con el problema *"Minsupp"*.

Pero incrementar en exceso el número de intervalos a la vez que se combinan intervalos adyacentes introduce dos nuevos problemas:

- *"ExecTime"*. Si un atributo cuantitativo tiene n valores (o intervalos), hay una media de $\mathcal{O}(n^2)$ rangos que contienen un valor o intervalo específico. Por tanto, el número de ítems por tupla se dispara, lo que también hará mucho mayor el tiempo de ejecución.

- *"ManyRules"*. Si un valor (o intervalo) de atributos cuantitativos tiene soporte mínimo, también lo tendrá cualquier rango en el que esté. Así, el número de reglas también se dispara. Muchas de estas reglas no serán interesantes y en consecuencia, constituirán un gasto innecesario de tiempo y memoria.

Para información más detallada sobre la definición de estas extensiones a las reglas de asociación así como a los algoritmos que nos permiten extraer-

las, remitimos al lector a los trabajos publicados en [Srikant y Agrawal, 1996, Zhang et al., 1997].

En lo que respecta a esta memoria, estas extensiones al problema de las reglas de asociación booleanas constituyen una primera aproximación al problema de los atributos numéricos que se ha resuelto con el concepto que más adelante definiremos de Reglas de Asociación Difusas.

2.3.6. Extracción de dependencias aproximadas

De manera informal, una **Dependencia Aproximada** puede entenderse como una dependencia *"casi"* funcional, en el sentido que estamos comentando de admitir la posibilidad de excepciones. De acuerdo con esto, las dependencias aproximadas están afectadas por un cierto grado de incertidumbre, como ocurre con las reglas de asociación, aunque claramente se diferencian de éstas en que una dependencia aproximada relaciona atributos entre sí, mientras que una regla de asociación relaciona valores de atributos.

Para aclarar este concepto, debemos volver atrás y referirnos al apartado 2.1.3 para recordar lo que entendíamos por dependencia funcional en una base de datos relacional. Como se apuntaba hacia el final de dicho apartado, una dependencia funcional puede ser muy difícil de hallar, por lo restrictivo de su definición. Una alternativa es entonces la de permitir la existencia de excepciones a la regla, en forma de tuplas que no cumplan la dependencia. Podemos medir entonces el grado de cumplimiento de dicha dependencia en términos de probabilidad, para lo cual se ha de manejar incertidumbre. En relación a dicha materia, algunos autores como [Piatetsky-Shapiro et al, 1993], [Kivinen y Mannila, 1995], [Pfahringer y Kramer, 1995], [Huhtala et al., 1998], [Cubero et al., 1998a] o [Delgado et al, 1999b], aportan interesantes discusiones en sus respectivos trabajos.

Como se propone en [Bosc et al., 1997, Cubero et al., 1998a], una de las formas para permitir la existencia de excepciones a la dependencia es la relajación del cuantificador universal de la definición (ecuación 2.1). La medida de cumplimiento se obtiene contando las excepciones y calculando el porcentaje de tuplas que cumplen la dependencia. Esto en la práctica se corresponde a las dependencias aproximadas, que también pueden encontrarse en la literatura

como determinaciones parciales o dependencias funcionales parciales.

En el caso que nos ocupa nos vamos a centrar en el modelo comentado en [Blanco et al., 2000, Delgado et al., 2000a]. Este enfoque nos permite calcular dependencias aproximadas a partir de reglas de asociación, mediante una sencilla transformación de la tabla correspondiente. De esta forma, además, podemos medir el grado de cumplimiento y el soporte de una dependencia aproximada mediante el factor de certeza y el soporte de una regla de asociación, respectivamente.

Para ello, partimos de la idea de que, si la dependencia funcional "$V \rightarrow W$" puede entenderse como una regla que relaciona tuplas por pares en cuanto a la igualdad de sus atributos (según vimos en el apartado 2.1.3, en la definición de dependencia funcional), y las reglas de asociación nos informan de la presencia de ítems en transacciones, podemos representar una dependencia aproximada como una regla de asociación, por medio de unas determinadas interpretaciones de los conceptos de ítem y transacción.

Definición 2.3.6 ([Blanco et al., 2000]) *Sean $RE = \{At_1, At_2, \ldots, At_m\}$ un esquema relacional, r una instancia de RE y $V, W \subset RE$ dos conjuntos disjuntos de atributos. Definimos los siguientes elementos:*

- *Un ítem es un objeto asociado a un atributo de nuestro esquema relacional RE. Para cada atributo $At_k \in RE$ denotaremos como it_{At_k} al ítem asociado.*
- *Definimos el ítemset I_V como*

$$I_V = \{it_{At_k} | At_k \in V\} \tag{2.9}$$

- *T_r es un conjunto de transacciones que, para cada par de tuplas $< t, s > \in r \times r$ (r es una instancia de RE), contiene una transacción $ts \in T_r$ que verifica*

$$it_{At_k} \in ts \Leftrightarrow t[At_k] = s[At_k] \tag{2.10}$$

Como consecuencia, se obtiene la igualdad $|T_r| = |r \times r| = n^2$.

De acuerdo con esta definición, una dependencia aproximada $V \rightarrow W$ en la relación r se corresponde con una regla de asociación $I_V \Rightarrow I_W$ en T_r

[Blanco et al., 2000, Delgado et al., 2000a]. El soporte y el factor de certeza de $I_V \Rightarrow I_W$ miden, respectivamente, el interés y la precisión de la dependencia $V \rightarrow W$. Y, en particular, se cumple la siguiente propiedad:

Proposición 2.3.1 ([Blanco et al., 2000])
Si $CF(V \rightarrow W) = 1$ *entonces* $V \rightarrow W$ *es una dependencia funcional.*

Es decir, $V \rightarrow W$ es una dependencia funcional cuando, en el conjunto de transacciones asociado todas las transacciones que satisfacen I_V satisfacen también I_W, o lo que es lo mismo, $CF(I_V \rightarrow I_W) = 1$.

2.3.6.1. Modelo de una dependencia aproximada

Entenderemos por modelo (aunque también puede encontrarse como *"teoría"* [Pfahringer y Kramer, 1995, Blanco et al., 2000]) de una dependencia aproximada, el conjunto de reglas de asociación que describen las relaciones entre los valores del antecedente y el consecuente de la dependencia.

Se suele asociar la calidad de una dependencia aproximada con la calidad del modelo que la describe. Las reglas de asociación que conforman el modelo de una dependencia aproximada se definen sobre r. Nos informan sobre la presencia de valores de atributos en las tuplas. En este apartado proporcionaremos una interpretación del soporte, confianza y factor de certeza de las reglas de asociación que integran el modelo de una dependencia aproximada. El hecho de que consideremos la confianza se debe a que el factor de certeza se calcula a partir de ésta, aunque en el fondo sólo consideremos el soporte y el factor de certeza como medidas de la bondad de una dependencia aproximada. En primer lugar, formalizaremos el concepto de modelo de una dependencia aproximada.

Definición 2.3.7 ([Blanco et al., 2000]) *El modelo de la dependencia aproximada* $V \rightarrow W$*, que notaremos por* $Th_{[V \rightarrow W]}$*, se define como el siguiente conjunto de reglas de asociación,*

$$\{Ru_{ij} \mid \exists t \in r \text{ with } t[V] = v_i \wedge t[W] = w_j\}$$

donde Ru_{ij} *es la regla de asociación* $[V = v_i] \Rightarrow [W = w_j]$*. Denotaremos a* s_{ij}*,* c_{ij} *y* cf_{ij} *como el soporte, confianza y factor de certeza de* Ru_{ij}*, respectivamente.*

A través de las siguientes propiedades se asocian el soporte y la precisión de una dependencia con el soporte y precisión de las reglas que conforman su modelo.

Proposición 2.3.2 ([Blanco et al., 2000])

$$Supp(V \rightarrow W) = \sum_{Ru_{ij} \in Th_{[V \rightarrow W]}} s_{ij}^2$$

Proposición 2.3.3 ([Blanco et al., 2000])

$$\frac{1}{Conf(V \rightarrow W)} =$$

$$= \sum_{Ru_{ij} \in Th_{[V \rightarrow W]}} \left(\left(\frac{s_{ij}^2}{Supp(V \rightarrow W)} \right) \frac{1}{c_{ij}} \right)$$

De acuerdo con estas propiedades, podemos interpretar el soporte y la confianza de una dependencia aproximada como una agregación o resumen del soporte y la confianza de las reglas de su modelo. Y lo mismo podemos decir del soporte de cualesquiera ítemsets I_V e I_W con respecto al soporte de los valores de V y W, respectivamente.

Debido al hecho de que el factor de certeza se define a partir del soporte y la confianza, el factor de certeza de una dependencia también puede verse como una agregación del soporte y la bondad de las reglas de su modelo. En concreto, se cumple la siguiente propiedad:

Proposición 2.3.4 ([Blanco et al., 2000]) *Si* $c_{ij} = c_0$ *y* $cf_{ij} = cf_0$ $\forall Ru_{ij} \in Th_{[V \rightarrow W]}$, *entonces* $Conf(V \rightarrow W) = c_0$ *y* $CF(V \rightarrow W) = cf_0$.

Lo que nos dice esta propiedad es que cuando todas las reglas de $Th_{[V \rightarrow W]}$ tienen la misma confianza y el mismo factor de certeza, éstos son la confianza y el factor de certeza de la dependencia asociada.

Tabla 2.8: Una relación, r

	A	B	C
t_1	a_1	b_1	c_1
t_2	a_2	b_1	c_2
t_3	a_1	b_1	c_3
t_4	a_3	b_2	c_3

Tabla 2.9: **(A)** El T-set T_r obtenido a partir de r **(B)** Dep. Aproximadas en r (reglas de asociación en T_r)

	it_A	it_B	it_C
t_1t_1	1	1	1
t_1t_2	0	1	0
t_1t_3	1	1	0
t_1t_4	0	0	0
t_2t_1	0	1	0
t_2t_2	1	1	1
t_2t_3	0	1	0
t_2t_4	0	0	0
t_3t_1	1	1	0
t_3t_2	0	1	0
t_3t_3	1	1	1
t_3t_4	0	0	1
t_4t_1	0	0	0
t_4t_2	0	0	0
t_4t_3	0	0	1
t_4t_4	1	1	1

A

$[B] \rightarrow [A], supp$ 37,5 %, CF 0,36
$[A] \rightarrow [B], supp$ 37,5 %, CF 1,0
$[C] \rightarrow [A], supp$ 25,0 %, CF 0,55
$[A] \rightarrow [C], supp$ 25,0 %, CF 0,55
$[C] \rightarrow [B], supp$ 25,0 %, CF 0,55
$[B] \rightarrow [C], supp$ 25,0 %, CF 0,20
$[B, C] \rightarrow [A], supp$ 25,0 %, CF 1,0
$[C] \rightarrow [A, B], supp$ 25,0 %, CF 0,55
$[B] \rightarrow [A, C], supp$ 25,0 %, CF 0,20
$[A, C] \rightarrow [B], supp$ 25,0 %, CF 1,0
$[A] \rightarrow [B, C], supp$ 25,0 %, CF 0,55
$[A, B] \rightarrow [C], supp$ 25,0 %, CF 0,55

B

2.3.6.2. Ejemplo de extracción de dependencias aproximadas

Veamos un ejemplo de lo anteriormente expuesto. Consideremos una relación r como la mostrada en la tabla 2.8. Aplicando la transformación correspondiente, obtendríamos el conjunto de transacciones T_r, que aparece en la tabla 2.9(A).

Una vez hecho esto y aplicando un algoritmo simple de extracción de reglas de asociación sobre T_r, obtendríamos el conjunto de reglas que aparecen en la

tabla 2.9(B). De acuerdo con nuestra definición, esas reglas de asociación en T_r se corresponderían con dependencias aproximadas en r. Y, en particular, como nos dice la proposición 2.3.1, habríamos obtenido las dependencias funcionales $A \rightarrow B$ y $B, C \rightarrow A$ (ya que $A, C \rightarrow B$ es una ampliación de $A \rightarrow B$).

2.3.6.3. Algoritmos

En [Blanco et al., 2000], podemos encontrar el algoritmo B.5 (página 240), una extensión del proceso clásico de extracción de ítemsets frecuentes, concretamente el algoritmo Apriori (algoritmo B.2), descrito con anterioridad en el apartado 2.3.4.5.

Este algoritmo parte del concepto de modelo de una dependencia aproximada (ver [Pfahringer y Kramer, 1995, Blanco et al., 2000] para más detalles), para obtener ésta a partir de las reglas de asociación que conforman su modelo. Para ello, han de ser introducidas algunas variaciones sobre el algoritmo, como la necesidad de usar el conjunto de variables $N(V, v)$ para almacenar el soporte de las apariciones $< V, v >$, con $V \subseteq RE$. La principal ventaja aportada por este algoritmo es que es posible obtener el soporte de cada v y el soporte de I_V de forma simultánea, de acuerdo con el siguiente resultado y con el algoritmo B.4 (del cual encontramos su desarrollo en la página 239):

Proposición 2.3.5 ([Blanco et al., 2000]) *El soporte de un ítemset I_V puede calcularse como*

$$S(I_V) = \frac{1}{n^2} \sum_{i=1}^{K} \sum_{p=1}^{N(V,v_i)} (2p - 1) \tag{2.11}$$

Más adelante, en el capítulo 4 veremos cómo es posible adaptar estos algoritmos para la obtención de dependencias aproximadas difusas, nuestro principal objetivo presentado en esta memoria.

2.4. Resumen

A lo largo de este capítulo, hemos recordado conceptos tan relevantes hoy día tales como el modelo relacional de bases de datos, tan extendido actualmente, así como las principales herramientas de minería de datos y extracción

de conocimiento susceptibles de ser aplicadas sobre dicho modelo. En el siguiente capítulo, también a modo introductorio, veremos cómo estos mismos conceptos pueden ser extendidos para manejar información que pueda venir afectada por un cierto grado de imprecisión, incertidumbre o vaguedad en general. Nos centraremos en algunos resultados concretos, por la importancia que éstos tendrán después en la exposición de los resultados propios de esta memoria.

3. Preliminares (II): Relajando conceptos

Si el anterior capítulo iba destinado a introducir algunos conceptos generales, éste que aquí comienza tiene como función la de ver cómo es posible extender estos conceptos anteriores, enriqueciéndolos para permitirnos manejar información más cercana al lenguaje natural, y que pueda estar afectada por ese motivo de algún grado de imprecisión, incertidumbre o vaguedad en general.

Comenzaremos resumiendo algunos conceptos clave de la Teoría de Subconjuntos Difusos, para ver a continuación cómo éstos pueden ser aplicados para la extensión del Modelo Relacional de Bases de Datos. En este punto comentaremos de manera algo más detallada los resultados obtenidos por nuestro grupo de investigación, ya que su implementación constituye la base del módulo FuzzyQueries 2+, que presentaremos más adelante como parte de esta tesis doctoral.

Seguidamente, recordaremos cómo es posible extender las técnicas de Minería de Datos y Extracción de Conocimiento mediante la lógica difusa para

analizar convenientemente información de carácter impreciso y cómo dichas extensiones resultan especialmente útiles en determinados casos reales.

3.1. La teoría de subconjuntos difusos

Durante mucho tiempo, las Ciencias de la Computación han estado estrechamente asociadas a las matemáticas y, en general, a las ciencias exactas. Como su propio nombre indica, el conocimiento con el que se trabajaba era muy concreto y estaba bien definido, y las técnicas y herramientas necesarias para su procesamiento se diseñaron y construyeron de acuerdo a las necesidades existentes. Conforme dichas necesidades han ido variando, igualmente ha sido necesario extender los modelos teóricos y prácticos existentes. Un ejemplo claro lo tenemos en el creciente interés sobre el procesamiento mediante un ordenador del lenguaje natural y, por ende, del razonamiento humano. Por naturaleza, ambos están afectados por cierto grado de vaguedad, de imprecisión, asumibles por el cerebro humano, pero no directamente por un ordenador.

En esta sección citamos uno de los modelos más extendidos para solucionar dichos problemas, introduciendo los conceptos sobre la teoría de subconjuntos difusos de Zadeh más relacionados con el contenido de esta memoria. Nos detendremos especialmente en algunos aspectos semánticos y de representación. Se puede ampliar el tema acudiendo al trabajo original [Zadeh, 1965]. Además, en [Yager et al, 1987] se recogen algunos de los artículos más interesantes publicados sobre el tema por L. A. Zadeh. Por último, [Dubois y Prade, 1979, Dubois y Prade, 1988, Zimmermann, 1991] recopilan los aspectos más importantes sobre la teoría de los subconjuntos difusos y la teoría de la probabilidad.

3.1.1. Conjuntos difusos

Entendemos por imprecisión una falta de especificidad en los datos, que puede derivarse de la imposibilidad de especificar una única posible asignación a una variable que represente cierta propiedad. Por ejemplo, en un caso aplicado a la agricultura, podemos tener cierto conocimiento sobre la profundidad de un suelo, sin poder especificar un valor exacto. Podríamos, usando una expresión muy común en lenguaje natural, decir que determinado suelo es *"profundo"*, en lugar de afirmar que su profundidad es exactamente de 150 cm. En

tal caso, el concepto de profundo se puede representar mediante un conjunto difuso (figura 3.1). Cuando hablamos de incertidumbre, queremos decir que no podemos afirmar rotundamente que la información con la que trabajamos sea totalmente cierta, por ejemplo, otorgando un grado de credibilidad o certeza a la afirmación anterior.

Pasaremos a definir en primer lugar el concepto de conjunto difuso, introducido por Zadeh en [Zadeh, 1965], así como los operadores habituales. Seguidamente, introduciremos al lector en los conceptos de relaciones y números difusos. Nuestra intención se reduce, exclusivamente, a la de fijar la notación y la terminología que seguiremos a lo largo de esta memoria. Para un estudio más detallado, podemos añadir a las referencias ya citadas algunos trabajos más recientes como los aparecidos en [Klir et al., 1995, Zadeh, 1996, Klir et al., 1997, Pedrycz, 1998b].

Definición 3.1.1 *Un conjunto difuso F es un conjunto de pares*

$$F = \{\mu_F(x)/x\}_{x \in X}$$

donde μ_F es la función de pertenencia de F, con rango e imagen dados por:

$$\mu_F : X \longrightarrow [0, 1]$$

siendo X un conjunto clásico.

Cabe destacar que, como caso particular, un conjunto clásico o "crisp" es un conjunto difuso. Es posible representar un conjunto difuso mediante su función de pertenencia, como se muestra en la figura 3.1.

Pasamos ahora a recordar brevemente otros conceptos relacionados con los conjuntos difusos.

- **Universo de discurso**. Es el conjunto clásico X sobre el que está definida la función de pertenencia. También se puede encontrar en la literatura como referencial.

- Un valor **crisp** es cualquier elemento del universo de discurso.

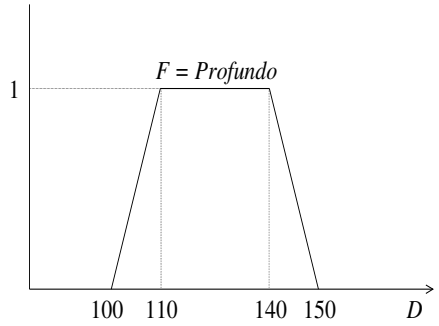


Figura 3.1: Ejemplo de un conjunto difuso para representar el concepto de "profundo"

- Cuando X es finito (numerable en general, $X = \{x_1, x_2, \ldots, x_n\}$), podemos utilizar la siguiente notación:

$$F = \mu_F(x_1)/x_1 + \mu_F(x_2)/x_2 + \ldots + \mu_F(x_n)/x_n = \sum_i \mu_F(x_i)/x_i \quad (3.1)$$

- Denotaremos por $\tilde{\mathcal{P}}(X)$ al conjunto (clásico) de todos los posibles conjuntos difusos definidos sobre un universo de discurso X, de lo que sabemos que tendrá sentido decir que $F \in \tilde{\mathcal{P}}(X)$. Y como un conjunto clásico es un caso particular de un conjunto difuso, también se verifica que

$$\mathcal{P}(X) \subset \tilde{\mathcal{P}}(X) \quad (3.2)$$

- El **soporte** de un conjunto difuso, $sop(F)$, es el conjunto de valores del universo con grado de pertenencia estrictamente positivo:

$$sop(F) = \{x \in X | \mu_F(x) > 0\} \quad (3.3)$$

- El **núcleo** de un conjunto difuso, $ker(F)$, es el conjunto de valores del universo con grado de pertenencia igual a 1:

$$ker(F) = \{x \in X | \mu_F(x) = 1\} \quad (3.4)$$

Por lo anteriormente visto, se verifica que $sop(F) \supseteq ker(F)$.

- Se dice que un conjunto difuso está **normalizado** si y sólo si

$$ker(F) \neq \emptyset \Leftrightarrow \exists x \in X | \mu_F(x) = 1 \tag{3.5}$$

- Definimos el **alfa corte** (o α-corte) de un conjunto difuso como sigue:

$$F_\alpha = \{x \in X | \mu_F(x) \geq \alpha\} \tag{3.6}$$

El α-corte de un conjunto difuso es, a su vez, otro conjunto, pero en este caso "crisp", como nos indica su definición.

- Dados dos conjuntos difusos F y G, definimos la intersección $(F \cap G)$ y la unión $(F \cup G)$, a partir de sus funciones de pertenencia:

$$\mu_{F \cap G}(x) = min\{\mu_F(x), \mu_G(x)\}$$
$$\mu_{F \cup G}(x) = max\{\mu_F(x), \mu_G(x)\}$$

Se utilizará indistintamente el símbolo $\vee$ ($\wedge$) tanto para denotar el operador máximo (mínimo), como para referirnos a la unión (intersección) difusa, por lo que las anteriores expresiones también podemos escribirlas como sigue:

$$\mu_{F \wedge G}(x) = \mu_F(x) \wedge \mu_G(x)$$
$$\mu_{F \vee G}(x) = \mu_F(x) \vee \mu_G(x)$$

3.1.2. Números difusos

Cuando el universo de discurso es un subconjunto de la recta real $\Re$, un conjunto difuso puede representar una valoración o una cantidad imprecisa. Para que esto realmente ocurra, y para facilitar la operabilidad, se suelen añadir ciertas restricciones adicionales, obteniendo entonces la definición de número difuso:

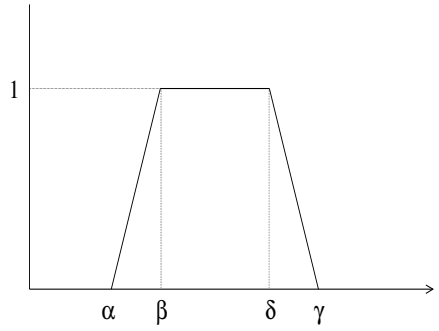


Figura 3.2: Número difuso definido por cuatro parámetros

Definición 3.1.2 *Un número difuso F es un subconjunto difuso de $\Re$ ($F \in \tilde{\mathcal{P}}(\Re)$) que verifica las siguientes propiedades:*

(I) *La función de pertenencia es convexa, es decir:*

$$\forall x, y \in \Re, \forall z \in [x, y], \mu_F(z) \geq min\{\mu_F(x), \mu_F(y)\} \tag{3.7}$$

(II) *μ_F es continua a la izquierda en todos los puntos. Por lo tanto, los α-cortes de F son intervalos cerrados de $\Re$.*

(III) *El soporte de F está acotado:*

$$\exists I, S \in \Re, tales\ que\ sop(F) \subseteq [I, S] \tag{3.8}$$

Como consecuencia de esta propiedad y de la anterior, se deduce que todos los α-cortes de F son intervalos cerrados y acotados de $\Re$, es decir, compactos.

(IV) *F está normalizado:*

$$sup_x \mu_F(x) = 1 \tag{3.9}$$

Una de las formas más cómodas mediante las que representar un números difusos es a través de una función trapezoidal. En este caso, sólo necesitamos 4 parámetros para delimitar el conjunto difuso. Éstos corresponden a los valores inferior y superior del núcleo y soporte, respectivamente, como se puede ver en la figura 3.2.

Otro concepto relacionado que merece destacarse es el de **distribución de posibilidad**, derivado de la Teoría de la posibilidad [Dubois y Prade, 1988, Zadeh, 1978]. Formalmente, una distribución de posibilidad es un conjunto difuso normalizado (es decir, en el que al menos uno de sus elementos tiene grado de pertenencia igual a 1, o dicho de otra forma, su núcleo es no vacío). En particular, todos los números difusos son distribuciones de posibilidad. Uno de los usos más comunes de este tipo de distribuciones es el de asociarlas con un término lingüístico, como veremos a continuación.

3.1.3. Etiquetas lingüísticas

En ocasiones resulta necesario describir una propiedad o el estado de un objeto o fenómeno mediante una expresión en lenguaje natural. Para ello se ha de usar lo que se conoce como una variable lingüística, la cual admite el que sus valores sean etiquetas lingüísticas.

Una **etiqueta lingüística** es un término extraído del lenguaje natural mediante el que expresamos una valoración imprecisa que, en ciertos casos, puede tener una representación mediante un conjunto difuso. Por ejemplo, *"profundo"* es una etiqueta lingüística, mientras que el intervalo [15,25] (que puede interpretarse como un conjunto difuso) no lo sería. Este concepto fue desarrollado por Zadeh en [Zadeh, 1975a, Zadeh, 1975b, Zadeh, 1976] y es muy utilizado cuando el universo de discurso es algún subconjunto de la recta real $\Re$.

Una de las posibles aplicaciones del concepto de etiqueta lingüística y de la que haremos uso más adelante es la definición de conjuntos de etiquetas sobre dominios. La figura 3.3 nos muestra un ejemplo, en el que se ha definido un conjunto de etiquetas {*Muy bajo, Bajo, Medio, Alto, Muy alto*} sobre un dominio numérico que en el presente caso podría corresponder a una característica tal como la Altura de un individuo, expresada en metros.

A la representación de una etiqueta lingüística como conjunto difuso se le denomina **Representación semántica**.

Existen conjuntos de variables cuya definición es más compleja porque se mueven en dominios subyacentes poco claros, que no resultan fáciles de trasladar a valores numéricos: Sabor de una comida, Color del pelo, etc.

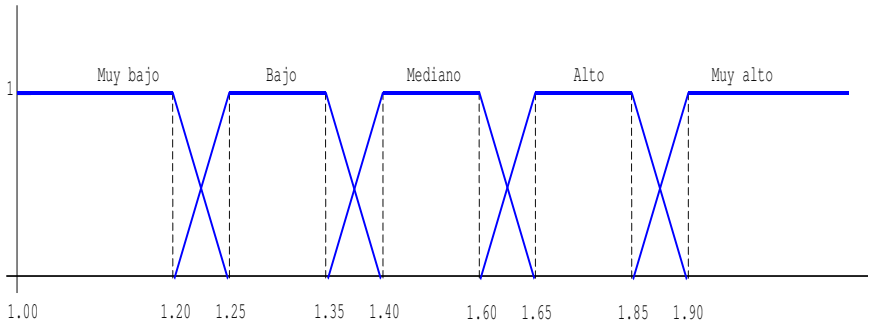


Figura 3.3: Etiquetas lingüísticas definidas sobre un dominio numérico

Los conjuntos de etiquetas lingüísticas definen jerarquías de valores sobre las variables que representan. Un problema importante asociado a estos conceptos es el de la **Granularidad**. Esta propiedad está relacionada con el número de etiquetas consideradas en el conjunto:

- Una granularidad alta (*coarse*) se corresponde con un número bajo de etiquetas. El dominio está poco particionado y tiene el inconveniente de que se pueda perder algo de expresividad.
- Por el contrario, una granularidad baja ($fine$) se asocia con un número alto de etiquetas. Una granularidad demasiado baja puede provocar un aumento de la complejidad en la descripción del dominio.

El análisis de la granularidad en la representación de la información mediante etiquetas es un problema muy común en áreas de conocimiento tales como las bases de datos y los sistemas de ayuda a la decisión.

Siempre existe una relación de semejanza definida sobre todo conjunto de etiquetas, de manera que se pueda conocer el grado de parecido entre cualesquiera dos etiquetas de dicho conjunto.

En la literatura se contemplan diversos tipos de conjuntos de etiquetas destacados:

- **Valoraciones de certeza**. Se definen sobre el dominio o referencial asociado al intervalo $[0, 1]$. Se descomponen a su vez en dos subtipos:

  - **Probabilisticas**, en las que el conjunto de etiquetas suele ser de la forma {*Casi seguro, Muy probable, Probable, Poco probable, Improbable*}. Mediante estas etiquetas se expresa la probabilidad de que el objeto cumpla una cierta propiedad.
  - **Posibilísticas**, donde el conjunto suele ser {*Casi seguro, Muy posible, Posible, Poco posible, Casi imposible*}. En este caso, se expresa la posibilidad de que se cumpla la propiedad dada.

- **Valoraciones asociadas al control difuso**. En esta ocasión, el referencial sobre el que se define el conjunto se corresponde o es proporcional al intervalo $[-1, 1]$, tomando una estructura como {*Positivo alto, Positivo, Nulo, Negativo, Negativo bajo*}.

- Por último, los **Cuantificadores** toman un conjunto numérico como referencial. Los describiremos con más detalle a continuación.

3.1.3.1. Cuantificadores difusos

Un cuantificador difuso [Zadeh, 1979b] expresa una cantidad difusa (cuantificador absoluto, por ejemplo, "alrededor de 5", "no menos de 7") o un porcentaje difuso (cuantificador relativo, p.e., "la mayoría", "unos pocos") por medio de términos lingüísticos. Un cuantificador absoluto se define como un conjunto difuso sobre el dominio de los enteros no negativos ($\mathbb{Z}^+$), mientras que un cuantificador relativo es un conjunto difuso sobre el intervalo $[0, 1]$.

Vamos a restringir nuestro campo de visión hacia los cuantificadores difusos coherentes, que, como veremos más adelante, nos resultarán más comodos para la evaluación de sentencias cuantificadas y, en particular, el cálculo de las medidas asociadas a las reglas de asociación (y también dependencias aproximadas) difusas, de acuerdo a lo expuesto en [Delgado et al., 2003a].

Definición 3.1.3 ([Cubero et al., 1995]) *Decimos que Q es un cuantificador difuso coherente, si su función de pertenencia $Q(\cdot)$ verifica las siguientes propiedades:*

- $Q(0) = 0$ *y* $Q(1) = 1$
- *Monotonía: Si* $x < y$, $Q(x) \leq Q(y)$.

3.1.4. Relaciones difusas

Dados dos elementos, no necesariamente incluidos en el mismo conjunto, es posible establecer una conexión entre ellos. Si esa conexión tiene un carácter impreciso o gradual, podemos formularla como una relación difusa.

Definición 3.1.4 *R es una **relación difusa** sobre $X_1, \ldots, X_n$ si:*

$$R \in \tilde{\mathcal{P}}(X_1 \times \ldots \times X_n)$$

Dicho de otra forma, una relación difusa es un conjunto difuso definido sobre un producto cartesiano de universos de discurso. Puede observarse que:

$$\tilde{\mathcal{P}}(X_1 \times \ldots \times X_n) \neq \tilde{\mathcal{P}}(X_1) \times \ldots \times \tilde{\mathcal{P}}(X_n)$$

Nos interesará trabajar con casos particulares de relaciones difusas como los siguientes:

Definición 3.1.5 *[[Zadeh, 1971, Rundensteiner et al, 1989]] Una relación difusa R, definida sobre $X \times X$, es una relación de semejanza si y sólo si verifica las siguientes propiedades:*

- *Reflexiva:* $R(x, x) = 1, \forall x \in X$
- *Simétrica:* $R(x, y) = R(y, x), \forall x, y \in X$

Definición 3.1.6 *[[Zadeh, 1971]] Se dice que una relación difusa sim definida sobre $X \times X$ es una relación de similitud si y sólo si es una relación de semejanza y verifica además la propiedad Max-min transitiva, esto es:*

$$sim(x, z) \geq max_{y \in X}\{min(sim(x, y), sim(y, z))\} \tag{3.10}$$

Las relaciones de similitud son una extensión del concepto clásico de relación de equivalencia. De hecho, los α-cortes de una relación difusa de similitud son relaciones crisp de equivalencia.

Un resultado interesante es el que demuestra Potoczny en [Potoczny, 1984], según el cual únicamente es necesario elicitar la similitud entre ciertos valores del conjunto, viniendo determinados el resto por la definición de max-min transitividad.

3.1.5. Cardinales difusos

Formalmente, entendemos por cardinal de un conjunto el número de elementos que pertenecen a dicho conjunto. Cuando extendemos el concepto de conjunto al caso difuso, de igual forma hemos de extender lo que entendemos por cardinal de un conjunto difuso. A continuación, introduciremos algunos conceptos relativos a diversos estudios realizados al respecto. Remitimos al lector a [Delgado et al., 2002] para un estudio detallado de los enfoques más interesantes aparecidos sobre este problema.

3.1.5.1. Algunos cardinales sobre conjuntos difusos

Dado F un conjunto difuso definido sobre el referencial $X = \{x_1, \ldots, x_n\}$ con una función de pertenencia $F : X \to [0, 1]$, la definición más sencilla del cardinal de F viene dada por DeLuca y Termini en 1972:

Definición 3.1.7 ([DeLuca y Termini, 1972]) *El **cardinal crisp** de F viene dado por*

$$P(F) = \sum_{i=1}^{n} F(x_i) \tag{3.11}$$

El principal inconveniente del cardinal crisp estriba en que cuando n es un valor alto, también se obtendrá un valor alto para el cardinal, incluso cuando los grados de pertenencia sean bajos. Además, en el caso que nos ocupa, dar como respuesta ese valor del cardinal podría resultar contraintuitivo, al devolver un número real. Un enfoque más trabajado es el de calcular el cardinal como un intervalo de valores reales, como hicieron Dubois y Prade en 1985.

Definición 3.1.8 ([Dubois y Prade, 1985]) *Con $|F_\alpha|$ el cardinal de cada α-corte de F, se define el enfoque intervalar de acuerdo a la definición de Dubois y Prade como el siguiente intervalo:*

$$DPs(F) = [|F_1|, P(F)] \tag{3.12}$$

En otros estudios, se ha considerado el propio cardinal como un conjunto difuso sobre el referencial $H = \{1, 2, \ldots, n\}$, y en la mayoría de ellos se con-

sidera una nueva función $f : H \longrightarrow [0,1]$ asociada a F y definida de la forma siguiente:

$$\forall k \in H; f_k = sup\{\alpha / |F_\alpha| \geq k\} \tag{3.13}$$

con $f_k = 0$ para $k = n+1$. Por medio de esta función, podemos encontrar las siguientes definiciones de cardinales:

Definición 3.1.9 ([Zadeh, 1983]) *Calculamos el cardinal de F con respecto a la definición de Zadeh como:*

$$\forall k \in H; ZA_F(k) = \begin{cases} \text{mín}(f_k, 1 - f_{k-1}) & si\ k \geq |F_1| \\ 0 & en\ otro\ caso \end{cases} \tag{3.14}$$

Definición 3.1.10 ([Dubois y Prade, 1985]) *Dubois y Prade aportan la siguiente definición para el cardinal de F:*

$$\forall k \in H; DP_F(k) = \begin{cases} f_k & si\ k \geq |F_1| \\ 0 & en\ otro\ caso \end{cases} \tag{3.15}$$

Definición 3.1.11 ([Delgado et al., 2000b]) *Por último, también encontramos en las referencias la definición de Delgado, Sánchez y Vila para el cardinal de F:*

$$\forall k \in H; DSV(k) = f_k - f_{k+1} \tag{3.16}$$

considerando en este caso $H = \{0, 1, 2, \ldots n\}$.

Las dos primeras generan una distribución de posibilidad para el cardinal, mientras que la última usa un enfoque probabilístico.

3.1.6. Evaluación de sentencias cuantificadas

Otro concepto que necesitamos incluir en estos apartados preliminares es el de sentencia cuantificada. Una sentencia cuantificada es una afirmación expresada en lenguaje natural sobre el número o porcentaje de objetos que verifican una cierta propiedad. Pueden aplicarse en muchos campos, como en la

realización de consultas flexibles en bases de datos o dentro del ámbito de los sistemas expertos. En [Liu y Kerre, 1998] podemos encontrar un amplio resumen de algunas de estas aplicaciones.

De acuerdo con [Zadeh, 1983], este tipo de sentencias se clasifican en dos tipos, sentencias tipo I y sentencias tipo II. Una sentencia tipo I tiene la forma

"Q de X son A",

donde $X = \{x_1, \ldots, x_n\}$ es un conjunto finito, Q es un cuantificador lingüístico y A es una propiedad difusa definida sobre el conjunto X. Por ejemplo, *"la mayoría de los alumnos son jóvenes"*. X sería el conjunto finito de alumnos, la propiedad A sería *"ser joven"* y el cuantificador Q sería *"la mayoría"*. Por otro lado, las sentencias tipo II pueden describirse en general como

"Q de D son A",

donde D es también una propiedad difusa sobre X. Un ejemplo de sentencia tipo II podría ser *"la mayoría de los alumnos jóvenes son altos"*. Claramente, las sentencias tipo I son un caso particular de las sentencias tipo II, en las que $D = X$.

Normalmente, los cuantificadores absolutos se usan para evaluar sentencias tipo I, mientras que para el caso de las sentencias tipo II, sólo tiene sentido la aplicación de cuantificadores relativos. Recordamos ambos tipos de cuantificadores en el apartado 3.1.3.1.

Para más información sobre el uso y definición de cuantificadores y sobre la evaluación de otros tipos de sentencias, recomendamos algunas de las aportaciones más recientes al respecto como las que se pueden encontrar en [Glöckner, 2003a] (donde se describe un algoritmo para el cálculo de diversos cuantificadores) y [Glöckner, 2003]. Por último, en [Díaz-Hermida et al., 2003] encontramos una clasificación de los cuantificadores más interesantes. Concretamente, se restringe al caso de cuantificadores semi-difusos (aplicados sobre conjuntos "crisp" pero con un grado de cumplimiento como resultado), aunque proponen un algoritmo mediante el cual extender este tipo especial de cuantificadores al caso difuso.

La evaluación de sentencias cuantificadas trata de obtener un grado de cumplimiento de las mismas en el intervalo $[0, 1]$. Este grado puede obtenerse mediante distintos métodos, pero en general, se puede representar como *Eval("Q de X son A")* (respectivamente, *Eval("Q de D son A")*) para una sentencia tipo I (respectivamente, tipo II). En [Delgado et al., 2002], se propone un conjunto de propiedades intuitivas que cualquier método de evaluación debería verificar.

De acuerdo con dichas propiedades, en [Delgado et al., 2000c] puede verse cómo no todos los métodos de evaluación existentes son apropiados para según qué casos. La evaluación mediante cuantificadores no coherentes es muy significativa tanto para sentencias tipo I como para las del tipo II.

Conforme a nuestros propósitos, para la evaluación de sentencias cuantificadas vamos a utilizar el método GD, definido en [Delgado et al., 2000b] como,

$$GD_Q(G/F) = \sum_{\alpha_i \in \triangle(G/F)} (\alpha_i - \alpha_{i+1}) Q\left(\frac{|(G \cap F)_{\alpha_i}|}{|F_{\alpha_i}|}\right) \tag{3.17}$$

donde $\triangle(G/F) = \wedge(G \cap F) \cup \wedge(F)$, siendo $\wedge(F)$ el conjunto de niveles de F, y $\triangle(G/F) = \{\alpha_1, \ldots, \alpha_p\}$ con $\alpha_i < \alpha_{i+1}$ para cada $i \in \{1, ..., p\}$. Suponemos que el conjunto F está normalizado. De no ser así, F se normaliza y el factor de normalización se aplica a $G \cap F$.

El algoritmo B.6 (página 241) nos muestra el proceso a través del cual calcular de forma eficiente (ver [Delgado et al., 2000c]) la evaluación de una sentencia cuantificada *"Q de D son A"* mediante el método GD. Para una mejor comprensión del funcionamiento del mismo, hemos de considerar una constante k, equivalente al número de α-cortes mediante los que vamos a representar A y D. Consideramos que un valor apropiado para k deber ser al menos superior a 10 para que la representación de la cardinalidad de los conjuntos sea buena. En lo sucesivo, vamos a tomar $k = 100$, considerándolo un valor aceptable.

Más adelante, veremos una aplicación directa del método para obtener las medidas de certeza e interés asociadas a reglas de asociación (y dependencias aproximadas) difusas.

3.2. Bases de datos difusas

Un problema muy común hoy día que nos puede aparecer a la hora de representar conocimiento en una base de datos estriba en la posibilidad de que dicho conocimiento pueda venir afectado de imprecisión o incertidumbre en su definición. Se hace patente la necesidad de poner en práctica nuevos modelos, a menudo como extensiones de anteriores propuestas, sobre las que pueden apoyarse, para representar y manejar este tipo de conocimiento. La figura 3.4 nos muestra un esquema muy básico de una de las posibles soluciones al problema. Cuando la información de partida es incompleta, imprecisa o incierta, es necesario extender los modelos existentes de representación de la misma en bases de datos para manejar dichas características.

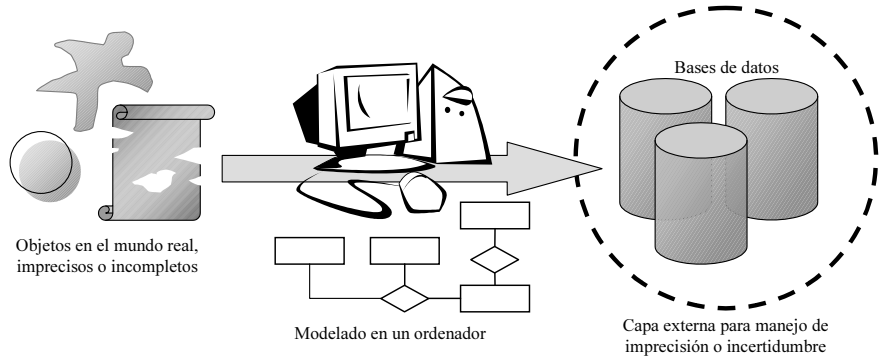


Figura 3.4: Representación de conocimiento impreciso en bases de datos

Los dos enfoques utilizados en el modelo relacional para la captura de información, como son el Álgebra Relacional y el Cálculo Relacional, parten en primera instancia de la premisa de que toda pregunta realizada a una BDR tendrá como respuesta una nueva relación, construida a partir de las ya existentes. Ninguno de ellos permitía, en su origen, la posibilidad de que las consultas se formularan de una manera imprecisa, esto es, un SBDR sería incapaz de responder a preguntas tales como:

"Hallar todos los individuos jóvenes con un salario no muy alto".

Como podemos ver, en la anterior consulta aparecen varios términos, de uso cotidiano en el lenguaje natural humano, pero a la vez cargados de impre-

cisión y por ello, imposibles de resolver por un SBDR clásico. A lo largo de los últimos años, diversos autores han abordado el problema de suavizar las restricciones al respecto presentes en el modelo relacional con objeto de admitir la imprecisión dentro del mismo. Merecen ser destacados el modelo de Buckles y Petry [Buckles y Petry, 1982, Buckles y Petry, 1984], que utiliza relaciones de similitud entre valores de atributos, el modelo propuesto por Umano y Fukami [Fukami et al, 1979], [Umano, 1982], [Umano y Fukami, 1994], que permite almacenar distribuciones de posibilidad, en el sentido expuesto por Zadeh en [Zadeh, 1978], o el modelo de Prade y Testemale, que podemos encontrar en [Prade y Testemale, 1984] y [Prade y Testemale, 1987], y el desarrollado por Zemankova y Kandel [Zemankova y Kandel, 1984]. Por otro lado, en [Bosc et al., 1994] y [Tahani, 1977] se describen algunos sistemas para realizar consultas flexibles sobre bases de datos clásicas.

El problema de relajar y hacer más flexible el modelo original de Codd puede tratarse desde dos niveles:

- Por un lado, añadir la posibilidad de manejar consultas imprecisas sobre bases de datos clásicas.
- Por otro, ampliar el modelo para gestionar y almacenar información imprecisa dentro del mismo, lo cual supone un paso más allá.

En ambos casos, la teoría de subconjuntos difusos de Zadeh [Zadeh, 1965], definida en el anterior apartado, proporciona una potente herramienta para representar cuestiones y datos imprecisos.

3.2.1. Modelo GEFRED

A continuación nos centraremos en un modelo concreto para representar Bases de Datos Relacionales Difusas, sobre el que basamos la aplicación Fuzzy-Queries 2+ que presentaremos más adelante.

Diversos modelos han hecho uso de la teoría de subconjuntos difusos de Zadeh [Zadeh, 1965] para ampliar el modelo relacional clásico, agrupándose generalmente en torno a dos tendencias: **Modelos de Unificación Mediante Relaciones de Similitud** y **Modelos Relacionales sobre Distribuciones de Posibilidad**. En la primera línea, podemos destacar trabajos como

los de Buckles y Petry [Buckles y Petry, 1982]. En el segundo enfoque tenemos, entre otros autores, a Prade y Testemale [Prade y Testemale, 1984]. Y, combinando lo mejor de ambos enfoques, encontramos el modelo **GEFRED** [Medina et al, 1994, Medina, 1994], el más general hasta la fecha, desde el momento en que maneja imprecisión e incertidumbre conjuntamente.

GEFRED se basa en las definiciones de Dominio Difuso Generalizado y Relación Difusa Generalizada, que extienden respectivamente a los dominios y relaciones clásicas.

3.2.1.1. Definiciones

Definición 3.2.1 *Sea U el Universo o Dominio de discurso, $\tilde{\mathcal{P}}(U)$ el conjunto de todas las distribuciones de posibilidad definidas sobre U, incluidas las que definen los tipos Unknown, Undefined y Null. Se define el* ***Dominio Difuso Generalizado*** *$D \subseteq \tilde{\mathcal{P}}(U) \cup Null$.*

Los tipos Unknown, Undefined y Null están definidos de acuerdo a lo expuesto en [Fukami et al, 1979, Umano, 1982].

- *Un valor es desconocido (Unknown) cuando no conocemos su valor. Tiene la siguiente distribución de posibilidad asociada,*

$$Unknown = \{1/a : a \in A\}$$

- *Un valor es indefinido (Undefined) si no puede aplicarse sobre el dominio considerado (p.e., fecha de nupcias para una persona soltera). Su distribución de posibilidad es la siguiente,*

$$Undefined = \{0/a : a \in A\}$$

- *Por último, un valor se considera nulo (Null) cuando tenemos un total desconocimiento sobre el mismo (es decir, ni sabemos qué valor toma ni si es o no aplicable). La distribución de posibilidad asociada es*

$$Null = \{1/Unknown, 1/Undefined\}$$

GEFRED utiliza relaciones de semejanza o cercanía sobre los dominios de los atributos, así como la posibilidad de establecer diversos grados de satisfacción de la consulta para cada uno de los atributos implicados en ella.

Definición 3.2.2 *Una **Relación Difusa Generalizada**, R, viene dada por un par de conjuntos, Cabecera ($\mathcal{H}$) y Cuerpo ($\mathcal{B}$), $R = (\mathcal{H}, \mathcal{B})$, definidos como sigue:*

- *Por Cabecera vamos a entender un conjunto finito de ternas de la forma $< atributo, dominio, compatibilidad >$ (siendo el último opcional),*

$$\mathcal{H} = \{(A_1 : D_1[, C_1]), (A_2 : D_2[, C_2]), \ldots, (A_m : D_m[, C_m])\}$$

  donde cada atributo A_j tiene asociado un dominio difuso generalizado subyacente, no necesariamente distinto, D_j ($j = 1, \ldots, m$). C_j es un atributo de compatibilidad *que toma valores en* $[0, 1]$.

- *El Cuerpo consiste en un conjunto de tuplas difusas generalizadas distintas, en el que cada tupla está compuesta por un conjunto de ternas $< atributo, valor, grado >$ (el grado es opcional),*

$$\mathcal{B} = \{(A_1 : \tilde{d_{i1}}[, c_{i1}]), (A_2 : \tilde{d_{i2}}[, c_{i2}]), \ldots, (A_m : \tilde{d_{im}}[, c_{im}])\}$$

  donde $i = 1, 2, \ldots, n$, siendo n el cardinal o número de tuplas de la relación, y donde $\tilde{d_{ij}}$ representa el valor de dominio que toma la tupla i sobre el atributo A_j, y c_{ij} el grado de compatibilidad asociado a este valor.

El grado de compatibilidad se usa cuando se efectúa alguna operación (por ejemplo, una consulta) y sirve para almacenar el grado en el que un valor de un atributo concreto de una tupla concreta ha satisfecho dicha operación (p.e., la condición exigida por una consulta). Por este motivo, las relaciones base de la base de datos no tienen atributos de compatibilidad.

Definición 3.2.3 *Sea R una relación difusa generalizada dada por*

$$R = \begin{cases} \mathcal{H} = \{(A_1 : D_1[, C_1]), \ldots, (A_m : D_m[, C_m])\} \\ \mathcal{B} = \{(A_1 : \tilde{d_{i1}}[, c_{i1}]), \ldots, (A_m : \tilde{d_{im}}[, c_{im}])\} \end{cases} \tag{3.18}$$

con $i = 1, 2, \ldots, n$, n el número de tuplas de la relación. Entonces, se definen:

- ***Componente de valor*** *de una relación difusa generalizada, R^v, como la parte de la relación dada por:*

$$R^v = \begin{cases} \mathcal{H}^v = \{(A_1 : D_1), \ldots, (A_m : D_m)\} \\ \mathcal{B}^v = \{(A_1 : \tilde{d_{i1}}), \ldots, (A_m : \tilde{d_{im}})\} \end{cases} \tag{3.19}$$

donde $\mathcal{H}^v$ y $\mathcal{B}^v$ son las ***componentes de valor*** *de la Cabecera y el Cuerpo, respectivamente.*

- ***Componente de compatibilidad*** *de una relación difusa generalizada, R^c, como la parte de la relación dada por:*

$$R^c = \begin{cases} \mathcal{H}^c = \{[C_1], \ldots [, C_m]\} \\ \mathcal{B}^c = \{[c_{i1}], \ldots [, c_{im}]\} \end{cases} \tag{3.20}$$

donde $\mathcal{H}^c$ y $\mathcal{B}^c$ son las ***componentes de compatibilidad*** *de la Cabecera y el Cuerpo, respectivamente.*

En una relación (definición 3.2.2), el **grado de compatibilidad** del valor de un atributo concreto (en una tupla) se obtiene como consecuencia de los procesos de manipulación realizados sobre esa relación y expresa el grado con el que ese valor ha satisfecho la operación realizada sobre él.

Sobre estas definiciones, GEFRED redefine los operadores del álgebra relacional en la llamada **Álgebra Relacional Difusa Generalizada**: Unión, Intersección, Diferencia, Producto cartesiano, Proyección, Selección, Reunión y División. Estos operadores se definen indicando la cabecera y el cuerpo de una relación difusa generalizada que se obtendrá como el resultado de la operación. Todos esos operadores están definidos en [Medina et al, 1994, Medina, 1994], con excepción de la división, que se encuentra definida en [Galindo et al., 1997, Galindo, 1999]. Dado que ampliar esta parte nos situaría fuera del contexto de esta memoria, remitimos al lector a dichos trabajos para profundizar más en tales conceptos.

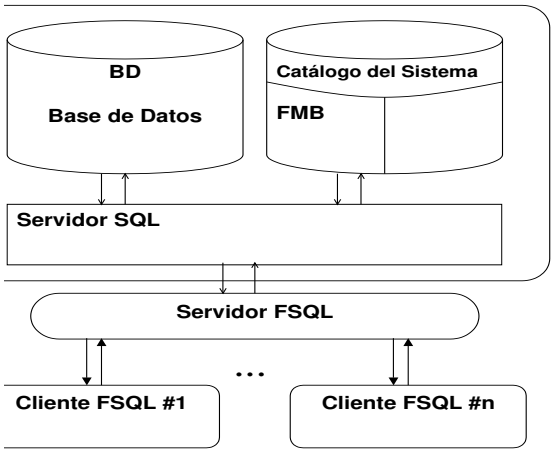
Los Sistemas de Bases de Datos Relacionales Difusas (SBDRD) extienden las capacidades de los sistemas clásicos, con el fin de poder manejar información difusa. Se construyen a partir de un modelo teórico y con unas directrices de construcción con las que se pueda abordar una implementación operativa.

3.2.2. Arquitectura FIRST

Basada en el modelo GEFRED, se ha desarrollado la arquitectura **FIRST** [Medina et al, 1995] para llevar a la práctica los objetivos propuestos en el modelo. Está construida sobre una arquitectura SBDR Cliente-Servidor, en concreto la suministrada por Oracle®, y aprovecha su misma estructura, añadiendo los nuevos componentes para manejar la información difusa.

Podemos ver un esquema de la arquitectura seguida en la figura 3.5. Resumidamente, las principales partes de que consta son las siguientes:

Figura 3.5: Arquitectura de FIRST



- **SBDR anfitrión**. Es el sistema usado como soporte para la implementación del modelo. Incluye la Base de Datos y el catálogo del sistema, así como las herramientas y recursos necesarios para manejar la información. Sobre éste se añade el servidor F-SQL.
- **BD**. Almacena toda la información permanente, tenga carácter difuso o no, siguiendo el modelo relacional.
- **FMB**. La Base de Metaconocimiento Difuso se construye bajo el catálogo del sistema anfitrión como una extensión del mismo, proporcionando información acerca de todas las estructuras, datos y definiciones difusas almacenadas en la BD.

- **Servidor de FSQL**. [Galindo et al, 1998] Amplía la capacidad del Servidor de SQL anfitrión para procesar peticiones difusas, expresadas en F-SQL (Fuzzy SQL, la extensión difusa sobre el lenguaje SQL que comentaremos a continuación). Para llevar eso a cabo, dispone de un procesador para la sintaxis F-SQL y de un protocolo para la comunicación con el cliente F-SQL. Asimismo, acepta peticiones SQL, que pasará sin necesidad de procesar al Servidor de SQL subyacente.

En el apéndice A resumiremos las principales características del lenguaje F-SQL al tiempo que describiremos el manual de usuario de la aplicación FuzzyQueries 2+ que, entre otras cosas, puede utilizarse como un cliente F-SQL.

3.3. Minería de datos difusa

Hasta ahora hemos visto como la teoría de los subconjuntos difusos nos proporciona una ayuda inestimable de cara a la representación de información afectada de incertidumbre, imprecisión o vaguedad en general, según se extrae de [Pedrycz, 1998a]. Gran parte de los problemas reales a los que hemos de enfrentarnos son intrínsecamente difusos. Por otro lado, nos puede interesar manejar información precisa mediante herramientas difusas, con el objeto de disminuir la granularidad o definir el contenido semántico de los resultados. El análisis de dicha información nos puede plantear algunos problemas como los que citamos a continuación:

- Por un lado, la definición de conjuntos difusos sobre el dominio de los datos genera una problemática asociada al cálculo del número de elementos que cumplen una determinada propiedad, expresando ésta como el grado de inclusión del elemento dentro del conjunto. Algunas soluciones al respecto se basan en la definición de un cardinal difuso, como las que vimos en el apartado 3.1.5. Este mismo planteamiento puede extenderse al cálculo de otras medidas descriptivas sobre datos de carácter difuso como, por ejemplo, la presentada en [Blanco et al., 2003].

- Por otro lado, el modelado de una situación real puede precisar de la definición de algún tipo de relación de similitud o semejanza (apartado

3.1.4) entre valores o elementos, en el caso de que necesitemos compararlos entre sí. La aplicación de este tipo de relaciones puede resultar útil en los casos en los que se desee enriquecer el modelo estableciendo un parecido semántico entre los datos.

De igual forma que es necesario extender los modelos de bases de datos para manejar información con carácter impreciso, también las técnicas de minería de datos y extracción de conocimiento han de adaptarse a las nuevas características de los datos.

3.3.1. Trabajos previos

Como veíamos en el apartado 2.3.5.2, uno de los principales problemas al trabajar con reglas de asociación es aquél referido a la baja granularidad que puede ofrecernos un conjunto de transacciones de ítems. Cuando el problema original se aplica sobre bases de datos con atributos categóricos o cuantitativos, nos enfrentábamos al problema de no hallar ítemsets lo suficientemente interesantes por no superar el umbral de mínimo soporte. La forma en que algunos autores [Srikant y Agrawal, 1996, Zhang et al., 1997] trataban de dar solución a dicho problema era mediante la definición de las reglas de asociación cuantitativas. Pero esto aún plantea algunos problemas como los que ya comentamos, relacionados con el ajuste del número de intervalos en los que dividir el dominio de los atributos.

Para solucionar esto último, algunos enfoques basados en la aplicación de conjuntos difusos a la minería de datos se muestran en trabajos como [Hong et al., 1999, Chien et al., 2001]. El camino seguido en ellos es el de convertir un conjunto de valores "crisp" en difusos, por medio del establecimiento de etiquetas lingüísticas como conjuntos difusos, y trabajar a continuación con éstas. Una forma de proceder similar la podemos encontrar en [Chan y Au, 1997], donde además se propone un algoritmo, F-APACS. Por último, en [Yang y Singhal, 1999] encontramos un interesante repaso de diversos enfoques difusos sobre reglas de asociación, emparentándolas con las dependencias funcionales (difusas).

Pero el principal problema del que parten los trabajos anteriores es que los datos de partida son valores ordinarios sobre los que se aplican conjuntos

difusos con el objeto, anteriormente comentado, de disminuir la granularidad de la información y aumentar la semántica de los resultados, para que éstos sean más accesibles al usuario final. En el siguiente apartado, veremos una definición de reglas de asociación difusas sobre información difusa de por sí.

3.3.2. Reglas de asociación difusas

En el capítulo anterior tratábamos el problema de extraer reglas de asociación en conjuntos de transacciones. Una de las formas representar una transacción es mediante un vector binario en el que cada posición corresponde a un ítem, indicando con un valor igual a 1 si el ítem aparece en la transacción, o 0 en caso contrario. Existen situaciones para las que esta representación resulta poco adecuada o insuficiente, y una de las propuestas que se ofrecen es la de trabajar con transacciones difusas.

Un problema común hoy día es el del estudio del uso que los usuarios hacen de Internet, de cara a proporcionarles un servicio de atención más eficiente. Esto incluye tanto filtrar aquella información que no les resulte útil como proporcionarles información relacionada potencialmente interesante. Esto último es muy interesante, por ejemplo, para el comercio electrónico. Una forma para conseguir esto es la minería sobre perfiles de usuario, de acuerdo con [Kimball y Merz, 2000]. Un perfil de usuario es un conjunto de datos con información acerca de las preferencias y el comportamiento de un usuario durante sus sesiones de navegación por páginas web. Y, de las herramientas existentes para lograr este cometido, las reglas de asociación parecen ser las más indicadas, tratando cada perfil como una transacción, y cada término de búsqueda, dirección URL, etc., como ítems.

El conocimiento que podamos extraer nos resultará más útil si en los perfiles encontramos información sobre la importancia relativa que el usuario da a los términos o sitios visitados. Por ejemplo, el hecho de que una persona visite una página web una sola vez no es indicativo de que la haya encontrado interesante. Para medir este factor, podemos usar conjuntos difusos [Kraft y Buell, 1993], transformando las estructuras correspondientes convenientemente. Es más, mediante el uso de conjuntos difusos para medir la importancia de un término en un perfil, se nos proporciona una mayor in-

formación sobre las preferencias de los usuarios [Martín-Bautista, 2000]. En este sentido, los perfiles de usuario pueden verse como conjuntos difusos, y la extracción de reglas de asociación sobre conjuntos de perfiles necesita del desarrollo de nuevos métodos.

Por otro lado, podemos hallar otros problemas inherentemente difusos en los que sea necesario aplicar directamente reglas de asociación difusas para poder manejarlos.

El modelo original de reglas de asociación [Agrawal et al., 1993] es extendido en [Delgado et al., 2003a], con el objeto de manejar información difusa en bases de datos. A continuación podemos encontrar un breve resumen de los conceptos más importantes.

Definición 3.3.1 ([Delgado et al., 2003a]) *Dado I un conjunto finito de ítems, llamamos transacción difusa a cualquier conjunto difuso no vacío $\tilde{\tau} \in I$.*

Notaremos por i_k a un ítem en $\tilde{\tau}$. Sea $\tilde{\tau}(i_k)$ el grado de pertenencia de i_k en $\tilde{\tau}$. De acuerdo con la anterior definición, es obvio que una transacción "crisp" u ordinaria es un caso especial de transacción difusa.

Podemos representar un conjunto de transacciones difusas por medio de una tabla, en la que las columnas representan a los ítems y las filas a las transacciones. La casilla para el ítem i_k y la transacción $\tilde{\tau}_j$ contiene un valor perteneciente al intervalo $[0, 1]$, el grado de pertenencia de i_k en $\tilde{\tau}$, $\tilde{\tau}(i_k)$. Podemos ver un ejemplo de lo anterior en la tabla 3.1.

Tabla 3.1: Tabla de transacciones difusas

	i_1	i_2	i_3	i_4
$\tilde{\tau}_1$	0,46	0	0	0,53
$\tilde{\tau}_2$	0,73	1	1	0
$\tilde{\tau}_3$	0	0	1	1
$\tilde{\tau}_4$	0	0,49	0,34	1

Sea $T \in I$ un conjunto de transacciones difusas (FT-set, de aquí en adelante). Entonces, representaremos el ítem i_k en T como,

$$\tilde{\Gamma}_{i_k} = \sum_{\tilde{\tau} \in T} \tilde{\tau}(i_k)/\tilde{\tau} \tag{3.21}$$

Sea $I_0 \in I$ un ítemset, es decir, un conjunto de ítems, que representaremos en T de la siguiente forma,

$$\tilde{\Gamma}_{I_0} = \bigcap_{i \in I_0} \tilde{\Gamma}_i = min_{i \in I_0} \tilde{\Gamma}_i \tag{3.22}$$

Análogamente al caso clásico, definimos una regla de asociación difusa de acuerdo a la siguiente definición.

Definición 3.3.2 ([Delgado et al., 2003a]) *Una **Regla de Asociación Difusa** es una implicación de la forma $A \Rightarrow C$ tal que $A, C \subset I$ y $A \cap C = \emptyset$. Llamaremos a A y C antecedente y consecuente, respectivamente.*

De nuevo, esta definición se refiere a la extensión al caso difuso de las reglas de asociación booleanas, en el que se trabaja con conjuntos de transacciones. Nuestro objetivo va más orientado a la aplicación de estas técnicas sobre bases de datos relacionales. Al igual que veíamos cómo era posible transformar un conjunto de transacciones en una tabla relacional y viceversa (apartado 2.2), es posible seguir un proceso análogo para extraer reglas de asociación difusas en bases de datos relaciones difusas.

Por otro lado, volviendo al caso de las reglas de asociación cuantitativas (apartado 2.3.5.2), encontrábamos algunos problemas derivados de la división en intervalos del dominio de los atributos. Si el número de intervalos es muy alto, el soporte de éstos puede resultar demasiado bajo como para ser tenido en consideración, mientras que si el número de intervalos decrece, aumentando entonces su amplitud, podemos pasar por alto información de posible interés. Si en lugar de restringirnos a intervalos, dividimos un dominio en distribuciones de posibilidad, podemos hallar un equilibrio entre los dos problemas citados.

Consideremos el siguiente caso. Sea RE un conjunto de atributos, $RE = \{At_1, \ldots, At_m\}$. Sea además $dom(At_k) = \{a_{k_1}, \ldots, a_{k_n}\}$ un dominio de etiquetas lingüísticas $\forall At_k \in RE$. Tomando de nuevo I como un conjunto de ítems, podemos especificar que cada ítem represente a un par *'At_k es a_{k_i}'*, y

T, un FT-set, notaremos por $\tilde{\tau}(At_k \ es \ a_{k_i})$ al grado en que se cumple dicha propiedad para cada $\tilde{\tau} \in T$.

Podemos refinar aún más esta representación considerando una relación difusa en lugar de un conjunto de transacciones difusas. Abusando de la notación, usaremos la misma representación tanto para una transacción difusa como para una tupla difusa. En ese caso, la celda para el atributo At_k y la tupla difusa $\tilde{\tau}_i$ contendría un par $< a_{k_i}, \mu_{\tilde{\tau}_i}(At_k) >$, donde $\mu_{\tilde{\tau}_i}(At_k)$ sería el grado de pertenencia de At_k a $\tilde{\tau}_i$. Un ejemplo de dicha representación es el mostrado en la tabla 3.2.

Tabla 3.2: Relación difusa

	At_1	At_2	...
$\tilde{\tau}_1$	$a_{1_1}, 0{,}46$	$a_{2_1}, 0$	...
$\tilde{\tau}_2$	$a_{1_2}, 0{,}73$	$a_{2_2}, 1$	...
$\tilde{\tau}_3$	$a_{1_3}, 0$	$a_{2_3}, 0$	...
$\tilde{\tau}_4$	$a_{1_4}, 0$	$a_{2_4}, 0{,}49$	...

3.3.2.1. Valoración de reglas de asociación difusas

Cuando trabajamos con bases de datos difusas, para medir el interés y la bondad de una regla de asociación difusa hemos de aplicar herramientas de razonamiento aproximado, debido a la imprecisión que puede afectar a las transacciones difusas. De acuerdo con lo expuesto en [Delgado et al., 2003a], utilizaremos un enfoque semántico basado en la evaluación de sentencias cuantificadas [Zadeh, 1983]. Para ello, utilizaremos un cuantificador coherente, Q, tal y como fue definido en el apartado 3.1.6.

Definición 3.3.3 ([Delgado et al., 2003a]) *El soporte de un ítemset es igual al resultado de evaluar la sentencia cuantificada Q de T son $\tilde{\Gamma}_{I_0}$.*

Definición 3.3.4 ([Delgado et al., 2003a]) *El soporte de la regla de asociación $A \Rightarrow C$ en el FT-set T, $Supp(A \Rightarrow C)$, equivale a la evaluación de la sentencia cuantificada Q de T son $\tilde{\Gamma}_{A \cup C} = Q$ de T son $(\tilde{\Gamma}_A \cap \tilde{\Gamma}_C)$.*

Definición 3.3.5 ([Delgado et al., 2003a]) *La confianza de la regla de asociación $A \Rightarrow C$ en el FT-set T, $Conf(A \Rightarrow C)$, se corresponde con el resultado de la evaluación de la sentencia cuantificada Q de $\tilde{\Gamma}_A$ son $\tilde{\Gamma}_C$.*

El caso del factor de certeza es más sencillo, y podemos seguir calculando de la misma forma en que lo hacíamos en el caso crisp (definición 2.3.5).

Para evaluar el tipo de sentencias utilizadas en las definiciones 3.3.4 y 3.3.5 vamos a seguir el método *GD*, que ya fue introducido en el apartado 3.1.6.

Nos hemos decidido por el cuantificador Q_M, definido por $Q_M(x) = x$, por ser coherente y porque además, las medidas obtenidas al usarlo en las definiciones 3.3.3, 3.3.4 y 3.3.5 se corresponden con las medidas ordinarias para el caso "crisp", como muestran las siguientes proposiciones:

Proposición 3.3.1 ([Delgado et al., 2003a]) *Sea $I_0 \subset RE$ tal que $\tilde{\Gamma}_{I_0}$ es crisp. Entonces, $supp(I_0)$ calculado a través de GD con Q_M es el soporte ordinario del ítemset.*

Proposición 3.3.2 ([Delgado et al., 2003a]) *Sea $A \Rightarrow C$ una regla de asociación clásica sobre T. Entonces, $Supp(A \Rightarrow C)$ calculado mediante GD utilizando Q_M es el soporte ordinario de la regla.*

Proposición 3.3.3 ([Delgado et al., 2003a]) *Sea $A \Rightarrow C$ una regla de asociación clásica sobre T. $Conf(A \Rightarrow C)$ calculado mediante GD usando Q_M es la confianza ordinaria de la regla.*

3.4. Resumen

En este capítulo hemos abordado las extensiones difusas de conceptos tales como bases de datos y herramientas de minería de datos, necesarios como base para los resultados que expondremos en los siguientes capítulos. Hemos hecho especial hincapié en resultados, en ambos campos, tales como un modelo relacional para bases de datos difusas y una extensión difusa sobre las reglas de asociación. La teoría de subconjuntos difusos, cuyos conceptos más relevantes con respecto a nuestra investigación han sido recordados también en este capítulo, constituirá una excelente herramienta de cara a obtener los objetivos propuestos.

4. Dependencias Aproximadas Difusas

Este capítulo puede entenderse perfectamente como el corazón de la presente memoria. A lo largo de la misma, hemos puesto en situación al lector recordando los conceptos más relevantes sobre el modelo relacional de bases de datos para la representación y manejo de conocimiento en un ordenador. De igual forma, hemos introducido algunas técnicas y herramientas de minería de datos y extracción de conocimiento, aquéllas más interesantes de cara a conseguir nuestros objetivos, que no son otros que los de extraer información no trivial y de potencial utilidad a partir de ese conocimiento almacenado.

Pero, dado que nuestro trabajo parte de la necesidad de trabajar con información que pueda venir afectada por imprecisión o incertidumbre, también hemos repasado cómo es posible, por medio de la Teoría de Subconjuntos Difusos, extender el modelo relacional clásico para permitir este tipo de información, así como algunas de las herramientas clásicas en Minería de datos, tales como las reglas de asociación difusas.

Partiendo de los precedentes expuestos, estamos en disposición de definir

en este capítulo el concepto de Dependencia Aproximada Difusa como una interesante herramienta de extracción de conocimiento impreciso que además continúa los avances al respecto que se han venido desarrollando hasta el momento presente.

4.1. Introducción

Como ya hemos visto en capítulos anteriores, la Extracción de Conocimiento (EC) en Bases de datos tiene como principal finalidad la de hallar información previamente desconocida y potencialmente útil en bases de datos. Básicamente, el proceso se puede dividir en tres etapas: preprocesamiento de la información (preparación de los datos), minería de datos (búsqueda de patrones interesantes en los datos) e interpretación de los resultados de la etapa anterior de cara a proporcionar la información final.

Existen varias maneras en las que la aplicación de la teoría de subconjuntos difusos es útil en la Extracción de Conocimiento [Pedrycz, 1998a]. En primer lugar, la participación del usuario en todas las etapas del proceso de EC resulta crucial, ya que, en particular, el objetivo primordial de la EC es el de proporcionar al usuario una información que pueda comprender. Las técnicas basadas en conjuntos difusos constituyen una herramienta apropiada para este propósito, especialmente en aquellos casos en los que deseamos expresar resúmenes o relaciones entre datos numéricos por medio de términos lingüísticos. Además, la expresión mediante etiquetas lingüísticas de los patrones resultan de gran ayuda de cara a procesar éstos en el paso final de la EC.

Sin embargo, existe otro importante factor que enlaza la EC y los conjuntos difusos: en muchas ocasiones, la información de partida es inherentemente imprecisa o incierta. Para tratar con este problema, se han estudiado y desarrollado gran cantidad de modelos de bases de datos: relacionales difusos, deductivos y orientados a objetos, por ejemplo. Un caso más común, y que proporciona un escenario similar, es el de los datos difusos que se obtienen a partir de datos “crisp” en la etapa de preprocesamiento por medio de agregaciones, resúmenes o cambios en el nivel de granularidad. El análisis de tal tipo de información requiere de herramientas específicas, desarrolladas como

extensiones difusas a partir de lo ya existente.

Para ilustrar este punto, veamos un ejemplo sencillo. Consideremos la tabla 4.1, donde encontramos Cat y Sal, dos atributos que representan la categoría laboral y el salario de un conjunto de trabajadores. Supongamos que la categoría toma valores dentro de un conjunto de etiquetas que indican el tipo de trabajo, por ejemplo $\{Gestor, Comercial, \ldots\}$ y que el salario es un valor numérico. Suele ser común la existencia de una relación directamente proporcional entre la categoría y el salario (a mayor categoría, mayor salario). Sin embargo, si queremos dar una representación de esta relación, las reglas del tipo *"Si Cat=Gestor, entonces Sal=123455"* no son la mejor solución, ya que pueden existir muchos valores numéricos diferentes para el salario de un Gestor, y la precisión y el contenido semántico de este tipo de reglas sería muy pobre. Lo preferible sería mostrar reglas como *"Si Cat=Gestor, entonces Sal=Alto"*, donde *Alto* sería una etiqueta lingüística asociada a un conjunto difuso definido sobre el dominio del salario. La extracción y evaluación de tales reglas requiere del desarrollo de herramientas específicas. Es decir, debemos extender las técnicas correspondientes de minería de datos al caso difuso. Las técnicas tratadas en este capítulo caen dentro de este campo.

Tabla 4.1: Ejemplo de una relación de empleados

	Cat	Sal
t_1	$Gestor$	123455
t_2	$Programador$	80177
t_3	$Comercial$	95075
t_4	$Gestor$	125776
t_5	$Programador$	79088
t_6	$Programador$	78797
$\vdots$	$\vdots$	$\vdots$

Nuestro objetivo en este capítulo es el de proporcionar una definición para la Dependencia Aproximada Difusa, como extensión del concepto de Dependencia Aproximada para el caso en que trabajemos sobre bases de datos difusas. Para ello, nos apoyaremos en algunos resultados ya existentes, tales

como la extracción de dependencias aproximadas por medio de reglas de asociación, por un lado, y la definición de reglas de asociación difusas, por otro. A partir de estas definiciones, otro objetivo adicional es el de proporcionar una metodología eficiente para la extracción de dependencias aproximadas difusas, adaptando los algorimos clásicos de extracción de reglas de asociación. Una de las principales ventajas a tener en cuenta sobre nuestra metodología es que no incrementamos en exceso la complejidad en tiempo y en espacio, ya que como veremos, éstas son únicamente multiplicadas por una constante.

4.2. Extensiones difusas de la dependencia funcional

Veíamos en el apartado 2.1.3, cuando definíamos una dependencia funcional, que la obtención de las mismas era bastante difícil, debido a lo restrictivo de sus condiciones de existencia. De hecho, una sola excepción a la regla general hace que la dependencia no se cumpla. Por este motivo, y con diversos objetivos, la definición de dependencia funcional se ha extendido mediante la introducción de algún tipo de imprecisión, incertidumbre o gradualidad, dando lugar a diversos tipos de dependencias "suavizadas".

Se puede hablar, principalmente, de dos aproximaciones a la extensión de dependencias funcionales, las dependencias funcionales difusas y las dependencias aproximadas, que ya introdujimos en el apartado 2.3.6.

En las primeras, se suele introducir algún grado de imprecisión en la definición, aumentando la granularidad del dominio de los atributos, suavizando el operador de igualdad en una relación de semejanza, suavizando el cuantificador o la implicación, o varias de estas opciones al mismo tiempo (como se nos describe en [Bosc et al., 1997], [Cubero et al., 1994a], [Cubero et al., 1994b] o [Cubero et al., 1998a]).

Algunas de las aplicaciones prácticas que se han propuesto para este tipo de dependencias son las de diseño de bases de datos, ingeniería inversa, resumen de información, optimización de consultas y, en general, comprensión de la estructura de la información. Las dependencias funcionales graduales proporcionan un cierto tipo de meta-conocimiento sobre asociaciones entre atributos basadas en la asociación entre valores de atributos. Por otro lado, las dependencias funcionales extendidas, con un grado de cumplimiento en

$\{0, 1\}$, pueden aplicarse al diseño de bases de datos.

Cada uno de los distintos enfoques en la definición de dependencias funcionales difusas introduce cierta imprecisión en uno o más de los elementos de la definición de dependencias funcionales. Aunque algunos elementos pueden extenderse al mismo tiempo, en los siguientes subapartados, extraídos de [Bosc et al., 1997, Cubero et al., 1998a, Delgado et al, 1999a], se considera sólo un elemento cada vez.

4.2.1. Dependencias funcionales sobre una relación difusa

En este enfoque, la relación r sobre la que trabajamos es difusa, y cada tupla tiene asociado un grado de pertenencia a r. Una definición de dependencia funcional extendida, propuesta por Kiss [Kiss, 1990, Kiss, 1991], es la siguiente:

$$T_r(X \rightarrow Y) = inf_{t,s \in r} min\{\mu_r(t), \mu_r(s), S_=(t[X], s[X])\} \rightarrow S_=(t[Y], s[Y]) \tag{4.1}$$

donde $S_=$ es la función clásica de igualdad. En [Bosc et al., 1997], se interpreta como asociada al grado de pertenencia de las excepciones a la relación. Cuanto menor éste, mayor el grado de asociación entre X e Y.

4.2.2. Relajación de la igualdad

La exigencia de igualdad entre los valores de atributos se relaja como una relación de semejanza. Una relación de semejanza (apartado 3.1.5) es una relación difusa a la que sólo se le exige que cumpla las propiedades reflexiva y simétrica. Se ha de definir una relación de semejanza para cada atributo del esquema relacional. Una dependencia funcional difusa puede definirse como

$$X \rightarrow Y \Leftrightarrow \forall t, s \in r, S_X(t[X], s[X]) \Rightarrow S_Y(t[Y], s[Y]) \tag{4.2}$$

Si la implicación es "crisp", el valor que se obtiene es también "crisp". Aunque también podría aplicarse la implicación de Rescher-Gaines, definida como $I(a, b) = 1$, si $a \leq b$, o 0 en otro caso. Si la implicación es difusa, hemos

de substituir el cuantificador universal por una t-norma sobre los valores de la implicación, y en tal caso obtendríamos un grado difuso.

Una interpretación válida es: "Cuanto más parecidos son los valores de X, más parecidos son los de Y".

Algunos autores como [Shenoi et al., 1992] y [Prade y Testemale, 1984] han abordado el problema de extender la definición de dependencia funcional siguiendo este criterio. En el primero, se establece una relación de similitud (ver apartado 3.1.6) tanto entre los valores del antecedente como en los del consecuente, mientras que en el segundo, se aplica una relación de semejanza sólo para los valores del consecuente.

4.2.3. Incorporación de información imprecisa

En este caso, lo que se hace es reemplazar los valores concretos de los atributos por unos más generales. De esta forma se introduce un cierto grado de imprecisión en los datos. Más concretamente, el valor crisp se substituye por un valor difuso, generalmente una etiqueta lingüística [Bosc et al., 1996, Cubero et al., 1998b]. El grado de compatibilidad entre el valor numérico crisp y la correspondiente etiqueta nos mide la imprecisión introducida por la substitución. Un caso particular de conjunto de etiquetas es el de las jerarquías crisp, en el que subimos al nivel superior de la jerarquía reemplazando los valores del nivel previo que aparecen en la relación con generalizaciones de los valores en el nivel actual. En este caso la imprecisión se asume en forma de valores disyuntivos, ya que cada término a un determinado nivel representa un subconjunto del dominio del atributo. Una posible definición sería la siguiente

$$(X, F) \rightarrow (Y, G) \Leftrightarrow \forall t \in r, F(t[X]) \Rightarrow G(t[Y]) \tag{4.3}$$

lo cual puede interpretarse como que "cuanto más F es X, más G es Y", siendo F y G dos etiquetas lingüísticas. De todas formas, los dominios de los atributos han de ser generalizados (es decir, se han definir las etiquetas sobre ellos), para que la dependencia se cumpla. Este tipo de dependencias puede usarse para resumir la relación original mediante el uso de etiquetas lingüísticas sobre el dominio en lugar de valores concretos.

4.2.4. Dominios imprecisos

En algunos modelos de bases de datos difusas, podemos encontrar atributos cuyo dominio es, al menos en parte, impreciso. Una tupla puede contener tanto valores concretos como imprecisos para un atributo dado. La cuestión principal es la de definir la medida de similitud entre valores del dominio. Cuando éstos son concretos, la comparación se realiza mediante el operador clásico de igualdad, pero el problema estriba en que uno o ambos valores sean imprecisos, en cuyo caso se admiten dos enfoques distintos, el enfoque sintáctico, que compara la igualdad entre etiquetas, de forma que $\mu_{=}(etiq_i, etiq_i) = 1$ y $\mu_{=}(etiq_i, x) = 0$ para cualquier otro valor x del dominio, y el enfoque semántico, en el que las etiquetas se comparan de acuerdo a su representación como distribuciones de posibilidad. El sintáctico compara las etiquetas una a una, de forma autoexcluyente. En cambio el semántico trata las etiquetas en términos de su representación como distribuciones de posibilidad. En este último caso, dados un atributo X sobre el dominio D_X, y una etiqueta L y un valor concreto v sobre el mismo dominio D_X, tenemos que

$$S_X(L, v) = L(v) \tag{4.4}$$

Por otro lado, dadas dos etiquetas L y M sobre D_X, se puede definir una relación de semejanza débil [Cubero et al., 1994b] como

$$S_X(L, M) = sup_{u,v} min(L(u), M(v), S_X(u, v)) \tag{4.5}$$

Si S_X es la igualdad clásica para valores concretos, la expresión 4.5 se torna en la medida propuesta por Chen [Chen et al, 1991, Chen et al, 1992],

$$S_X(L, M) = sup_u min(L(u), M(u)) \tag{4.6}$$

También en [Cubero et al., 1994b], se define una relación de semejanza fuerte como

$$S_X(L, M) = inf_{u,v} max(1 - L(u), 1 - M(v), S_X(u, v)) \tag{4.7}$$

Algunos autores como Chen y Cubero han seguido este enfoque. La definición de dependencia funcional difusa debida a Chen es la siguiente

$$X \rightarrow_{\varphi} Y \Leftrightarrow \forall t, s \in r \; si \; t[X] = s[X], t[Y] = s[Y], \tag{4.8}$$

$$si \; no, (S_X(t[X], s[X]) \Rightarrow_{G\ddot{o}} S_Y(t[Y], s[Y])) \geq \varphi$$

donde φ es un umbral en $[0, 1]$. La parte del "si" no es necesaria si las etiquetas sobre D_X y D_Y están normalizadas. Esta definición usa la implicación de Gödel, definida como $I(a, b) = 1$ si $a \leq b$, y b en otro caso. Podemos desestimar el umbral y obtener a cambio un grado de cumplimiento de la dependencia reemplazando la implicación por una "crisp", como se describe en [Bosc et al., 1997].

La definición de dependencia funcional difusa debida a Cubero tiene la semántica de que "si los valores del antecedente son iguales, se espera que los valores del consecuente sean fuertemente semejantes". Sea S_Y una relación de semejanza fuerte definida sobre D_Y. Entonces

$$X \rightarrow_{\varphi} Y \Leftrightarrow \forall t, s \in r \; si \; t[X] = s[X], S_Y(t[Y], s[Y]) \geq \varphi \tag{4.9}$$

donde φ es un umbral de semejanza asociado al atributo Y. Cada atributo ha de tener asociado un umbral que defina el mínimo grado en que dos valores son semejantes. Esta definición es menos estricta que la de una dependencia funcional clásica, aunque es posible encontrar una definición más estricta, involucrando una relación de semejanza y su correspondiente umbral para X.

4.2.5. Relajación del cuantificador universal

La idea es la de permitir la existencia de excepciones a la dependencia. La medida de cumplimiento se obtiene contando las excepciones y calculando el porcentaje de tuplas que cumplen la dependencia mediante un cuantificador difuso. Esto en la práctica se corresponde a las dependencias aproximadas [Sánchez, 1999, Delgado et al., 2000a, Blanco et al., 2000], que también pueden encontrarse en la literatura como determinaciones parciales o dependencias funcionales parciales. Como ya vimos en el apartado 2.3.6, estas generalizaciones de la dependencia funcional no tienen por qué ser de carácter difuso.

4.3. Trabajos relacionados

En los capítulos preliminares, se introdujeron algunos conceptos alusivos a las herramientas existentes sobre Minería de Datos y Extracción de Conocimiento, que constituirán la base a partir de la cual asentar nuestra nueva definición. Nos referimos en concreto a la obtención de Dependencias Aproximadas a partir de su modelo, consistente en un conjunto de Reglas de Asociación (como se describe en [Pfahringer y Kramer, 1995, Blanco et al., 2000]). Extendiendo tales medidas por medio de la teoría de los subconjuntos difusos, nos hallamos con las Reglas de Asociación Difusas, donde nos quedaremos con la definición expuesta en [Delgado et al., 2003a] (ver apartado 3.3.2). En la siguiente sección veremos cómo es posible fusionar ambas definiciones para lograr el objetivo propuesto en este capítulo.

Hasta donde llega nuestro conocimiento, la metodología que exponemos en este capítulo es relativamente novedosa y original, ya que no hemos sido capaces de hallar ningún trabajo análogo en la bibliografía existente. No obstante, hemos encontrado algunos enfoques similares a los que conviene hacer mención. Extendiendo al caso difuso la definición de dependencia aproximada dada por [Huhtala et al., 1998], basada en la definición de particiones sobre el conjunto de tuplas de una relación, nos encontramos con los trabajos expuestos en [Wang y Tsai, 1999, Wang et al., 1999, Wang et al., 2000].

4.3.1. El enfoque de Wang et al.

Wang et al. presentan una nueva técnica de minería de datos para la extracción de dependencias aproximadas en bases de datos difusas, sobre las que existen definidas una serie de relaciones de proximidad o semejanza (que en posteriores trabajos extiende a similitud).

Los autores se adhieren a lo expuesto en [Shenoi y Melton, 1989], según lo cual el uso de bases de datos basadas en relaciones de semejanza o similitud (en función de las restricciones que queramos imponer sobre los datos), viene especialmente indicado para la descripción y el manejo de información categórica sobre dominios discretos. Frente a esta idea, tenemos los modelos basados en conjuntos difusos, que resultan más apropiados para su aplicación sobre dominios numéricos (algo de lo que ya hablamos con anterioridad en

esta memoria, en el apartado 3.1.2).

Tabla 4.2: Relación difusa de empleados

Emp#	*Categoría*	*Exp.*	*Salario*
1	Vendedor	3	$37K$
2	Ingeniero en Diseño	10	$40K$
3	Ingeniero de Sistemas	5	$45K$
4	Ingeniero en Software	5	$45K$
5	Contable	12	$47K$
6	Contable	5	$50K$
7	Secretaria	10	$53K$
8	Secretaria	15	$55K$

La definición que estos autores nos ofrecen es la siguiente. Una dependencia aproximada sobre un esquema relacional RE se puede expresar como $X \rightarrow A$, donde $X \subseteq RE$ y $A \in RE$. Informalmente, una dependencia aproximada $X \rightarrow A$ se cumple si todas las tuplas que coinciden aproximadamente en X coinciden también de forma aproximada en A.

Formalmente, la dependencia se cumple o es válida en una cierta relación difusa r sobre RE si para cada par de tuplas t_i y $t_l \in r$ tenemos:

$$Si\ [t_i]_{D_j}^{\alpha_j} = [t_l]_{D_j}^{\alpha_j} para\ todo\ D_j \in X, entonces\ [t_i]_A^{\alpha_j} = [t_l]_A^{\alpha_j} \tag{4.10}$$

donde $[t_i]_{D_j}^{\alpha_j}$ representa la clase de equivalencia de la tupla t_i con respecto a un atributo D_j a un cierto nivel α_j. La notación se explica en más detalle a continuación.

Dos tuplas t_i y t_l serán equivalentes con respecto a un atributo D_j a un cierto nivel α_j si t_{ij} y t_{lj} pertenecen a la misma clase de equivalencia de D_j. Las clases de equivalencia de D_j vienen determinadas por el nivel α_j y definidas por una relación de similitud dada. En general, un atributo D_j particiona el conjunto de las tuplas de una relación en un conjunto de clases de equivalencia. La clase de equivalencia de una tupla $t_i \in r$ con respecto a un atributo D_j a un nivel α_j se nota como $[t_i]_{D_j}^{\alpha_j}$, es decir,

Tabla 4.3: Relaciones de similitud sobre los dominios de atributos de la tabla 4.2

CAT.	$Ing.Sw$	Con	$Ing.Sis$	$Vend$	$Ing.Dis$
$Secr$	0.6	0.7	0.6	0.5	0.6
$Ing.Sw$		0.6	0.8	0.5	0.8
Con			0.6	0.5	0.6
$Ing.Sis$				0.5	0.8
$Vend$					0.5

EXP.	5	10	12	15
3	0.9	0.7	0.7	0.5
5		0.7	0.7	0.5
10			0.9	0.7
12				0.7

SAL.	40	45	47	50	53	55
37	0.9	0.7	0.7	0.5	0.5	0.5
40		0.7	0.7	0.5	0.5	0.5
45			0.9	0.5	0.5	0.5
47				0.5	0.5	0.5
50					0.9	0.9
53						0.9

$$[t_i]_{D_j}^{\alpha_j} = \{t_l \in r | t_{lj} \approx_{\alpha_j} t_{ij}\} \tag{4.11}$$

El conjunto de clases de equivalencia $\pi_{D_j}^{\alpha_j} = \{[t_i]_{D_j}^{\alpha_j} | t_i \in r\}$ es una partición de r bajo D_j a un cierto nivel α_j. O lo que es lo mismo, $\pi_{D_j}^{\alpha_j}$ es una colección de conjuntos disjuntos (clases de equivalencia) de tuplas, tal que los elementos de cada conjunto pertenecen a una clase de equivalencia en D_j, y la unión de los conjuntos equivale a la relación r. El rango $|\pi|$ de una partición es el número de clases de equivalencia en π.

Los autores se apoyan en el concepto de refinamiento de una partición para obtener las dependencias aproximadas. Se dice que una partición π es un refinamiento de otra partición π' si cada clase de equivalencia en π es un subconjunto de alguna clase de equivalencia en π'. Se puede enunciar entonces el siguiente lema,

Lema 4.3.1 ([Huhtala et al., 1998]) *Una dependencia aproximada $X \rightarrow A$ se cumple si y sólo si π_X refina a $\pi_{\{A\}}$.*

Incluso existe una forma más simple de hallar la dependencia aproximada $X \rightarrow A$. Si π_X refina a $\pi_{\{A\}}$, entonces si se añade A a X ninguna de las clases de equivalencia de π_X se ve incrementada, por lo que $\pi_{X \cup \{A\}} = \pi_X$. En consecuencia, también nos encontramos el siguiente lema,

Lema 4.3.2 ([Huhtala et al., 1998]) *Una dependencia aproximada $X \rightarrow A$ se cumple si y sólo si $|\pi_X| = |\pi_{X \cup \{A\}}|$.*

La principal novedad que presentan Wang et al. en sus trabajos es la introducción de relaciones de semejanza y similitud para trabajar con bases de datos relacionales difusas, siguiendo la idea propuesta en [Huhtala et al., 1998], y utilizando una versión extendida del algoritmo propuesto en dicho trabajo.

Por contra, de lo extraído a partir de sus trabajos, nos encontramos con el inconveniente de que no se define ninguna medida que nos informe acerca de la certeza o interés de los resultados.

4.4. Definición

Como hemos visto en la sección 4.2, es posible extender el concepto de dependencia funcional de varias formas, suavizando alguno de los elementos de la regla definida por la ecuación 2.1. Nuestra intención es la de considerar tantos casos como podamos, integrando tanto dependencias aproximadas (excepciones) como dependencias difusas. Con esta idea en mente, además de admitir excepciones a la regla, hemos considerado la relajación de algunos elementos de la definición de una dependencia funcional, lo cual nos permite tener en cuenta algunos de los enfoques descritos en [Bosc et al., 1997]. En particular, vamos a considerar un grado de pertenencia asociado a los pares $< atributo, valor >$, como en el caso de las reglas de asociación difusas, pero también usaremos relaciones difusas de similitud para suavizar la igualdad de la ecuación 2.1.

La notación que usaremos será la siguiente. Sea $RE = \{At_1, \ldots, At_m\}$ un esquema relacional, y r una relación difusa sobre RE en los siguientes términos: la intersección entre un atributo At_k y una tupla difusa $\tilde{t}$ será un par $< \tilde{t}(At_k), \mu_{\tilde{t}}(At_k) >$, siendo $\tilde{t}(At_k)$ el valor de At_k en $\tilde{t}$, y $\mu_{\tilde{t}}(At_k)$ el grado de cumplimiento asociado. La tabla 4.4 muesta un ejemplo con una relación difusa, r, definida sobre un esquema relacional $RE = \{A, B, C\}$.

Tabla 4.4: Una relación difusa, r

	A	B	C
$\tilde{t}_1$	$a_1, 0{,}46$	$b_1, 0{,}76$	$c_1, 0{,}53$
$\tilde{t}_2$	$a_1, 0{,}73$	$b_2, 0{,}06$	$c_1, 0{,}31$
$\tilde{t}_3$	$a_1, 0{,}4$	$b_2, 0{,}28$	$c_1, 0{,}66$
$\tilde{t}_4$	$a_2, 0{,}41$	$b_1, 0{,}49$	$c_1, 0{,}34$

Consideremos además S_{At_i}, una relación difusa de similitud sobre el dominio del atributo At_i, $dom(At_i)$. Sea $S_{RE} = \{S_{At_k} | At_k \in RE\}$. Para ser más precisos, vamos a suponer que las relaciones de S_{RE} son max-min transitivas, es decir,

$$S_{At_k}(x_i, x_j) \geq \bigvee_{l=1}^{n} min(S_{At_k}(x_i, x_l), S_{At_k}(x_l, x_j)), \forall x_i, x_j \in dom(At_k) \quad (4.12)$$

Siguiendo con el ejemplo que venimos describiendo, la tabla 4.5 muestra las relaciones difusas de similitud definidas sobre los atributos de la relación difusa r.

Tabla 4.5: Relaciones difusas de similitud para A y B

a_2	0.3		b_2	0.8
	a_1			b_1

Vamos a definir las dependencias aproximadas difusas en una relación como reglas de asociación difusas sobre un FT-set (conjunto de transacciones difusas) determinado, obtenido a partir de dicha relación, siguiendo la misma idea de partida que vimos en la sección 2.3.6, mediante la que veíamos cómo las dependencias aproximadas se definían como reglas de asociación sobre un T-set (conjunto de transacciones) dado.

Definición 4.4.1 *Sea $I_{RE} = \{it_{At_k} | At_k \in RE\}$ el conjunto de ítems asociados al conjunto de atributos RE. Definimos el FT-set T'_r asociado a la tabla r con atributos en RE como sigue: para cada par de tuplas $< \tilde{t}, \tilde{s} >$ en $r \times r$, tenemos una transacción difusa $\widetilde{ts}$ en T'_r dada por*

$$\widetilde{ts}(it_{At_k}) = min(\mu_{\tilde{t}}(At_k), \mu_{\tilde{s}}(At_k), S_{At_k}(\tilde{t}(At_k), \tilde{s}(At_k))) \; \forall it_{At_k} \in T'_r \quad (4.13)$$

De esta forma, el grado de pertenencia de un cierto ítem it_{At_k} a la transacción asociada a las tuplas $\tilde{t}$ y $\tilde{s}$ tiene en cuenta el grado de pertenencia del valor de At_k en ambas tuplas junto con la similitud entre ellos. Este valor representa el grado en el que las tuplas $\tilde{t}$ y $\tilde{s}$ coinciden sobre At_k, es decir, el tipo de ítems que están relacionados por la ecuación 2.1. Partiendo de la relación que mostrábamos en la tabla 4.4 y siguiendo nuestra definición, el conjunto de transacciones difusas correspondiente sería el que aparece en la tabla 4.6.A.

Representaremos el ítem i_{At_k} en T'_r como,

$$\tilde{\Gamma}_{i_{At_k}} = \sum_{\tilde{t} \in T'_r} \tilde{t}(i_{At_k})/\tilde{t} \tag{4.14}$$

Sea $I_0 \in I_{RE}$ un ítemset, que representaremos en T'_r de la siguiente forma,

$$\tilde{\Gamma}_{I_0} = \bigcap_{i \in I_0} \tilde{\Gamma}_i = min_{i \in I_0} \tilde{\Gamma}_i \tag{4.15}$$

Tabla 4.6: **(A)** El FT-set T'_r obtenido a partir de r **(B)** Dep. Aproximadas Difusas en r (reglas de asociación difusas en T'_r)

	it_A	it_B	it_C
$\widetilde{t_1t_1}$	0.46	0.76	0.53
$\widetilde{t_1t_2}$	0.46	0.06	0.31
$\widetilde{t_1t_3}$	0.4	0.28	0.53
$\widetilde{t_1t_4}$	0.3	0.49	0.34
$\widetilde{t_2t_1}$	0.46	0.06	0.31
$\widetilde{t_2t_2}$	0.73	0.06	0.31
$\widetilde{t_2t_3}$	0.4	0.06	0.31
$\widetilde{t_2t_4}$	0.3	0.06	0.31
$\widetilde{t_3t_1}$	0.4	0.28	0.53
$\widetilde{t_3t_2}$	0.4	0.06	0.31
$\widetilde{t_3t_3}$	0.4	0.28	0.66
$\widetilde{t_3t_4}$	0.3	0.28	0.34
$\widetilde{t_4t_1}$	0.3	0.49	0.34
$\widetilde{t_4t_2}$	0.3	0.06	0.31
$\widetilde{t_4t_3}$	0.3	0.28	0.34
$\widetilde{t_4t_4}$	0.41	0.49	0.34

A

$[B] \rightarrow [A], supp$ 20,56 %, CF 0,35
$[A] \rightarrow [B], supp$ 20,56 %, CF 0,13
$[C] \rightarrow [A], supp$ 33,44 %, CF 0,40
$[A] \rightarrow [C], supp$ 33,44 %, CF 0,24
$[C] \rightarrow [B], supp$ 21,0 %, CF 0,21
$[B] \rightarrow [C], supp$ 21,0 %, CF 0,41
$[B, C] \rightarrow [A], supp$ 20,12 %, CF 0,81
$[C] \rightarrow [A, B], supp$ 20,12 %, CF 0,18
$[B] \rightarrow [A, C], supp$ 20,12 %, CF 0,32
$[A, C] \rightarrow [B], supp$ 20,12 %, CF 0,50
$[A] \rightarrow [B, C], supp$ 20,12 %, CF 0,12
$[A, B] \rightarrow [C], supp$ 20,12 %, CF 0,90

B

De acuerdo con lo que acabamos de exponer, nuestra definición de dependencia aproximada difusa va a ser la siguiente:

Definición 4.4.2 *Sean $X, Y \subseteq RE$ tales que $X \cap Y = \emptyset$ y $X, Y \neq \emptyset$. La **Dependencia Aproximada Difusa** $X \rightarrow Y$ sobre r se corresponde con la Regla de Asociación Difusa $I_X \Rightarrow I_Y$ en T'_r.*

Observemos cómo en la definición anterior se sigue un razonamiento análogo al expuesto en el apartado 2.3.6, cuando definíamos una dependencia aproximada apoyándonos en el concepto de regla de asociación, concepto que aquí extendemos en la forma expuesta en [Delgado et al., 2003a].

De esta misma forma, el soporte y el factor de certeza de la dependencia $X \rightarrow Y$ (o regla de asociación $I_X \Rightarrow I_Y$ en T'_r) se calculan a partir de T'_r como ya explicamos en la sección 3.3.2, y se emplean para medir la importancia y la precisión de $X \rightarrow Y$, respectivamente.

Definición 4.4.3 *El soporte de la dependencia aproximada difusa $X \rightarrow Y$ ($I_X \Rightarrow I_Y$ en T'_r), $Supp(X \rightarrow Y)$, equivale a la evaluación de la sentencia cuantificada Q de T'_r son $\tilde{\Gamma}_{I_X \cup I_Y} = Q$ de T'_r son $(\tilde{\Gamma}_{I_X} \cap \tilde{\Gamma}_{I_Y})$.*

Definición 4.4.4 *La confianza de la dependencia aproximada difusa $X \rightarrow Y$ ($I_X \Rightarrow I_Y$ en T'_r), $Conf(X \rightarrow Y)$, se corresponde con el resultado de la evaluación de la sentencia cuantificada Q de $\tilde{\Gamma}_{I_X}$ son $\tilde{\Gamma}_{I_Y}$.*

El caso del factor de certeza es más sencillo y, como ya ocurriera para las reglas de asociación difusas (apartado 3.3.2), podemos seguir calculando de la misma forma en que lo hacíamos en el caso crisp, recordando para ello la definición 2.3.5,

Definición 4.4.5 *El factor de certeza de la dependencia aproximada difusa $X \rightarrow Y$ ($I_X \Rightarrow I_Y$ en T'_r) se define como,*

$$CF(X \rightarrow Y) = CF(I_X \Rightarrow I_Y) = \frac{(Conf(I_X \Rightarrow I_Y)) - supp(I_Y)}{1 - supp(I_Y)} \qquad (4.16)$$

si $Conf(I_X \Rightarrow I_Y) > supp(I_Y)$, y

$$CF(X \rightarrow Y) = CF(I_X \Rightarrow I_Y) = \frac{(Conf(I_X \Rightarrow I_Y)) - supp(I_Y)}{supp(I_Y)} \qquad (4.17)$$

si $Conf(I_X \Rightarrow I_Y) < supp(I_Y)$, o 0 en otro caso.

A partir de la ecuación 4.13, resulta obvio deducir que $n' = |T'_r| = n^2$, conociendo que $n = |r|$. Sin embargo, como veremos más adelante, vamos a poder calcular el soporte de un ítemset I_X en tiempo $\mathcal{O}(n)$ con respecto al número de tuplas.

La dependencia aproximada difusa $X \rightarrow Y$ se cumple totalmente (cuando el factor de certeza es $CF(X \rightarrow Y) = 1$) en una relación r si y sólo si $\widetilde{ts}(I_X) \leq \widetilde{ts}(I_Y)\ \forall \widetilde{ts} \in T'_r$ (recordemos que $\widetilde{ts}(I_X) = min_{At_k \in X} \widetilde{ts}(it_{At_k})\ \forall X \subseteq RE$). Es más, puesto que las reglas de asociación difusas constituyen una generalización de las reglas de asociación "crisp", de igual forma podemos decir que las dependencias aproximadas difusas generalizan a las dependencias aproximadas clásicas.

La tabla 4.6.B nos muestra algunas de las dependencias aproximadas difusas que pueden obtenerse a partir de r, con sus correspondientes soportes (expresado en %) y factores de certeza. Podemos destacar algunos resultados obtenidos, como la dependencia $[B, C] \rightarrow [A]$, con un CF = 0.81, o como $[A, B] \rightarrow [C]$, con un CF = 0.90, resultando esta última trivial, si tenemos en cuenta que el valor en la columna C es siempre el mismo.

4.5. Comparación con el método de Wang et al.

Para mostrar un ejemplo adicional, podemos tomar prestado el conjunto de datos utilizado en [Wang et al., 1999]. La tabla 4.2 nos muestra una relación difusa en la que se representan la Categoría laboral, la Experiencia y el Salario de ocho Empleados. Las relaciones de similitud sobre los dominios de los atributos Categoria, Experiencia y Salario se muestran en la tabla 4.3. Por último, para mantener la misma semántica que aparece en nuestra definición, daremos por sentado que cada posible par $< Atributo, valor >$ tiene asociado un grado de cumplimiento igual a 1.

Aplicando la metodología expuesta sobre la relación difusa de la tabla 4.2, y tomando en consideración las relaciones de similitud existentes entre los valores de los atributos (tabla 4.3), se obtiene el conjunto de dependencias aproximadas difusas que se muestra en la tabla 4.7.

Este ejemplo nos sirve además para mostrar una primera ventaja sobre el método propuesto en [Wang et al., 1999]. En dicho trabajo, y al no establecer

en principio ninguna medida que nos informe de la bondad de las dependencias obtenidas, sólo se concluye con que la dependencia $[Cat, Exp] \rightarrow [Sal]$ se cumple en el conjunto de datos usado como ejemplo. En este sentido, creemos que podemos afirmar que nuestra metodología aporta más riqueza a los resultados. En la tabla 4.7, podemos observar que no sólo obtenemos también la dependencia $[Cat, Exp] \rightarrow [Sal]$, basándonos en el valor de su factor de certeza, 0.82. Además, encontramos otra dependencia aproximada difusa que, en principio, merecería ser tenida en cuenta. Tal es el caso de la dependencia $[Cat, Sal] \rightarrow [Exp]$, con un CF = 0.86. Por otro lado, y en añadidura, por medio de nuestra metodología mostramos el conjunto de todas las posibles dependencias aproximadas difusas que pueden extraerse de los datos del ejemplo, otorgando la posibilidad de cribar u ordenar dicho conjunto de dependencias de acuerdo al interés o importancia que puedan tener para nosotros, apoyándonos en el factor de certeza.

Tabla 4.7: Dependencias aproximadas difusas obtenidas sobre la tabla 4.2

$[Exp] \rightarrow [Cat], supp$ 64,69 %, $conf$ 74,07 %, CF 0,26
$[Cat] \rightarrow [Exp], supp$ 64,69 %, $conf$ 86,92 %, CF 0,63
$[Sal] \rightarrow [Cat], supp$ 61,87 %, $conf$ 81,53 %, CF 0,51
$[Cat] \rightarrow [Sal], supp$ 61,87 %, $conf$ 84,56 %, CF 0,59
$[Sal] \rightarrow [Exp], supp$ 64,69 %, $conf$ 87,91 %, CF 0,66
$[Exp] \rightarrow [Sal], supp$ 64,69 %, $conf$ 75,37 %, CF 0,30
$[Exp, Sal] \rightarrow [Cat], supp$ 60,62 %, $conf$ 89,12 %, CF 0,72
$[Sal] \rightarrow [Cat, Exp], supp$ 60,62 %, $conf$ 78,53 %, CF 0,45
$[Exp] \rightarrow [Cat, Sal], supp$ 60,62 %, $conf$ 69,42 %, CF 0,22
$[Cat, Sal] \rightarrow [Exp], supp$ 60,62 %, $conf$ 94,56 %, CF 0,86
$[Cat] \rightarrow [Exp, Sal], supp$ 60,62 %, $conf$ 80,61 %, CF 0,51
$[Cat, Exp] \rightarrow [Sal], supp$ 60,62 %, $conf$ 92,95 %, CF 0,82

4.6. Algunos casos particulares

Varios son los posibles escenarios en los que el concepto de Dependencia Aproximada Difusa nos puede resultar útil. En cada uno de ellos, podemos realizar una instanciación específica del concepto, en función de las relaciones de similitud que empleemos, la presencia o no de grados difusos e incluso del cuantificador empleado en el cálculo del soporte y la confianza (y, por ende, del factor de certeza) de la dependencia aproximada difusa. Algunos ejemplos de esas situaciones son los siguientes:

- Supongamos en primer lugar que estamos interesados en la búsqueda de dependencias funcionales clásicas. En tal caso, sea S_{At_k} la igualdad clásica $\forall At_k \in RE$, y sea r una relación "crisp". A esto añadamos que, en la expresión de la definición 3.3.5 (correspondiente al cómputo de la confianza), emplearemos el cuantificador difuso $\forall$ definido como $\forall(x) = 1$ si y sólo si $x = 1$ y 0 en otro caso. Entonces, lo que podamos extraer serán dependencias funcionales clásicas, y el factor de certeza de $X \rightarrow Y$ será 1 si y sólo si la dependencia funcional $X \rightarrow Y$ se cumple en r, y 0 en otro caso.

- Sea r una relación "crisp", y definamos S_{At_k} como la igualdad clásica $\forall At_k \in RE$. Si empleamos el cuantificador Q_M (definido como $Q_M(x) = x$) en la expresión de la definición 3.3.5 para el cálculo de la confianza, lo que obtendremos como resultado serán las dependencias aproximadas tal y como se definen en [Delgado et al., 2000a, Blanco et al., 2000] (ver apartado 2.3.6).

- Supongamos ahora que la cardinalidad de $dom(At_k)$ es muy alta en relación con el número de tuplas de r (el caso típico es que $dom(At_k) = \Re$, como ocurría con el atributo Sal (salario) en el ejemplo de partida). Una forma común de analizar relaciones entre At_k y otros atributos es el empleo de un conjunto de etiquetas lingüísticas $Lab(At_k)$ que reemplace al dominio original, o bien disminuir la granularidad de la descripción de At_k en general (de nuevo, podemos considerar el ejemplo introductorio, donde vimos que $Lab(Sal) = \{Alto, Medio, \ldots\}$). Si ahora queremos

encontrar dependencias en las que intervenga At_k, podemos definir una relación de semejanza como

$$S_{At_k}(x, y) = max_{L \in Lab(At_k)} min\{L(x), L(y)\} \forall x, y \in dom(At_k) \quad (4.18)$$

calculando la envolvente convexa de cara a obtener una relación de similitud si es necesario.

- Las relaciones de similitud pueden resultar útiles no sólo para resolver problemas de granularidad, sino también cuando el dominio de un atributo toma valores cuyas semánticas se solapan. Por ejemplo, consideremos el atributo *Color del pelo* y supongamos que en la base de datos hallamos valores tales como *"rubio"*, *"amarillo"*, *"claro"*, *"rojo"*, *"anaranjado"*, etc. (una posible causa estaría en que diferentes usuarios hubieran introducido información en la base de datos sin establecer un criterio a la hora de asentar el conjunto de valores que puede tomar el atributo). Si queremos asociar el color del pelo con otros atributos, podríamos estar interesados en tomar en consideración que *"amarillo"* y *"rubio"* son parecidos hasta cierto punto, entre otros casos. Esto puede llevarse a cabo definiendo una relación de similitud adecuada sobre el dominio del atributo. De esta forma, es de esperar que las dependencias en las que interviniera este atributo reflejarían en mayor medida las posibles relaciones entre el color del pelo y otras características.

En los ejemplos anteriormente citados estamos considerando que trabajamos con información de tipo "crisp", que suele ser lo habitual. En el caso de que estuviéramos ante una base de datos difusa que contuviera información imprecisa o incierta (en forma de grados difusos y relaciones de similitud), la utilidad (e incluso la necesidad) de una definición apropiada de dependencias aproximadas difusas se hace aún más patente.

Volviendo a la aplicación final y a la utilidad de las dependencias aproximadas difusas, éstas nos pueden facilitar información acerca de las posibles relaciones existentes (por medio de dependencias funcionales suavizadas en general) entre atributos de una base de datos. Este tipo de relaciones puede

verse como el resultado de un análisis exploratorio, y pueden resultarnos de suma utilidad desde el momento en que si una dependencia aproximada difusa $X \rightarrow Y$ se cumple con un alto grado de precisión, sabemos que existe un conjunto de reglas de asociación en las que intervienen los valores de X e Y que también se cumplen con una precisión bastante alta. Es decir, obtenemos un resumen de la precisión y la importancia de las relaciones entre X e Y.

Por lo tanto, el proceso de búsqueda de relaciones de interés entre atributos de una base de datos podría consistir en:

- Realizar una búsqueda de dependencias aproximadas difusas, por medio del algoritmo que describiremos en la próxima sección.
- Interpretar los resultados obtenidos y, en caso de requerir un análisis a un nivel más profundo de una o varias dependendencias aproximadas difusas presuntamente interesantes, realizar una búsqueda de reglas de asociación.
- Si no es posible extraer dependencias aproximadas difusas, también pueden usarse las reglas de asociación difusas para buscar información sobre posibles asociaciones a nivel local entre los valores de atributos.

Esta metodología, por ejemplo, ha sido la que un grupo de expertos ha empleado (aplicando dependencias aproximadas "crisp") para el análisis de bases de datos reales con información sobre suelos en [Aranda et al., 2003].

4.7. Implementación

Para que las recién definidas dependencias aproximadas difusas nos resulten un concepto útil en la práctica, deberemos implementar algoritmos eficientes capaces de obtener éstas a partir de bases de datos reales. Esto no es una tarea fácil, ya que estamos hablando de un conjunto de transacciones de tamaño n^2, siendo n el número de tuplas de una relación. Puesto que n ya puede ser un valor alto en bases de datos reales, con mayor motivo lo será n^2, razón por la cual calcular el FT-set T'_r a partir de r y a continuación trabajar con este conjunto de transacciones difusas nos puede resultar excesivamente

costoso. A esto, hemos de añadir que debemos tener en cuenta los grados difusos, las relaciones difusas de similitud y el cálculo de sentencias cuantificadas, factores éstos que también incrementarán la complejidad de la operación final.

En el campo de la minería de datos y la extracción de conocimiento, se han desarrollado diversos algoritmos para la extracción de reglas de asociación, algunos de los cuales ya citamos en la sección 2.3.4. En nuestro caso, vamos a partir de la filosofía seguida por el algoritmo original Apriori, introducido en [Agrawal y Srikant, 1995], por motivos de simplicidad principalmente, aunque las modificaciones que vamos a proponer bien podrían aplicarse a algoritmos más recientes y eficientes. Puesto que extraeremos dependencias aproximadas difusas a partir de reglas de asociación, nuestro algoritmo mantendrá la misma semántica que el definido en [Blanco et al., 2000], pero con la complejidad añadida de tener que manejar transacciones difusas. Para conseguir esto último, hemos de realizar algunos cambios, similares a los comentados en [Delgado et al., 2003a], y que se detallarán más adelante en esta misma sección.

Como ya recordamos en los capítulos preliminares, un algoritmo de extracción de reglas de asociación comprende dos fases, principalmente. En la primera, se calcula el conjunto de ítemsets frecuentes, es decir, los ítemsets más interesantes, con un soporte por encima de un cierto umbral, que llamamos *minsupp*. El proceso tiene lugar de forma iterativa, calculando en primer lugar todos los 1-ítemsets (ítemsets con un solo elemento), después los 2-ítemsets, y así sucesivamente. Cada iteración del algoritmo requiere de una pasada sobre el conjunto de transacciones, y es por ello que esta fase es la más costosa en cuanto a tiempo de ejecución.

Una vez que tenemos todos los ítemsets interesantes, un análisis sobre los mismos nos revelará las reglas de asociación con una precisión superior a un umbral determinado, denominado *minconf* (o *mincf* en nuestro caso). Este paso suele ser el mismo en todos los algoritmos de extracción de reglas, por lo que no lo volveremos a detallar aquí, remitiendo al lector al apartado B.3.

En esta sección, proporcionaremos nuestra metodología para adaptar los algoritmos ya existentes para la extracción de reglas de asociación, y en especial la primera etapa de éstos, para nuestro caso de descubrimiento de dependencias aproximadas difusas. Nuestra metodología se centra en cómo calcular

eficientemente el soporte de los atributos (nuestros ítems), teniendo además que considerar valores difusos y relaciones de similitud. Antes de pasar a describir los algoritmos en si, mostraremos un resumen de los principales aspectos de nuestro método.

4.7.1. Obtención eficiente del soporte de los ítemsets

Volvamos de nuevo al caso "crisp". Para calcular el soporte de un atributo X (ítemset I_X en T'_r), hemos de almacenar el soporte en r para cada valor $x \in dom(X)$, lo cual puede obtenerse en $\mathcal{O}(n)$. Ésta suele ser la información que se almacena por cualquier algoritmo que extraiga reglas de asociación en r (recordemos que para este caso concreto los ítems son pares $< atributo, valor >$, mientras que cuando se trata de dependencias aproximadas, los ítems son los propios atributos). A partir de esos valores de soporte, y asumiendo que como S_X tenemos la igualdad clásica, el soporte de I_X en T'_r puede obtenerse en tiempo $\mathcal{O}(K)$, siendo $K = |dom(X)|$, como

$$S(I_X) = \frac{1}{n^2} \sum_{x \in dom(X)} x^2 \tag{4.19}$$

Más concretamente, una vez que hemos obtenido el soporte para cada valor de X, solamente hemos de calcular la sumatoria de los cuadrados de dichos valores para obtener el soporte de I_X. Por ahora, no tendremos en cuenta las relaciones difusas de similitud, sino que volveremos a retomarlas más adelante. Destaquemos que el proceso anterior necesita, en el peor de los casos, esto es, cuando $K = n$, de un tiempo de ejecución de orden $\mathcal{O}(n)$.

En el caso "crisp", incluso es posible obtener el soporte de cada x y el soporte de I_X de forma simultánea, de acuerdo con el resultado expuesto con anterioridad en la proposición 2.3.5, y con el algoritmo B.4 (página 239), generado a partir de dicho resultado.

4.7.2. Tratamiento de la imprecisión en los valores

Si tenemos en cuenta la posibilidad de que existan grados difusos asociados a los valores de X en las tuplas, hemos de emplear un conjunto finito de α-cortes equidistribuidos para cada $x \in dom(X)$. El tamaño de este con-

junto dependerá del grado de precisión que se requiera, pero en cualquier caso se trata de un valor constante. Nosotros hemos optado por emplear 100 α-cortes, un valor que consideramos suficiente, tal y como se sugiere en [Delgado et al., 2003a], y que probaremos adecuado en el estudio empírico que incluimos más adelante en este mismo capítulo. Para ello, hemos de redondear o truncar los grados difusos. Durante la fase de exploración de las tuplas de r, lo que almacenaremos será el número de veces en que un cierto valor x aparece con un cierto grado. Cada valor x tiene asociado un vector unidimensional que denominamos $N(X, x)$, con el propósito de almacenar los resultados anteriores. En el cálculo de $N(X, x)$ se emplea un tiempo $\mathcal{O}(n)$. A partir de los vectores $N(X, x)$ es posible obtener un vector similar pero para I_X, al que llamamos V_X, en tiempo $\mathcal{O}(K)$. Cada posición de este último vector almacena el número de transacciones en T'_r en las que I_X aparece con un cierto grado.

El soporte de cada x e I_X puede obtenerse (como resultado de la evaluación de las correspondientes sentencias cuantificadas) a partir de sus correspondientes vectores en tiempo $\mathcal{O}(1)$ por medio del algoritmo B.7 (página 242). Este algoritmo es una modificación del algoritmo B.6, definido anteriormente. Tal modificación es necesaria debido a la particularidad de trabajar con relaciones de similitud, lo cual nos obliga a tratar el vector V_X, asociado a cada ítemset I_X, como una lista de α-cortes. La modificación consiste en eliminar los acumuladores que se usaban previamente, ya que, como veremos a continuación, dicha acumulación se realiza en un paso anterior, involucrando las clases de equivalencia definidas por las relaciones de similitud. Aún así, una vez más, la complejidad final en tiempo sigue siendo $\mathcal{O}(n)$. Por otro lado, los requisitos en espacio son de un entero largo para cada $x \in dom(X)$ (como ocurre en cualquier algoritmo de extracción de reglas de asociación), pero multiplicado por un valor constante (que será el número de α-cortes que consideremos).

4.7.3. Operaciones con relaciones de similitud

Por último, está el tema de las relaciones de similitud. Los α-cortes de las relaciones difusas de similitud (recordemos, max-min transitivas) nos proporcionan un conjunto de clases de equivalencia "crisp" sobre $dom(X)$. Esta información ha de ser tenida en consideración durante el cómputo de V_X a par-

tir de su conjunto asociado de vectores $N(X, x)$ para los valores $x \in dom(X)$. La idea que subyace bajo esto es que los vectores $N(X, x)$ de aquellos valores $x \in dom(X)$ que son iguales a un cierto nivel de acuerdo con S_X se suman para formar un único vector a dicho nivel. Lo que queremos decir con esto es que si dos valores $x_1, x_2 \in dom(X)$ son similares en un grado β (es decir, $S_X(x_1, x_2) = \beta$), entonces a partir de unos ciertos niveles $\alpha \leq \beta$, hemos de considerarlos como el mismo valor. De esta manera, aplicaremos la ecuación 4.19 en cada nivel sobre las clases de equivalencia inducidas por S_X en ese nivel. Esta información puede incorporarse al proceso del cálculo de V_X, a partir del cual obtendremos el soporte de I_X, sin que por ello se incremente la complejidad en cuanto a tiempo de ejecución, si nos atenemos a lo mostrado en el algoritmo B.10, que podemos hallar en la página 244. En particular, las clases de equivalencia pueden obtenerse antes de que comience el proceso de extracción, siguiendo el algoritmo B.9 (página 244).

Volviendo a las relaciones de similitud como tales, imponen una fuerte restricción con la max-min transitividad, de forma que puede que no cumplan todas las relaciones difusas definidas sobre un dominio. Para garantizar esta condición, sin embargo, es posible calcular la clausura transitiva de una relación de semejanza, con lo que obtendremos la relación de similitud deseada. En [Bezdek y Harris, 1978] se comentan tres posibles algoritmos para obtener la clausura transitiva de una matriz simétrica asociada a una relación difusa, a saber:

- Por medio de la composición $(\vee, \wedge)$ iterativa, como se describe en los trabajos [Zadeh, 1971] y [Tamura et al., 1971].
- Un algoritmo de exploración columna-fila, como el que podemos encontrar en [Kandel y Yelowitz, 1974].
- El procedimiento del mínimo árbol de expansión de Prim, descrito en [Dunn, 1974].

De acuerdo con la representación elegida para nuestras relaciones de similitud, el algoritmo propuesto en [Kandel y Yelowitz, 1974] nos resulta el más sencillo de llevar a la práctica, por lo que será el que elijamos. El funcionamien-

to del mismo se describe en el algoritmo B.8, cuyo desarrollo se muestra en la página 243.

4.7.4. Dependencias sobre valores no atómicos

Tal y como la hemos expresado, nuestra definición se reduce al caso en el que una celda de nuestra tabla tuviera la estructura $< \tilde{t}(At_k), \mu_{\tilde{t}}(At_k) >$, indicando con ello que el grado en que el atributo At_k toma el valor $\tilde{t}(At_k)$ es $\mu_{\tilde{t}}(At_k)$. Pero lo más corriente en problemas reales, y ahí radica la riqueza en cuanto a expresividad que nos proporciona la extensión al caso difuso, será que un mismo atributo pueda tomar varios valores simultáneamente, de forma que éstos se encuentren solapados en un cierto grado (p.e., el atributo *Color del pelo* podría tomar los valores (*rubio*/0.8, *castaño*/0.3) para un mismo individuo).

Una primera solución para afrontar este problema es la de considerar cada par $< valor, grado >$ como perteneciente a un atributo (o columna) distinto, y aplicar tal cual el algoritmo de minería de datos. No obstante, debemos tener en cuenta una puntualización, la de no permitir reglas en las que un mismo atributo (aunque tomando distintos valores) apareciese al mismo tiempo en el antecedente y en el consecuente. Para ello bastaría con modificar el procedimiento básico de generación de las reglas (o dependencias) (algoritmo B.3, página 238), que ya fue descrito en el apartado 2.3.4.5, para incluir dicha restricción. No obstante, este problema se presenta más complejo cuando intervienen relaciones de similitud entre distribuciones de posibilidad (y no ya sólo valores de atributos), por lo que este aspecto será objeto de un estudio más profundo en futuras ampliaciones de este trabajo.

4.7.5. Algoritmo

Por último, el algoritmo B.11, mostrado en la página 245, es la adaptación, de acuerdo con la metodología expuesta con anterioridad en este mismo capítulo, del algoritmo básico de obtención de ítemsets frecuentes para el caso concreto de extracción de dependencias aproximadas difusas. A modo de resumen, recordemos que las modificaciones previamente expuestas no incrementan la complejidad del algoritmo de extracción de reglas de asociación, sea éste cual sea, aunque el tiempo y el espacio sí se ven modificados, multiplicándolos por

una constante que depende principalmente del número de α-cortes considerados.

Entrando en más detalle acerca del funcionamiento del algoritmo B.11, la función $\rho(z, k)$ toma como argumento un valor real z y lo asocia al valor más cercano en el conjunto finito de niveles que usamos para representar los grados difusos. Los ítemsets se van obteniendo de forma ordenada, de acuerdo a su tamaño. La variable l nos muestra el tamaño actual que se está considerando, al tiempo que actúa como un contador de la etapa actual del algoritmo. El conjunto L_l almacena los l-ítemsets que se están analizando y, al final de cada iteración, se queda con aquéllos que han resultado ser frecuentes. Antes, y debido a la consideración de las relaciones de similitud entre valores, hemos de calcular convenientemente el soporte para cada ítemset, para lo cual empleamos en primer lugar el procedimiento descrito en el algoritmo B.10 y, en segundo lugar, el algoritmo B.7 para evaluar la sentencia cuantificada correspondiente.

El procedimiento $CrearNivel(i, L)$ genera el siguiente conjunto de i-ítemsets para ser analizados, de forma que cada subconjunto de $i - 1$ ítems sea frecuente (o lo que es lo mismo, que ya estuviera en L_{i-1}), junto con los contadores asociados. Dado que cada subconjunto de un ítemset frecuente es también un ítemset frecuente (como ya se veía en el algoritmo original presentado en [Agrawal y Srikant, 1995]), con este procedimiento evitamos el análisis de ítemsets que no verifiquen esta propiedad, ahorrando con ello tiempo y espacio.

4.8. Aplicaciones a un caso real sobre datos médicos

En estos últimos apartados del presente capítulo, aplicaremos nuestra metodología sobre dos problemas concretos, discutiendo cuando sea posible los resultados que se obtengan.

Para poner en práctica los conceptos anteriormente definidos, tenemos a nuestra disposición unos datos sobre arteriosclerosis, obtenidos como resultados del estudio longitudinal sobre factores de riesgo asociados a dicha enfermedad, **STULONG**. Dicho estudio se realizó en el 2º Departamento de Medicina, 1ª Facultad de Medicina de la Universidad de Charles y el Hospital Universitario de Charles, U nemocnice 2, Praga 2 (director, Prof. M. Ascher-

mann, MD, SDr, FESC), bajo la supervisión del Prof. F. Boudík, MD, ScD, con la colaboración de M. Tomecková, MD, PhD y el Prof. Asoc. J. Bultas, MD, PhD. Los datos fueron transferidos a un soporte electrónico por el Centro Europeo de Informática Médica, Estadística y Epidemiología de la Universidad de Charles y la Academia de las Ciencias (director, Prof., Dr., J. Zvarova, SDr). Los datos se encuentran accesibles a través de Internet en la dirección `http://euromise.vse.cz/challenge2003/`. En la actualidad, el análisis de dichos datos se encuentra subvencionado por la beca CR Nr LN 00B 107 del Ministerio de Educación de la República Checa.

Desde aquí queremos expresar nuestro más sincero agradecimiento a las entidades anteriormente mencionadas por permitirnos utilizar dicho conjunto de datos para nuestra experimentación.

4.8.1. Conjuntos de datos

Los conjuntos de datos facilitados por STULONG consisten en cuatro tablas como las que se describen a continuación:

- **Entry**. Contiene información sobre 1417 sujetos, examinados durante su ingreso. Para cada uno de ellos, se recoge un total de 244 atributos, de los cuales 64 están codificados, son categorías asociadas a intervalos sobre diversas variables o bien resultados de transformaciones sobre el resto de atributos.

- **Control**. A lo largo de 20 años, se han monitorizado distintos factores de riesgo o evidencias clínicas de arteriosclerosis en pacientes. Para cada uno de ellos se recoge un total de 66 atributos, los cuales o bien están codificados o se corresponden con intervalos sobre dominios numéricos.

- **Letter**. A través de un cuestionario por correo, se obtuvo información adicional sobre el estado de salud de 403 sujetos. La información resultante se recoge en un total de 62 atributos.

- **Death**. Por último, esta tabla recoge información sobre el fallecimiento de 389 pacientes, descrita por medio de 5 atributos.

Centrándonos en la tabla de ingresos (Entry), encontramos una separación semántica entre los atributos, destacando los agrupamientos que se recogen en las tablas siguientes, contenidas en el apéndice C:

- Tabla C.1. Atributos relativos a factores sociales.
- Tabla C.2. Atributos alusivos al ejercicio físico, tanto dentro como fuera del trabajo.
- Tabla C.3. Información sobre el consumo de tabaco.
- Tabla C.4. Información sobre el consumo de alcohol.
- Tabla C.5. Atributos con información sobre análisis físicos (peso, altura, etc.).
- Tabla C.6. Información sobre análisis químicos realizados al paciente (nivel de colesterol y triglicéridos).

La gran mayoría de los atributos considerados son categóricos, con la excepción de algunos como el peso y la altura. En medicina suele tenerse en cuenta un valor denominado Índice de Masa Corporal (BMI), que se obtiene como un campo calculado a partir de los valores del peso y la altura, de la forma siguiente:

$$BMI = (peso\ en\ kg./(altura\ en\ m)^2)$$

De acuerdo también con las fuentes de datos, se considera que una persona tiene sobrepeso cuando su BMI supera o iguala al valor 25. En otro caso, se considera que la persona es delgada.

4.8.2. Preprocesamiento de los datos

Los datos suministrados por el proyecto STULONG consisten en matrices de datos en bruto sobre las que, antes de proceder a un análisis con garantías de éxito, ha de realizarse alguna transformación sobre varios de sus atributos. Cuando hablábamos de las reglas de asociación generalizadas y cuantitativas (apartados 2.3.5.1 y 2.3.5.2, respectivamente), discutíamos cómo uno de los

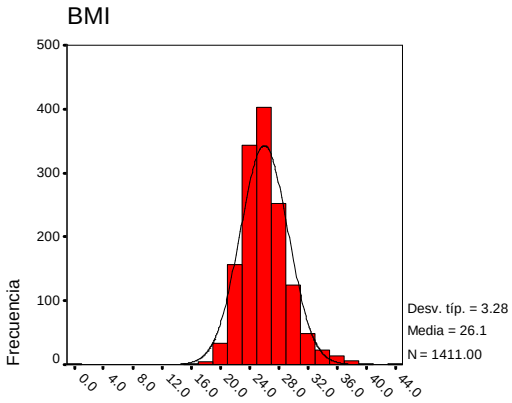


Figura 4.1: Histograma de la medida BMI

principales problemas a los que nos enfrentamos cuando tratamos con atributos sobre dominios numéricos es el del aumento de la granularidad y dispersión en los datos. Esto, llevado al caso de la extracción de reglas de asociación o dependencias aproximadas, provoca la aparición del problema que conocemos con el nombre de "soporte mínimo" ("Minsupp").

Por otro lado, los resultados que obtengamos pueden ser descritos más adecuadamente si previamente se ha definido un conjunto de etiquetas lingüísticas sobre el dominio de los atributos problemáticos. En el caso que nos ocupa, y siguiendo la anterior propuesta, definimos un conjunto de tres intervalos equidistribuidos (que podrían corresponderse perfectamente con el conjunto de etiquetas {*"bajo", "medio", "alto"*}) sobre los atributos $SYST1$ (Presión sanguínea I sistólica), $DIAST1$ (Presión sanguínea I diastólica), $SYST2$ (Presión sanguínea II sistólica), $DIAST2$ (Presión sanguínea II diastólica), $TRIC$ (pliegues dérmicos sobre músculo tríceps), $SUBSC$ (pliegues dérmicos sobre músculo subescapular), $CHLST$ (Colesterol) y $TRIGL$ (Triglicéridos) dentro de la tabla *Entry*, y cuya descripción podemos ver en las subtablas que aparecen en el apéndice C. De igual forma, siguiendo la idea propuesta en el anterior apartado, el índice de masa corporal (BMI) fue categorizado en dos intervalos, indicando si el paciente sufre o no de sobrepeso.

Para proporcionar más riqueza al modelo, los intervalos originales fueron suavizados para ser transformados en conjuntos difusos. Nuestro interés se

centra en evitar que los intervalos se solapen, debido a que ello supondría el que, a partir de un cierto valor de corte, los valores serían indistinguibles y de poca utilidad para nuestro propósito. Por esta razón, estableceremos el punto de corte en 0 para cada par de intervalos definidos sobre el mismo dominio.

Esto a su vez nos genera otro problema, y es que los valores cercanos a los límites de los intervalos habrán de ser obviados. Nos interesa, por tanto, que el número de valores descartados sea el mínimo. Para ello, con ayuda del paquete estadístico SPSS, estudiamos la distribución de los valores en cada dominio. Asumiendo que dicha distribución es normal, y calculando los percentiles 47 y 53, dejamos fuera entre un 5 % y un 7 %, aproximadamente, de los valores del dominio en torno al centro de la distribución. Tomando esa fracción como amplitud de la pendiente del intervalo, reducimos en lo posible el número de valores perdidos.

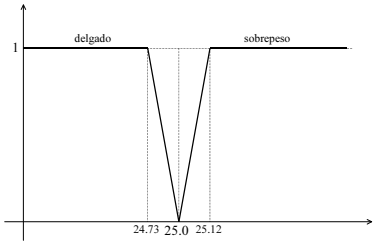


Figura 4.2: Etiquetas sobre el atributo BMI

Para estudiar un ejemplo concreto de este proceso, nos centraremos en el atributo BMI. El primer paso fue estudiar la distribución de los valores de dicho atributo, cuyo histograma mostramos en la figura 4.1. Dado que nos interesa establecer el punto de separación entre etiquetas lingüísticas en el valor 25, tomamos dicho valor como mediana y calculamos los percentiles 47 y 53 en torno a él.

Con esos valores ya podemos construir las distribuciones de posibilidad que asociaremos a las etiquetas “delgado” y “sobrepeso”, que definimos sobre el atributo. La figura 4.2 muestra dichas distribuciones sobre el dominio del atributo BMI.

En cuanto a los atributos de tipo categórico que aparecen en la tabla *Entry*, el proceso a seguir con ellos fue el de definir una relación de simili-

tud difusa entre sus valores. Las relaciones de similitud resultan especialmente adecuadas para la descripción de información analógica sobre dominios discretos, como una alternativa a las distribuciones de posibilidad. Dichas relaciones suelen construirse a partir de una interpretación semántica de los valores de los atributos, y ése fue el criterio que seguimos, y que hemos considerado interesante.

Las relaciones de similitud definidas sobre los atributos de interés se muestran en la tabla C.7. Dado que las relaciones para los atributos $CERVDI$ (consumo diario de cerveza), $VINODI$ (consumo diario de vino) y $LICDI$ (consumo diario de licores) resultaron ser las mismas, hemos omitido las dos últimas tablas para evitar una innecesaria redundancia.

Las descripciones de los grupos de atributos, junto con las tablas que nos muestran las relaciones de similitud definidas, pueden encontrarse en el apéndice C.

4.8.3. Resumen de las cuestiones planteadas y resultados

Junto con la descripción de la base de datos, hemos incluido una descripción más detallada de los resultados obtenidos mediante nuestra metodología en el apéndice C. Para facilitar la lectura, en este apartado nos limitaremos a mostrar las tablas con las dependencias aproximadas difusas entre atributos de varios grupos semánticos.

La lectura de dichas tablas debe hacerse como sigue: cada celda contiene un par de valores entre 0 y 1. El primero de ellos corresponde al factor de certeza de la dependencia con el atributo de la fila como antecedente y el atributo de la columna como consecuente. El segundo valor de la celda corresponde al factor de certeza de la dependencia recíproca.

Hemos restringido el análisis a la tabla *Entry*, que consta de 1417 casos, sobre los que consideramos 32 atributos. Hemos omitido aquellos casos en los que el soporte no superaba el umbral impuesto de 0.1, por considerar que no nos puedan resultar interesantes. Asimismo, destacamos en negrita aquellos valores del factor de certeza que puedan ser más interesantes.

Tabla 4.8: Dependencias entre factores sociales y actividad física

	AESTUD	STATUS	EDUCAL	RESLAB
FISENT		**0.67**/0.14	0.24/0.37	0.25/0.28
FISTRT	0.14/**0.47**	**0.58**/0.28	0.14/**0.49**	0.18/**0.47**
TRASNT	0.20/0.32	**0.64**/0.14	0.19/0.32	0.26/0.32
TTRANST	0.17/**0.47**	**0.57**/0.22	0.16/**0.46**	0.21/**0.44**

Tabla 4.9: Dependencias entre factores sociales y tabaco

	AESTUD	STATUS	EDUCAL	RESLAB
INTENSF		**0.68**/0.07		
TFUM		**0.64**/0.11		0.26/0.25
TSINFUM	0.10/**0.64**	**0.42**/**0.39**	0.09/**0.65**	0.13/**0.64**

4.8.3.1. Relaciones en las que intervienen factores sociales

Las tablas 4.8, 4.9, 4.10 y 4.11 contienen, respectivamente, las dependencias aproximadas difusas que pudimos hallar entre los factores sociales y la actividad física en el trabajo, los hábitos de consumo de tábaco y alcohol, y algunas características físicas.

Lo primero que puede apreciarse es la presencia de un alto valor del factor de certeza para aquellas dependencias con el atributo $STATUS$ en el consecuente. Ello es debido a que dicho atributo tiene un soporte muy alto, como consecuencia de que la distribución de valores asociada está muy desequilibrada (de 1417 sujetos, más de 1200 estaban casados). Por esta razón, las dependencias en las que aparece dicho atributo en el consecuente pueden llevar a confusión.

Remitimos al lector al apéndice C donde encontrará un desglose más detallado de los resultados obtenidos relativos a estas cuestiones.

4.8.3.2. Relaciones en las que interviene la actividad física

Hemos incluido en las tablas 4.12, 4.13, 4.14 y 4.15 las dependencias aproximadas difusas que se obtuvieron involucrando la actividad física en el trabajo

Tabla 4.10: Dependencias entre factores sociales y alcohol

	AESTUD	STATUS	EDUCAL	RESLAB
ALCOHOL	0.21/0.35	**0.63**/0.15	0.19/0.34	0.24/0.31
CERV10	0.16/**0.43**	**0.58**/0.21	0.16/**0.43**	0.21/**0.41**
CERV12	0.10/**0.62**	**0.47**/**0.39**	0.10/**0.62**	0.13/**0.61**
VINO	0.16/**0.43**	**0.58**/0.21	0.16/**0.44**	0.21/**0.41**
LICOR	0.16/**0.43**	**0.58**/0.21	0.16/**0.43**	0.20/**0.41**
CERVDI	0.21/0.33	**0.65**/0.14	0.20/0.32	0.24/0.29
VINODI	0.20/0.33	**0.64**/0.15	0.19/0.33	0.24/0.31
LICDI	0.20/0.31	**0.64**/0.14	0.19/0.30	0.25/0.29

Tabla 4.11: Dependencias entre factores sociales y características físicas

	AESTUD	STATUS	EDUCAL	RESLAB
BMI	0.16/**0.44**	**0.58**/0.23	0.15/**0.45**	0.20/**0.42**
SYST1		**0.65**/0.12		0.25/0.26
DIAST1	0.19/0.32	**0.63**/0.14	0.19/0.32	0.24/0.30
SYST2		**0.65**/0.12		0.25/0.25
DIAST2	0.19/0.33	**0.63**/0.15	0.18/0.33	0.23/0.30

con los hábitos de consumo de tábaco y alcohol, algunas características físicas (BMI y presión sanguínea) y con el nivel de colesterol.

Hemos de destacar que estas dependencias se calcularon sobre el conjunto total de sujetos, por lo que de nuevo hemos de remitir al apéndice C para encontrar una descripción más detallada de los resultados obtenidos.

4.8.3.3. Relaciones en las que interviene el consumo de alcohol

Para finalizar, mostraremos un último ejemplo de dependencias aproximadas difusas sobre el conjunto de datos STULONG. Las tablas 4.16 y 4.17 recogen, respectivamente, las dependencias entre consumo de alcohol y caracteres físicos y entre consumo de alcohol y de tabaco. Como se detalla en el apéndice C, el atributo $CERV7$ (consumo de cerveza de 7^o) no fue tenido

Tabla 4.12: Dependencias entre actividad física y tabaco

	FISENT	FISTRT	TRASNT	TTRANST
INTENSF		**0.50**/0.11		**0.45**/0.13
TFUM	0.27/0.24	**0.47**/0.18	0.30/0.24	**0.42**/0.19
TSINFUM	0.13/**0.62**	0.26/**0.51**	0.15/**0.51**	0.23/**0.55**

Tabla 4.13: Dependencias entre actividad física y alcohol

	FISENT	FISTRT	TRASNT	TTRANST
ALCOHOL	0.27/0.31	**0.46**/0.23	0.29/0.30	**0.41**/0.25
CERV10	0.22/**0.39**	**0.40**/0.30	0.24/**0.39**	0.35/0.33
CERV12	0.14/**0.59**	0.29/**0.50**	0.16/**0.59**	0.23/**0.50**
VINO	0.22/**0.40**	**0.40**/0.31	0.24/**0.39**	0.35/0.33
LICOR	0.22/**0.39**	**0.39**/0.30	0.24/**0.38**	0.35/0.33
CERVDI	0.27/0.29	**0.46**/0.21	0.30/0.29	**0.42**/0.24
VINODI	0.27/0.31	**0.46**/0.23	0.28/0.30	**0.41**/0.24
LICDI	0.27/0.28	**0.46**/0.21	0.29/0.27	**0.41**/0.23

en cuenta, debido a que su alto soporte (cercano al 100 %) provocaba el que las dependencias en las que se veía involucrado como consecuente tuvieran un engañoso valor de su factor de certeza muy cercano a 1.

4.8.4. Discusión de los resultados

Como parte de la experimentación realizada para esta memoria, hemos analizado los datos proporcionados por el proyecto STULONG con el objeto de extraer dependencias aproximadas difusas. Para ello, hemos transformado el dominio de algunos atributos, usando conjuntos difusos, en un intento de proporcionar información más fácilmente asimilable. En los anteriores subapartados, hemos tratado de responder a una serie de cuestiones analíticas propuestas sobre los datos en bruto. Los resultados obtenidos, en forma de un conjunto de dependencias aproximadas difusas, podrían ser aplicados a la descripción de los datos facilitados o, más adelante, en la predicción de val-

Tabla 4.14: Dependencias entre actividad física y características físicas

	FISENT	FISTRT	TRASNT	TTRANST
BMI	0.21/**0.41**	**0.39**/0.32	0.23/**0.40**	0.34/0.34
SYST1	0.27/0.26	**0.46**/0.19	0.29/0.25	**0.42**/0.21
DIAST1	0.25/0.29	**0.44**/0.22	0.28/0.29	**0.39**/0.23
SYST2	0.27/0.25	**0.47**/0.18	0.29/0.24	**0.42**/0.20
DIAST2	0.25/0.29	**0.45**/0.22	0.27/0.29	**0.39**/0.24

Tabla 4.15: Dependencias entre actividad física y colesterol

	FISENT	FISTRT	TRASNT	TTRANST
CHLST	0.28/0.24	**0.47**/0.17	0.30/0.23	**0.42**/0.19
TRIGL		**0.49**/0.13		**0.45**/0.14

ores. No obstante, aunque los resultados parecen ser interesantes hasta donde nosotros sabemos, apoyándonos en el valor del factor de certeza, sería deseable la intervención de expertos médicos en la materia con objeto de dar una interpretación más completa. Esto último será posible gracias al hecho de que los resultados experimentales serán presentados en un Discovery Challenge, que se celebrará próximamente. Este evento se desarrollará como una sesión especial, dedicada a la extracción de conocimiento, de la edición del 2003 del ECML/PKDD, patrocinado por entidades como KD-Net y la editorial Kluwer.

4.9. Caso real sobre datos agronómicos

A continuación, mostraremos los resultados empíricos de aplicar nuestra metodología sobre una base de datos con información sobre la composición y estructura de suelos dedicados al cultivo del olivar. El objetivo que buscamos es el de hallar una relación entre los componentes y la estructura del suelo y el color de éste.

El color es una característica muy destacada de los suelos, determinada con cierta facilidad por expertos y no expertos y que permite la estimación

Tabla 4.16: Dependencias entre alcohol y características físicas

	BMI	SYST1	DIAST1	SYST2	DIAST2
ALCOHOL	**0.40**/0.24	0.25/0.30	0.28/0.29	0.24/0.31	0.28/0.29
CERV10	0.35/0.33	0.21/**0.39**	**0.38**/0.24	0.20/**0.40**	0.24/**0.38**
CERV12	0.25/**0.52**	0.14/**0.60**	0.16/**0.59**	0.13/**0.60**	0.17/**0.58**
VINO	0.35/0.32	0.21/**0.40**	0.24/**0.38**	0.20/**0.40**	0.24/**0.38**
LICOR	0.35/0.33	0.21/**0.40**	0.24/**0.38**	0.20/**0.40**	0.24/**0.38**
CERVDI	**0.41**/0.23	0.25/0.28	0.29/0.27	0.25/0.29	0.29/0.27
VINODI	**0.40**/0.24	0.25/0.30	0.28/0.28	0.24/0.30	0.28/0.28
LICDI	**0.41**/0.22	0.25/0.28	0.29/0.27	0.24/0.28	0.29/0.27

cualitativa de los materiales que componen los horizontes del suelo y de los procesos que son o han sido operativos en el mismo [Bigham y Ciolkosz, 1993]. Numerosos autores han emprendido el estudio de las relaciones del color con las propiedades del suelo [Torrent et al., 1980, Schwertmann, 1993, Schulze et al., 1993]. En [Sánchez-Marañón et al., 1997], se profundiza en los denominados "Suelos rojos mediterráneos", típicos del entorno climático mediterráneo, utilizando herramientas estadísticas para sugerir y contrastar un determinado número de hipótesis que relacionan algunas propiedades del suelo con el color del mismo. Nuestro objetivo en este apartado será el de emplear técnicas de minería de datos, haciendo especial énfasis en el tratamiento de la imprecisión sobre los datos, y comparar los análisis globales con los resultados estadísticos, contrastando la eficacia y operatividad del método.

4.9.1. Fuentes bibliográficas empleadas

La base de datos que consideraremos incluye información edáfica de tres mesoambientes del sur y sureste de la Península Ibérica bajo clima mediterráneo: Sierra de Gádor, Sierra Nevada y Sureste (que incluye parte de las provincias de Murcia y Almería). Los datos de la Sierra de Gádor se han extraído de [Oyonarte, 1990], los datos para el Sureste, del mapa de suelos escala 1:100000 [LUCDEME, 1981, LUCDEME, 1987a, LUCDEME, 1987b] y los datos de Sierra Nevada, de [Sánchez-Marañón, 1992]. La base de datos comprende un total

Tabla 4.17: Dependencias entre consumo de alcohol y tabaco

	INTENSF	TFUM	TSINFUM
ALCOHOL		0.23/0.30	**0.61**/0.15
CERV10	0.13/**0.44**	0.20/**0.40**	**0.56**/0.22
CERV12	0.08/**0.65**	0.13/**0.60**	**0.44**/**0.40**
VINO	0.13/**0.44**	0.20/**0.40**	**0.56**/0.22
LICOR	0.13/**0.44**	0.20/**0.40**	**0.56**/0.22
CERVDI		0.23/0.28	**0.61**/0.14
VINODI		0.23/0.30	**0.61**/0.15
LICDI		0.24/0.28	**0.62**/0.14

de 36 variables (de las cuales sólo 34 intervienen en el análisis) y 541 casos. Las variables numéricas continuas se categorizaron mediante un algoritmo de discretización basado en las k-medias. El experto consideró adecuado usar tres clases (asociables al conjunto {*"alto", "medio", "bajo"*}) por variable. Describiremos el preprocesamiento de los datos más adelante.

El color de suelo se puede cuantificar utilizando varios sistemas de color. Entre ellos, destaca la escala Munsell [Munsell, 1954, Soil Survey Staff, 1975], basada en tres parámetros:

- **Hue.** Se relaciona con la longitud de onda dominante en la radiación reflejada.
- **Value.** Expresa la proporción de luz reflejada.
- **Chroma.** Expresa la pureza relativa o intensidad cromática del color.

La medida del color se estandariza llevando al suelo a unos estados de humedad definidos por lo que cada variable de color se expresa en seco y en húmedo. Basándonos en la matriz de correlaciones de [Sánchez-Marañón, 1992], se seleccionaron aquellas propiedades que están más correlacionadas (positiva o negativamente) con el Hue, Value y Chroma, respectivamente: % de Arcilla, % de Carbono Orgánico (CO) y % de Arena, siendo la relación entre estas propiedades y el color la que presenta más interés.

Finalmente, la tabla C.17 muestra los atributos que fueron tenidos en cuenta en el análisis, así como su tipo. Podemos agrupar el conjunto de atributos en relación a su semántica, tal y como hacemos en la tabla C.18. No obstante, antes de pasar al proceso de extracción de conocimiento, hemos de realizar un análisis previo sobre el conjunto de datos. En el siguiente apartado describiremos en más detalle todo este proceso.

4.9.2. Preprocesamiento de los datos

Antes de proceder a la etapa propiamente dicha de extracción de dependencias aproximadas difusas, hemos de preprocesar la información suministrada por los expertos edafólogos que nos asistieron durante la experimentación. Un análisis exploratorio previo de dichos datos por medio de dependencias aproximadas y reglas de asociación, clásicas en ambos casos, reveló muy poca información en cuanto a posibles relaciones existentes entre atributos, debiendo tratar la información a un nivel más local para buscar posibles asociaciones entre los valores de atributos.

Es por ello que, en un intento de obtener mejores resultados, optamos por suavizar las relaciones entre los valores de los atributos mostrados en la tabla C.17, introduciendo algunos factores de incertidumbre. En el caso de los atributos de carácter categórico, establecimos una relación de similitud entre los valores del dominio de cada uno de ellos. Por otro lado, para eliminar granularidad en los datos numéricos, particionamos los dominios de estos atributos por medio de un algoritmo de discretización basado en las k-medias como los comentados en [Hussain et al., 1999]. Aconsejados por los expertos, optamos por establecer sobre cada dominio numérico una distribución de posibilidad asociada al conjunto de etiquetas lingüísticas { *"alta", "media", "baja"*}. Posteriormente, los intervalos obtenidos fueron transformados en distribuciones de tipo trapezoidal, en un proceso análogo al detallado en el anterior apartado dedicado a la experimentación, por lo que no volveremos a describirlo.

Una primera ventaja de esta etapa de preprocesamiento fue la de poder contar en todo momento con la asistencia y el asesoramiento de expertos en el área de conocimiento, algo bastante necesario no sólo en la fase de interpretación de resultados, sino también en esta primera fase exploratoria, sobre

todo a la hora de establecer una semántica en la definición de relaciones de similitud y etiquetas lingüísticas.

Para no entorpecer la lectura del presente capítulo, hemos desplazado las tablas de descripción de los datos experimentales, así como las tablas que nos muestran las relaciones de similitud y los conjuntos de etiquetas lingüísticas definidos, al apéndice C.

4.9.3. Resultados e interpretación

En este apartado, nos apoyaremos en la asistencia de los expertos humanos para dar una interpretación de los resultados y discutir la bondad de los mismos. En primer lugar, como una fase previa de exploración, aplicamos a los datos un algoritmo de extracción de dependencias aproximadas clásicas (el algoritmo B.5 que ya apareciera en el apartado 2.3.6). Las mejores dependencias aproximadas "crisp" que se obtuvieron, basándonos en el valor del factor de certeza (aplicando un umbral de 0.7 sobre éste), son las que se muestran a continuación,

$[HUE\_SECO] \rightarrow [HUE\_HUME]$, *supp* $17,35\,\%, CF\ 0,91$
$[HUE\_HUME] \rightarrow [HUE\_SECO]$, *supp* $17,35\,\%, CF\ 0,88$
$[ALTITUD] \rightarrow [PMEDIA]$, *supp* $35,51\,\%, CF\ 0,80$
$[TMEDIA] \rightarrow [PMEDIA]$, *supp* $31,87\,\%, CF\ 0,78$
$[FE] \rightarrow [COD\_ECOL]$, *supp* $28,62\,\%, CF\ 0,76$
$[PMEDIA] \rightarrow [ALTITUD]$, *supp* $35,51\,\%, CF\ 0,75$
$[COD\_ECOL] \rightarrow [PMEDIA]$, *supp* $31,16\,\%, CF\ 0,71$

De las dependencias mostradas, la primera resulta trivial desde el punto de vista edáfico, ya que simplemente asocia los dos estados de humedad (en seco y en húmedo) en los que se mide el parámetro de color, Hue. Este parámetro, como indicábamos más arriba, nos proporciona información sobre la longitud de onda de la radiación reflejada por la muestra del suelo. El Hue, frente a otros parámetros de color como el Value (luminosidad) o el Chroma (intensidad), es difícilmente alterable al pasar el suelo del estado húmedo al seco. Por esta razón, es lógico que aparezcan tan relacionados, cosa que no ocurre con los parámetros Value o Chroma.

La segunda dependencia nos muestra, en boca de los expertos, la estrecha relación entre los tres parámetros climáticos implementados en la base de datos. Siguiendo el patrón climático regional, a una mayor altura corresponde una mayor precipitación, y viceversa, asociación que aparece como dependencia aproximada en la anterior lista. También se revela la asociación entre la temperatura y la precipitación. En la zona geográfica estudiada, a una mayor temperatura corresponde invariablemente una menor precipitación, prácticamente sin excepciones microclimáticas.

Otra de las dependencias aproximadas nos indica la existencia de una asociación entre la precipitación y el mesoambiente (atributo COD_ECOL). Esta relación resulta bastante lógica, ya que se ha recopilado información sobre tres mesoambientes: dos montañosos (Sierra Nevada y Sierra de Gádor) y las zonas bajas del Sureste español. Basándonos en las dependencias ya comentadas, y conociendo el estricto gradiente de altitud Sureste-Sierra de Gádor-Sierra Nevada, era de esperar que se presentase esta asociación. Por su parte, el contenido de Hierro libre (atributo FE) parece asociarse al grado de evolución de los suelos, por lo que esta interesante dependencia nos está indicando que existe una relación general evolutiva entre los mesoambientes.

A pesar de todo, hay que resaltar que, mediante los métodos clásicos, no es posible obtener dependencias a nivel global (apoyándonos de nuevo en el umbral de CF utilizado) entre algún parámetro del color (Hue, Value, Chroma) y algún otro tipo de variable descriptiva del suelo, aspecto éste de especial interés para los expertos.

Sin embargo, con la nueva metodología es posible obtener hasta 34 dependencias aproximadas difusas con un factor de certeza superior al umbral considerado de 0.7, lo que supone multiplicar casi por 6 la posibilidad de obtener información potencialmente útil de la base de datos. De los resultados obtenidos, seleccionaremos a continuación aquellas dependencias que muestran asociaciones entre parámetros de color y otras variables de suelos,

$[CO] \rightarrow [VALUE\_HU]$, *supp* 27,6 %, CF 0,75
$[CEC] \rightarrow [VALUE\_HU]$, *supp* 26,31 %, CF 0,7
$[ARCILLA] \rightarrow [CROMA\_SE]$, *supp* 28,9 %, CF 0,99
$[CO] \rightarrow [CROMA\_SE]$, *supp* 31,4 %, CF 0,91

$[CARBONAT] \rightarrow [CROMA\_SE]$, *supp* 27,12 %, CF 0,8
$[CEC] \rightarrow [CROMA\_SE]$, *supp* 27,11 %, CF 0,71
$[AGUA] \rightarrow [CROMA\_SE]$, *supp* 29,12 %, CF 0,97

El porcentaje de carbono orgánico (atributo CO), la capacidad de intercambio de cationes (atributo CEC), el porcentaje de arcilla (atributo $ARCILLA$) y el de carbonatos (atributo $CARBONAT$) y el agua útil (atributo $AGUA$) del suelo forman parte del grupo más importante de las propiedades analíticas del suelo, por lo que, en consecuencia, estas reglas tienen un elevadísimo interés, en opinión de los expertos.

Las relaciones entre CEC y CO y el Value son bastante conocidas en el área de conocimiento estudiada. Al aumentar el contenido de carbono orgánico, o de "humus", en el suelo, siempre aumenta el Value (disminuyendo la luminosidad, tornándose más oscuro). En los suelos del área de estudio esto ocurre siempre, porque heredan el color claro de las rocas de la zona (caliza, dolomita, arenisca, ...), existiendo muy poco material geológico (no humidificado) de colores oscuros. Se comprueba también que este efecto es muchísimo más marcado en estado húmedo (atributo $VALUE\_HU$, Value en húmedo) que en seco (atributo $VALUE\_SE$, Value en seco, para el que el CF de la correspondiente dependencia, que no se muestra más arriba, llega a valer únicamente 0.54). La materia orgánica, en el ámbito de estudio abarcado, presenta una CEC mucho más elevada (en torno a 300 cmol(+)/kg) que el resto de componentes del suelo (por ejemplo, en el caso de la arcilla, en torno a 30 cmol(+)/kg), luego están estrechamente relacionadas y varían en el mismo sentido. Es por este motivo por el que la dependencia $[CEC] \rightarrow [VALUE\_HU]$ es fácilmente interpretable, aunque el factor de certeza sea algo menor.

Por otro lado, la asociación entre % de arcilla y Chroma en seco es casi perfecta. El aumento del % de arcilla, en todos los suelos del mundo, determina un Chroma más elevado porque está invariablemente asociado a la liberación de partículas finas (es decir, de un tamaño a la escala de la arcilla) de óxido de hierro que actúan como pigmentos, intensificando el color del suelo. Este efecto es mucho más marcado en seco, puesto que al humedecer la muestra disminuye la reflexión. El sentido de la variación del Chroma en seco con el carbono orgánico no es tan evidente, si bien aparecen muy asociados. En principio, el

aumento de humus provoca una pérdida de intensidad del color del suelo, pero para confirmarlo habría que ver las asociaciones locales (por medio de reglas de asociación difusas, por ejemplo) entre las distintas categorías de CO y $CROMA\_SE$. Hasta no tener resuelta dicha relación, sería imposible para los expertos averiguar el sentido de las dependencias que asocian el Croma en seco con la CEC y la cantidad de agua útil, ya que ambas dependen estrechamente del contenido en arcilla y carbono orgánico. Aún así, el que nos aparezcan ambas dependencias resulta hasta cierto punto razonable.

Por último, la dependencia entre el contenido de carbonatos y el Chroma en seco, $[CARBONAT] \rightarrow [CROMA\_SE]$, se presenta muy interesante, en opinión de los expertos. A priori, existirían argumentos para sostener tanto una dependencia negativa como positiva entre los atributos, aunque la hipótesis por la que se inclinarían los expertos sería más bien la primera, ya que un elevado porcentaje de carbonatos debería llevar a hallar un color blancuzco y poco intenso para el suelo. En este caso, sería muy conveniente acudir al estudio de relaciones a nivel local.

No obstante, como conclusión y a la vista de estos resultados, se puede decir que, a nivel de expertos edafólogos, los ofrecidos por el modelo difuso resultan mucho más satisfactorios que los que anteriormente se obtuvieron por técnicas "crisp".

4.10. Análisis empírico

Por último, dedicaremos este apartado a un estudio práctico de la bondad del procedimiento presentado en el apartado 4.7.5. Los procesos de minería de datos son, por lo general, bastante costosos en tiempo y en memoria cuando el problema supera una cierta complejidad, tanto si hablamos del tamaño del conjunto de datos como de la propia complejidad de éstos.

En capítulos anteriores hemos visto cómo, para poder representar información imprecisa, hemos de definir estructuras auxiliares sobre los modelos existentes para permitir a éstos manejar ese tipo de información. Todo esto se traduce en un aumento de la complejidad en la descripción de los datos. De ahí que, si decíamos que un proceso de minería de datos puede ser ya de por sí costoso, aplicado sobre datos de carácter difuso puede disparar los

requerimientos en tiempo y memoria.

4.10.1. Estudio de la eficiencia

En el caso que nos ocupa, en nuestra búsqueda de dependencias aproximadas difusas, hemos partido del algoritmo Apriori (ver página 237) para extracción de reglas de asociación, por su simplicidad y por ser uno de los más conocidos. Originalmente, siendo n el número de transacciones y m el número de ítems, en el peor caso hemos de considerar hasta 2^m ítemset. Dado que para cada uno de ellos hemos de obtener su soporte, esto es, contar cuántas veces aparecen en la relación, el orden total del algoritmo llega a ser de $\mathcal{O}(n \cdot 2^m)$. No obstante, si expresamos el orden en función únicamente del número de tuplas, podemos reducir éste a $\mathcal{O}(n)$. En lo sucesivo, entenderemos que expresamos el orden en función del número de tuplas.

Más adelante, para obtener reglas de asociación difusas, el algoritmo Apriori puede extenderse como se muestra en [Delgado et al., 2003a], multiplicando el orden por una constante k, correspondiente al número de α-cortes que considerábamos para almacenar los grados difusos.

Vimos en el apartado 2.3.6 cómo, para extraer dependencias aproximadas en términos de reglas de asociación podíamos definir una transformación sobre la relación original [Blanco et al., 2000], de forma que las reglas de asociación extraídas en la relación transformada se corresponden con dependencias aproximadas en la relación original. El principal inconveniente estribaba en que se aumentaba el orden del algoritmo de n a n^2, al aumentar en esa proporción el número de tuplas a considerar (hemos de considerar n^2 transacciones a partir de n tuplas). No obstante, a través de un resultado descrito en dicho trabajo (y en el apartado 2.3.6.3 de esta memoria), comprobábamos cómo era posible reducir el orden del algoritmo para que éste se halle en $\mathcal{O}(n)$.

Nuestro algoritmo (que podemos encontrar en la página 245), parte de la extensión de Apriori comentada en el párrafo anterior, a la que hemos de añadir la consideración de los α-cortes. De esta forma, en el mejor de los casos, podemos pensar que el orden de nuestro algoritmo se encuentra aceptablemente en $\mathcal{O}(k \cdot n)$. Considerando que el número de niveles, k, es constante, el orden del algoritmo sigue siendo en esencia $\mathcal{O}(n)$.

Sin embargo, hemos de incluir otro factor, el de considerar relaciones de similitud entre los valores de los atributos. De acuerdo con nuestro algoritmo, estas relaciones entran en juego cuando hemos de contar el soporte total de un conjunto de atributos, proceso que lleva a cabo el algoritmo B.10 (que podemos encontrar en la página 244). Siendo m el número de atributos y n el número de tuplas de la relación, en el peor de los casos habrá que considerar hasta $m \cdot \binom{n}{2}$ posibles pares de atributos relacionados, tomando n como el tamaño máximo del dominio de cada atributo.

Puesto que hemos de realizar el paso anterior en cada iteración, el orden resultante, en el peor de los casos, puede llegar a ser hasta de $\mathcal{O}(n \cdot m \cdot \binom{n}{2})$ (teniendo en cuenta además el factor multiplicativo k), lo cual puede resultar bastante costoso para bases de datos suficientemente voluminosas.

Como hemos visto, el verdadero cuello de botella de nuestro algoritmo se encuentra en el tratamiento de las relaciones de similitud, por lo que en futuros trabajos nos centraremos especialmente en este aspecto, para estudiar cuándo debe ser conveniente su uso y hasta qué punto puede mejorarse el cálculo con las mismas. En el capítulo dedicado a la implementación, veremos una primera solución al problema.

En la tabla 4.18 mostramos algunas medidas reales de tiempo y memoria obtenidas en las ejecuciones de nuestro algoritmo realizadas durante la etapa de experimentación. Los datos corresponden a la extracción de dependencias aproximadas difusas en las dos bases de datos previamente descritas (el conjunto de datos STULONG y la base de datos sobre Color de suelos), tomando los siguientes umbrales: número de niveles $k = 100$, soporte mínimo $minsupp = 0{,}1$ y factor de certeza mínimo, $minCF = -1{,}0$.

Los experimentos se realizaron sobre un PC bajo Microsoft Windows 2000 SP3 con las siguientes características: procesador Pentium4, 1.7 GHz, con 1GB RAM. Dicho PC se conectaba remotamente a un servidor de bases de datos Oracle® (Oracle9i Enterprise Edition Release 9.2.0.1.0) que cuenta con un procesador Pentium III Dual a 1GHz y 1.5GB RAM.

Las columnas representan, en este orden, el número de ítemsets obtenidos en cada iteración, el número de dependencias aproximadas difusas obtenidas, el tiempo total empleado y la memoria usada. Este último dato es aproximado, y ha de considerarse como meramente indicativo, ya que en la cifra total de

Tabla 4.18: Comparación de resultados de ejecuciones

	1-ít.	2-ít.	3-ít.	DAD's	t(s)	mem.(KB)
STULONG	31	362	1102	7336	687.919	64401
BDColor	30	308	1383	8914	1686.665	103997

KBs utilizados hay que contar la memoria usada por el resto de estructuras necesarias para el funcionamiento del programa.

Recordemos que la tabla de datos de STULONG consta de un total de 1417 casos descritos por 32 variables, mientras que la base de datos sobre Color de suelos viene descrita por 34 atributos para un total de 541 casos. Para las relaciones de similitud definidas sobre los valores de los atributos de la primera base de datos se cuenta con un total de 73 pares, frente a los 1040 pares con los que cuenta la segunda base de datos. Esto se refleja en el significativo aumento en el tiempo de ejecución que se nos muestra en la tabla. Por lo demás, el resto de valores resulta similar (o, en todo caso, proporcional) para ambos conjuntos de datos.

4.10.2. Análisis experimental basado en el número de niveles

Una segunda ronda de experimentos está destinada a la comparación de resultados en función del parámetro k, esto es, el número de niveles que usamos para almacenar el soporte de los ítemsets difusos.

Nuestro objetivo va a ser ahora el de comparar diversos conjuntos de dependencias aproximadas difusas obtenidos en función de k. Tomando 0 como mínimo umbral de soporte y -1 como mínimo factor de certeza, realizamos varias ejecuciones de nuestro algoritmo para $k = 3, 5, 10, 20, 50, 100$.

Para estudiar hasta qué punto dependen los resultados de k realizamos un ANOVA (análisis de las varianzas) sobre los soportes y factores de certeza de los conjuntos resultantes de reglas, mostrando los resultados de este análisis a continuación.

4.10.2.1. ANOVA sobre los resultados de datos médicos

Se realizó un total de 6 ejecuciones, correspondientes a los distintos valores de k considerados, del algoritmo de extracción de dependencias aproximadas difusas sobre el conjunto de datos médicos suministrados por STULONG, tomando como umbral el mínimo posible, esto es, 0.

Recogimos los valores de soporte y factor de certeza para todas las dependencias obtenidas, y asumimos las tres condiciones de las que precisa el análisis de la varianza para poder ser aplicado:

- Asumimos que las distribuciones de los soportes y los factores de certeza son normales.
- Asumimos que las varianzas para cada población son iguales.
- Consideramos que cada muestra es una muestra aleatoria de valores, y los errores que puedan afectar a una muestra son independientes de los que puedan afectar a otra.

Realizamos un ANOVA sobre los datos, y mostramos los resultados obtenidos en la tabla 4.19.

Tabla 4.19: Resultados del ANOVA para la base de datos médicos

		Suma de cuadrados	gl	Media cuadrática	F	Sig.
SOPORTE	Inter-grupos	513.170	5	102.634	9.109	.000
	Intra-grupos	1076703.83	94620	11.379		
	Total	1077217.00	94625			
CF	Inter-grupos	.032	5	.006	.199	.963
	Intra-grupos	3040.843	94620	.032		
	Total	3040.875	94625			

A la vista de los resultados mostrados en la tabla, vemos cómo apenas si hay diferencia entre las medias de la población asociada al factor de certeza, pero sí para el soporte. Ello nos viene indicado por la última columna de la

tabla, que mide la significación que se le da al estadístico F, siendo ésta inferior a 0.05 para el caso del soporte.

Por esta razón, conviene estudiar mediante tests *Post Hoc* la medida en que existe esa variación entre medias para el caso del soporte. Estos tests miden las importancia de la diferencia entre las medias de las distintas poblaciones. Se aplicaron los tests estadísticos de HSD de Tukey, Scheffé y DMS. Éstos nos aportan algo más de información, tal y como se extrae de la tabla 4.20. En dicha tabla, mostramos los resultados proporcionados por el test DMS, al ser los más restrictivos, aunque los tres tests aporten aproximadamente la misma conclusión. Las celdas de la tabla se marcan con una *"x"* si entre las medias de las dos poblaciones hay una diferencia significativa a nivel ,05.

Tabla 4.20: Tabla cruzada Post Hoc para el soporte para la base de datos médicos

	3	5	10	20	50	100
3			x	x	x	x
5				x	x	x
10	x				x	x
20	x	x				
50	x	x	x			
100	x	x	x			

Los resultados de los tests estadísticos *Post Hoc* revelan cómo, para este ejemplo en particular, los soportes de las dependencias no van a variar para valores de k iguales a 20, 50 y 100 (a los que habría que añadir 10, si nos regimos por los tests HSD de Tukey y Scheffé). Tampoco habrá variación tomando como valores de k 3 y 5, aunque siempre será más conveniente tomar la precisión más alta.

El porqué de la pequeña variación entre valores en el caso de los factores de certeza podemos encontrarlo en el hecho de que la mayoría de los atributos considerados en este ejemplo son categóricos y los grados de cumplimiento suelen ser iguales a 1 en gran parte de los mismos, a pesar de que entren en juego las relaciones difusas de similitud.

En cambio, al tratar con atributos numéricos sobre cuyos dominios se ha definido un conjunto de etiquetas lingüísticas asociadas a distribuciones de

posibilidad, la variación debería ser más notoria, como veremos que ocurre en el ejemplo siguiente.

4.10.2.2. ANOVA sobre los resultados de datos de color de suelos

De forma análoga para el conjunto de datos anterior, procedimos a recoger los soportes y factores de certeza de las dependencias aproximadas difusas obtenidas de 6 ejecuciones de nuestro algoritmo sobre la base de datos de color de suelos.

Asumidas las condiciones citadas anteriormente, procedimos a realizar el ANOVA, cuyos resultados se muestran en la tabla 4.21. Como se extrae de dicha tabla, vemos cómo de nuevo existe una dependencia entre el valor que tomemos para k y los valores de soporte y factor de certeza que se obtendrán como resultado del algoritmo. El valor del estadístico F se sale de escala, con una significación nula.

Tabla 4.21: Resultados del ANOVA para la base de datos de Color de suelos

		Suma de cuadrados	gl	Media cuadrática	F	Sig.
SOPORTE	Inter-grupos	38991.344	5	7798.269	247.390	.000
	Intra-grupos	1307033.864	41464	31.522		
	Total	1346025.208	41469			
CF	Inter-grupos	7.079	5	1.416	29.153	.000
	Intra-grupos	2013.612	41464	.049		
	Total	2020.691	41469			

Podemos tratar de estudiar más a fondo esa relación, aplicando los mismos tests estadísticos *Post Hoc* que en el caso anterior, esto es, los tests HSD de Tukey, Scheffé y DMS.

Sobre los resultados de los distintos tests *Post Hoc* aplicados, se pueden construir las tablas cruzadas que se muestran en la tabla 4.22. Para el caso del soporte, los tres tests devuelven la misma tabla, mientras que para el factor de certeza hay pequeñas variaciones que no llegan a influir demasiado en la

interpretación. La tabla que se muestra es la devuelta por el test Scheffé.

Tabla 4.22: Tablas cruzadas Post Hoc para el soporte y el factor de certeza para la base de datos del color de suelos

	3	5	10	20	50	100
3		x	x	x	x	x
5	x		x	x	x	x
10	x	x		x	x	x
20	x	x	x		x	x
50	x	x	x	x		
100	x	x	x	x		

	3	5	10	20	50	100
3						
5			x	x	x	x
10		x			x	x
20		x				
50		x	x			
100		x	x			

De la tabla 4.22 se desprende que, en función del número de niveles que consideremos, la valoración de las dependencias va a verse influída. Para el soporte, los tests aplicados nos dicen que la diferencia entre medias es significativa al nivel ,05 salvo en los casos $k = 50$ y $k = 100$. Esto se interpreta como que, para este ejemplo en concreto, hubiéramos obtenido los mismos resultados tomando cualquiera de estos dos valores como número de niveles.

Para el caso de la certeza, nos aparece una menor diferencia entre medias, destacable sobre todo en el caso $k = 3$, aunque podríamos quedarnos con la esquina inferior derecha de la tabla, en la que se nos dice que resultaría indiferente utilizar un número de niveles igual a 20, 50 ó 100 en nuestro algoritmo.

4.11. Conclusiones y discusión

En este capítulo hemos propuesto una metodología para obtener lo que hemos llamado Dependencias Aproximadas Difusas a partir de una base de datos. Las dependencias aproximadas difusas generalizan en varios sentidos a las dependencias funcionales suavizadas, y nos proporcionan información sobre relaciones a nivel de atributos. Hemos enumerado algunas situaciones en las que los conceptos introducidos puede ser útiles, tanto para analizar datos clásicos como, obviamente, pues éste era nuestro objetivo primordial, datos difusos. La metodología propuesta puede implementarse modificando

algoritmos ya existentes para la minería de reglas de asociación sin que por ello se incremente la complejidad teórica, aunque sí se vean incrementados el tiempo de ejecución y el espacio de memoria necesario, al deber multiplicarlos por una constante relacionada con el número de α-cortes (nivel de precisión) que consideremos. Como añadido, incluímos un estudio experimental sobre los factores que influyen en la eficiencia del algoritmo y la bondad de los resultados.

Hemos aplicado esta metodología para adaptar el algoritmo Apriori con el objeto de que nos permita obtener dependencias aproximadas difusas. Nuestros primeros experimentos nos sugieren que tanto el tiempo como el espacio de memoria empleados en el proceso de minería pueden considerarse aceptables, habida cuenta de lo costosos que pueden llegar a ser este tipo de procesos.

En el capítulo siguiente, veremos cómo, en particular, podemos aplicar dependencias aproximadas difusas en el análisis de correspondencias entre particiones difusas sobre un mismo conjunto de objetos.

5. Dependencias Aproximadas y Análisis de Correspondencias

En el marco general de esta memoria, partíamos del hecho de que en la actualidad nos hallamos en la situación de que a menudo se deben manejar grandes cantidades de información, a menudo proviniente de distintas fuentes. Uno de los problemas que se nos puede plantear al respecto es el de la fusión de conocimiento proviniente de distintas fuentes [Appriou et al., 2001]. En particular un caso concreto de fusión de conocimiento es la integración de dos o más clasificaciones diferentes sobre un mismo conjunto de datos.

Este problema comprende varias fases, siendo la primera de ellas un análisis de correspondencias, para el que se puede utilizar un conjunto de técnicas bien definidas en Estadística. Nuestra aportación en este capítulo es la de una metodología alternativa al análisis de correspondencias basada en técnicas de minería de datos. Mientras que el análisis clásico de correspondencias basa su interpretación en la medida de distancias sobre un mapa perceptual, nuestro enfoque se centrará en la interpretación facilitada por la valoración de un conjunto de reglas o dependencias. Adicionalmente, nuestra metodología

puede aplicarse en aquellos casos en los que haya que tratar con particiones de carácter difuso sobre los datos.

5.1. Análisis de correspondencias clásico

El análisis de correspondencias aparece por primera vez en [Benzécri, 1963], con la finalidad básica la de describir las relaciones existentes entre dos variables nominales, mediante la representación gráfica los datos provenientes de una tabla de contingencia, obtenida como la tabulación cruzada de las dos variables. Partiendo de dicha tabla, transforma los datos no métricos en un nivel métrico y realiza una reducción dimensional (similar al análisis factorial) y un mapa perceptual (similar al análisis multidimensional). A modo de ejemplo, las preferencias por una marca de los encuestados pueden ser tabuladas de forma cruzada con variables demográficas (p.e., género, categorías de renta, ocupación) indicando cuánta gente que prefiere cada una de las marcas entra dentro de cada categoría de las variables demográficas. A través del análisis de correspondencias, la asociación o "correspondencia" de marcas y las características distintivas de aquellos que prefieren cada marca se muestran en un mapa bi o tridimensional, tanto de marcas como características de los encuestados. Las marcas percibidas como similares están localizadas en una cercana proximidad unas de otras. De la misma forma, las características más distintivas de los encuestados que prefieren cada marca están determinadas también por la proximidad de las categorías de las variables demográficas respecto de la posición de la marca. El análisis de las correspondencias proporciona una representación multivariante de la interdependencia de datos no métricos que no es posible realizar con otros métodos.

5.1.1. Formulación

Dadas dos variables I y J, mediante las que representamos dos particiones o conjuntos de clases (o modalidades) definidas sobre el conjunto de objetos que estamos estudiando, una población de n individuos, podemos construir una tabla de contingencia cruzando las modalidades de las dos variables, tal y como se muestra en la tabla 5.1.

Tabla 5.1: Tabla de contingencia

	1	...	j	...	J
1			$\vdots$		
$\vdots$			$\vdots$		
i	...	...	k_{ij}	...	...
$\vdots$			$\vdots$		
I			$\vdots$		

El valor k_{ij} representa el número de individuos que poseen a la vez la modalidad i de la primera variable y la modalidad j de la segunda. La interpretación que nosotros le daremos es que k_{ij} representa el número de elementos comunes entre las clases i y j. En cualquier caso, debe cumplirse que $\sum_i \sum_j k_{ij}$ es igual al número total de objetos.

Otra representación alternativa que se suele considerar a menudo es la de una tabla de frecuencias relativas, obtenida dividiendo cada efectivo k_{ij} por el efectivo total, o número de elementos, n. Esta nueva tabla define una medida de probabilidad sobre el conjunto producto $I \times J$. Sus marginales, o probabilidades marginales, tienen por término general $f_{\cdot j}$ o $f_{i\cdot}$.

Se cumplen las siguientes igualdades:

- $f_{ij} = k_{ij}/n$
- $f_{i\cdot} = \sum_j f_{ij}$
- $f_{\cdot j} = \sum_i f_{ij}$
- $\sum_i f_{i\cdot} = \sum_j f_{\cdot j} = \sum_i \sum_j f_{ij} = 1$

Una tabla de contingencia expresa por tanto la relación entre dos variables. Para una medida de probabilidad, se dice que existe independencia cuando $\forall i$ y $\forall j$ se cumple

$$f_{ij} = f_{i\cdot} f_{\cdot j}$$

Existe relación entre las dos variables cuando en algunas celdas de la tabla, f_{ij} es diferente del producto $f_{i\cdot}f_{\cdot j}$.

- Si $f_{ij} > f_{i\cdot}f_{\cdot j}$, se dice que las modalidades i y j se atraen.
- Si $f_{ij} < f_{i\cdot}f_{\cdot j}$, existe una repulsión entre estas dos modalidades.

Entre los objetivos del análisis de correspondencias se cuenta el intento de obtener una tipología de las filas, una tipología de las columnas y relacionar ambas entre sí. Dos filas (columnas) se considerarán próximas si se asocian del mismo modo al conjunto de las columnas (filas). Esta aproximación permite estudiar la desviación de la tabla de la hipótesis de independencia.

La tabla original no se utiliza directamente. En lugar de eso, los datos han de transformarse en perfiles, dividiendo cada término en una fila por la frecuencia marginal de dicha fila (e igualmente para las columnas). Los perfiles-fila y perfiles-columna resultantes se pueden representar como puntos en sendos mapas perceptuales que tratarán de ajustarse midiendo las distancias entre perfiles. Para ello se usa la distancia χ^2.

El análisis de correspondencias se puede utilizar también para reducir la dimensión de los datos conservando la mayor información posible, con vistas a un tratamiento estadístico ulterior (clasificación, regresión, análisis discriminante, ...) o a una transmisión de información.

Para encontrar más información sobre el análisis de correspondencias, recomendaremos algunos trabajos como [Escofier, 1992, Cox, 1994], o, si lo que se desea es obtener una visión más general sobre el análisis multivariante, invitamos al lector a la consulta de [Hair et al, 1999].

5.2. Definición del problema de acuerdo a nuestra metodología

El análisis de correspondencias clásico basa su funcionamiento en la interpretación de la cercanía entre clases sobre un mapa perceptual que se construye a partir de la tabla de contingencia. Frente a esta técnica, expondremos a continuación una metodología alternativa para afrontar dicho análisis me-

Tabla 5.2: Tabla $r_{\mathcal{AB}}$

Objeto	**tupla**	$X_{\mathcal{A}}$	$X_{\mathcal{B}}$
o_1	t_{o_1}	A_1	B_2
o_2	t_{o_2}	A_2	B_2
o_3	t_{o_3}	A_1	B_1
$\vdots$	$\vdots$	$\vdots$	$\vdots$

diante la extracción e interpretación de reglas de asociación y dependencias aproximadas.

Para ello, formularemos el problema de una forma distinta. En lugar de representar la información en una tabla de contingencia, desglosaremos el conjunto de objetos, indicando a qué clases pertenece cada objeto por separado. Formalmente, sea O un conjunto de objetos, finito, sobre el que tenemos definidas dos particiones, $\mathcal{A} = \{A_1, A_2, \ldots, A_p\}$ y $\mathcal{B} = \{B_1, B_2, \ldots, B_q\}$, o lo que es igual, $A_i, B_j \subseteq O$ y $A_i, B_j \neq \emptyset$, con $A_{i_1} \cap A_{i_2} = \emptyset \ \forall i_1, i_2 \in \{1, \ldots, p\}$ y $B_{j_1} \cap B_{j_2} = \emptyset \ \forall j_1, j_2 \in \{1, \ldots, q\}$. Además

$$\bigcup_{A_i \in \mathcal{A}} A_i = \bigcup_{B_j \in \mathcal{B}} B_j = O$$

Como hemos mencionado anteriormente, vamos a representar las particiones $\mathcal{A}$ y $\mathcal{B}$ por medio de una tabla, por lo que usaremos la notación para bases de datos relacionales. Sea $r_{\mathcal{AB}}$ esta tabla. Cada fila (tupla) y columna (atributo) de $r_{\mathcal{AB}}$ vendrán asociadas a un objeto y a una partición, respectivamente. De esta forma, asumimos que $|r_{\mathcal{AB}}| = |O|$.

Notaremos por t_o a la tupla asociada al objeto o, y $X_{\mathcal{P}}$ como el atributo asociado a la partición $\mathcal{P}$. El valor correspondiente a la tupla t_o para el atributo $X_{\mathcal{P}}$, $t_o[X_{\mathcal{P}}]$, será la clase para o de acuerdo con $\mathcal{P}$, esto es, $t_o[X_{\mathcal{P}}] \in \mathcal{P}$. Mostramos un esquema de lo expuesto en la tabla 5.2.

Obviamente, dados $A_i \in \mathcal{A}$ y $o \in A_i$, se cumple que $t_o[\mathcal{A}] = A_i$.

5.2.1. Correspondencias

Nuestro propósito es el de estudiar los siguientes tipos de posibles correspondencias entre $\mathcal{A}$ y $\mathcal{B}$:

Definición 5.2.1 Correspondencia local. *Sean $A_i \in \mathcal{A}$ y $B_j \in \mathcal{B}$. Decimos que existe una correspondencia local de A_i hacia B_j, que notamos como $A_i \Rightarrow B_j$, si $A_i \subseteq B_j$.*

Definición 5.2.2 Correspondencia parcial. *Del mismo modo, existe una correspondencia parcial de $\mathcal{A}$ hacia $\mathcal{B}$, que notaremos por $\mathcal{A} \Rrightarrow \mathcal{B}$, si $\forall A_i \in \mathcal{A}$ $\exists B_j \in \mathcal{B}$ tal que $A_i \subseteq B_j$.*

Definición 5.2.3 Correspondencia global. *Por último, diremos que existe una correspondencia global entre $\mathcal{A}$ y $\mathcal{B}$, denotada por $\mathcal{A} \equiv \mathcal{B}$, cuando ocurre que $\mathcal{A} \Rrightarrow \mathcal{B}$ y $\mathcal{B} \Rrightarrow \mathcal{A}$.*

Las correspondencias parciales y globales asocian particiones, mientras que las correspondencias locales hacen lo mismo entre clases. Las correspondencias locales son interesantes por dos motivos:

- Si se cumple una correspondencia parcial dada, las correspondencias locales nos permiten describir las correspondencias particulares entre clases de $\mathcal{A}$ hacia $\mathcal{B}$. A este conjunto de correspondencias lo denominaremos modelo de la correspondencia, en el mismo sentido propuesto en el apartado 2.3.6.1.

- Si una correspondencia parcial dada no se cumple en su totalidad, aún es posible que las correspondencias locales nos aporten información sobre algunas relaciones menos generales entre particiones.

En este punto hemos de destacar el hecho de que no estamos interesados únicamente en correspondencias perfectas, sino también en correspondencias que admitan posibles excepciones. Esto es así porque resulta muy difícil en general encontrar una correspondencia perfecta entre particiones. Al mismo tiempo, si sólo unos pocos objetos incumplen la correspondencia, este mismo hecho continúa siendo un dato interesante para nosotros. En consecuencia, nos interesa medir la bondad de las correspondencias (sean locales, parciales o globales) entre particiones.

5.2.2. Propiedades

Algunas propiedades sobre las recién definidas correspondencias son las siguientes:

Proposición 5.2.1 *$\mathcal{A} \Rrightarrow \mathcal{B}$ si y sólo si para cada $o \in O$, $t_o[X_\mathcal{A}] \subseteq t_o[X_\mathcal{B}]$.*

Demostración:

$\Rightarrow$ Si $\mathcal{A} \Rrightarrow \mathcal{B}$, entonces tenemos que $\forall A_i \in \mathcal{A}\ \exists B_j \in \mathcal{B}$ tal que $A_i \subseteq B_j$. En particular, $\forall o \in O\ \exists B_j \in \mathcal{B}$ tal que $t_o[\mathcal{A}] \subseteq B_j$. Y como también ocurre que $o \in t_o[\mathcal{A}]$ y $o \in t_o[\mathcal{B}]$, entonces $t_o[\mathcal{B}] = B_j$ y en consecuencia $t_o[\mathcal{A}] \subseteq t_o[\mathcal{B}]$.

$\Leftarrow$ Sea $A_i \in \mathcal{A}$. Como $A_i \neq \emptyset$, esto quiere decir que $\exists o \in O$ tal que $o \in A_i$ y por lo tanto $A_i = t_o[\mathcal{A}] \subseteq t_o[B] \in \mathcal{B}$. En conclusión, $\mathcal{A} \Rrightarrow \mathcal{B}$.

□

Lo que nos quiere decir la proposición 5.2.1, es que vamos a considerar el que exista una correspondencia parcial $\mathcal{A} \Rrightarrow \mathcal{B}$ si, para cualquier $o \in O$, conociendo la clase a la que pertenece de acuerdo con $\mathcal{A}$, somos capaces de determinar la clase a la que pertenece de acuerdo con $\mathcal{B}$.

Proposición 5.2.2 *Si $\mathcal{A} \Rrightarrow \mathcal{B}$ entonces $\forall B_j \in \mathcal{B}\ \exists \mathcal{A}' \subseteq \mathcal{A}$ tal que*

$$\bigcup_{A_i \in \mathcal{A}'} A_i = B_j$$

Demostración: Supongamos que tenemos la correspondencia parcial $\mathcal{A} \Rrightarrow \mathcal{B}$ y sea $B_j \in \mathcal{B}$. Sea también $\mathcal{A}' = \{t_o[\mathcal{A}] \mid o \in B_j\}$. Obviamente, ocurre lo siguiente

$$B_j \subseteq \bigcup_{A_i \in \mathcal{A}'} A_i$$

Por otra parte, $t_o[\mathcal{A}] \subseteq t_o[\mathcal{B}] = B_j\ \forall t_o[\mathcal{A}] \in \mathcal{A}'$ (prop. 5.2.1). En consecuencia,

$$\bigcup_{A_i \in \mathcal{A}'} A_i \subseteq B_j$$

Por lo tanto

$$\bigcup_{A_i \in \mathcal{A}'} A_i = B_j$$

□

Corolario 5.2.2.1 *Si* $\mathcal{A} \equiv \mathcal{B}$ *entonces* $\forall A_i \in \mathcal{A}\ \exists B_j \in \mathcal{B}$ *tal que* $A_i = B_j$.

Corolario 5.2.2.2 *Si* $\mathcal{A} \equiv \mathcal{B}$ *entonces* $\mathcal{A} = \mathcal{B}$.

5.3. Correspondencias locales

Para estudiar correspondencias locales entre varias particiones aplicaremos reglas de asociación [Agrawal et al., 1993], y más concretamente, el modelo estudiado en [Sánchez, 1999, Berzal et al., 2001a]. Remitimos al lector al lugar apropiado de esta memoria (apartado 2.3.4) para más detalles sobre representación y medidas de interés y precisión.

5.3.1. Análisis de correspondencias locales mediante reglas de asociación

El análisis de correspondencias locales puede realizarse a partir de la obtención de reglas de asociación en la tabla $r_{\mathcal{AB}}$, donde representamos las particiones $\mathcal{A}$ y $\mathcal{B}$.

Siguiendo el mismo esquema de representación de las reglas de asociación, vamos a notar los ítems que estamos usando como $[X_{\mathcal{A}} = A_i]$ y $[X_{\mathcal{B}} = B_j]$, y las reglas

$$[X_{\mathcal{A}} = A_i] \Rightarrow [X_{\mathcal{B}} = B_j]$$

y

$$[X_{\mathcal{B}} = B_j] \Rightarrow [X_{\mathcal{A}} = A_i]$$

nos muestran las posibles correspondencias locales entre las clases A_i y B_j.

En concreto, el factor de certeza de estas reglas nos va a medir la precisión de las correspondencias locales asociadas. Si notamos como $Supp(A_i, B_j)$, $Conf(A_i, B_j)$ y $CF(A_i, B_j)$ el soporte, confianza y factor de certeza, respectivamente, de la regla

$$[X_{\mathcal{A}} = A_i] \Rightarrow [X_{\mathcal{B}} = B_j]$$

podemos definir entonces las siguientes propiedades:

Proposición 5.3.1 *El soporte para el ítem* $[X_{\mathcal{A}} = A_i]$ *se calcula como*

$$Supp([X_{\mathcal{A}} = A_i]) = \frac{|A_i|}{|O|} \tag{5.1}$$

Demostración: Trivial. □

Proposición 5.3.2 *El soporte de la regla* $[X_{\mathcal{A}} = A_i] \Rightarrow [X_{\mathcal{B}} = B_j]$ *se obtiene mediante la expresión*

$$Supp(A_i, B_j) = \frac{|A_i \cap B_j|}{|O|} \tag{5.2}$$

Demostración: Trivial. □

Proposición 5.3.3 *La confianza de la regla* $[X_{\mathcal{A}} = A_i] \Rightarrow [X_{\mathcal{B}} = B_j]$ *equivale a calcular*

$$Conf(A_i, B_j) = \frac{Supp(A_i, B_j)}{Supp([X_{\mathcal{A}} = A_i])} = \frac{|A_i \cap B_j|}{|A_i|} \tag{5.3}$$

Demostración: Trivial. □

Proposición 5.3.4 *Si* $CF(A_i, B_j) = 1$, $A_i \subseteq B_j$.

Demostración: Sea $CF(A_i, B_j) = 1$. Entonces, por la ecuación (2.7) se deduce que $Conf(A_i, B_j) = 1$, y por la ecuación (5.3) $|A_i \cap B_j| = |A_i|$. Por lo tanto, $A_i \subseteq B_j$. □

Es decir, una regla con máxima precisión representa una correspondencia local entre la clase del antecedente y la clase del consecuente. De lo que resulta inmediato el siguiente corolario,

Corolario 5.3.4.1 *Si* $CF(A_i, B_j) = CF(B_j, A_i) = 1$, $A_i = B_j$.

En general, si $CF(A_i, B_j) > 0$, saber que $o \in A_i$ aumenta nuestra creencia en que $o \in B_j$. El valor de $CF(A_i, B_j)$ mide este incremento.

Proposición 5.3.5 *Si* $CF(A_i, B_j) = 0$,

$$\frac{|A_i \cap B_j|}{|B_j|} = \frac{|A_i|}{|O|}. \tag{5.4}$$

Demostración: Supongamos que $CF(A_i, B_j) = 0$. De ahí obtenemos que $Conf(A_i, B_j) = Supp([X_{\mathcal{B}} = B_j])$, y por la ecuación (5.3) tenemos que

$$\frac{Supp(A_i, B_j)}{Supp([X_{\mathcal{A}} = A_i])} = Supp([X_{\mathcal{B}} = B_j])$$

donde podemos despejar y

$$\frac{Supp(A_i, B_j)}{Supp([X_{\mathcal{B}} = B_j])} = Supp([X_{\mathcal{A}} = A_i])$$

Por lo que podemos concluir que

$$\frac{|A_i \cap B_j|}{|B_j|} = \frac{|A_i|}{|O|}$$

□

Esta proposición nos indica que la proporción de objetos de B_j que están en A_i es igual que la proporción de objetos de O en A_i, es decir, cuando $CF(A_i, B_j) = 0$ tenemos una independencia estadística entre los hechos $o \in A_i$ y $o \in B_j$.

Proposición 5.3.6 *Si* $CF(A_i, B_j) = -1$, $A_i \cap B_j = \emptyset$.

Demostración: Sea $CF(A_i, B_j) = -1$. Entonces, por la ecuación (2.8) deducimos $Conf(A_i, B_j) = 0$, y por la ecuación (5.3) $|A_i \cap B_j| = 0$. En consecuencia, $A_i \cap B_j = \emptyset$. □

En general, si $CF(A_i, B_j) < 0$, saber que $o \in A_i$ reduce nuestra creencia en que $o \in B_j$. El valor de $CF(A_i, B_j)$ mide este decremento.

5.4. Correspondencias parciales y globlales

Por definición, el análisis de correspondencias parciales y globales puede realizarse a partir de las correspondencias locales, pero seguimos necesitando una forma de integrar la información proporcionada por éstas. Las dependencias aproximadas [Bra y Paredaens, 1983, Ziarko, 1991] constituyen una interesante herramienta para integrar la información dada por un conjunto de reglas de asociación.

5.4.1. Análisis de correspondencias parciales mediante dependencias aproximadas

Para el análisis de correspondencias parciales usaremos dependencias aproximadas, aplicadas en el sentido de que si la dependencia $X_{\mathcal{A}} \rightarrow X_{\mathcal{B}}$ se cumple, existe una correspondencia parcial de $\mathcal{A}$ a $\mathcal{B}$, cuya precisión viene medida por el factor de certeza de la dependencia, $X_{\mathcal{A}} \rightarrow X_{\mathcal{B}}$. Hemos de destacar las siguientes propiedades:

Proposición 5.4.1 *$CF(X_{\mathcal{A}}, X_{\mathcal{B}}) = 1$ si y sólo si $\mathcal{A} \Rrightarrow \mathcal{B}$.*

Demostración:

$\Rightarrow$ Si $CF(X_{\mathcal{A}}, X_{\mathcal{B}}) = 1$ entonces $X_{\mathcal{A}} \rightarrow X_{\mathcal{B}}$ es una dependencia funcional, es decir, existe una correpondencia perfecta entre los valores de $X_{\mathcal{A}}$ y $X_{\mathcal{B}}$. Como consecuencia de ello, $CF(A_i, B_j) = 1$, y así también $A_i \subseteq B_j$, por lo que finalmente, $\mathcal{A} \Rrightarrow \mathcal{B}$.

$\Leftarrow$ Si $\mathcal{A} \Rrightarrow \mathcal{B}$ entonces $A_i \subseteq B_j \; \forall A_i \in \mathcal{A}, B_j \in \mathcal{B}$. Por lo tanto, obtenemos que $CF(A_i, B_j) = Conf(A_i, B_j) = 1 \; \forall A_i \in \mathcal{A}, B_j \in \mathcal{B}$ y por la proposición (2.3.4), $CF(X_{\mathcal{A}}, X_{\mathcal{B}}) = 1$.

□

Proposición 5.4.2 ([Blanco et al., 2000]) *Si $X_{\mathcal{A}}$ y $X_{\mathcal{B}}$ son independientes, $CF(X_{\mathcal{A}}, X_{\mathcal{B}}) = 0$.*

Como vimos en [Sánchez, 1999, Delgado et al., 2000a, Blanco et al., 2000], existe una relación entre dependencias aproximadas y reglas de asociación, al poder representar las primeras en términos de las segundas. Dicha relación es extensible al caso de las correspondencias parciales y locales.

Una interesante extensión es aquélla en la que tenemos una jerarquía de particiones anidadas. En este caso, podemos representar los diferentes niveles jerárquicos como columnas en nuestra tabla, y analizar las correspondencias entre particiones a distintos niveles.

Otra propuesta que surge inmediatamente es la de ampliar el estudio al caso de particiones difusas, algo que será comentado más adelante en este mismo capítulo.

5.4.2. De correspondencias parciales a globales

Hemos usado el factor de certeza como medida de precisión para correspondencias tanto locales como parciales, por lo que nos interesa usar la misma medida para las correspondencias globales. De ahí la siguiente definición:

Definición 5.4.1 *El factor de certeza de* $\mathcal{A} \equiv \mathcal{B}$ *es*

$$\text{mín}\{CF(X_{\mathcal{A}}, X_{\mathcal{B}}), CF(X_{\mathcal{B}}, X_{\mathcal{A}})\}$$

El razonamiento implícito en esta definición parte del hecho de que una dependencia global $\mathcal{A} \equiv \mathcal{B}$ se cumple cuando las dependencias parciales $\mathcal{A} \Rrightarrow \mathcal{B}$ y $\mathcal{B} \Rrightarrow \mathcal{A}$ se cumplen, y resulta común obtener el factor de certeza de una conjunción de hechos como el mínimo de los factores de certeza de dichos hechos. La siguiente proposición también se cumple:

Proposición 5.4.3 ([Sánchez, 1999])

- $CF(A_i, B_j) > 0$ *sii* $CF(B_j, A_i) > 0$.
- *Si* $CF(A_i, B_j) \leq 0$,

$$CF(A_i, B_j) = CF(B_j, A_i)$$

Corolario 5.4.3.1 *El factor de certeza de* $\mathcal{A} \equiv \mathcal{B}$ *tiene siempre el mismo signo que los factores de certeza de* $\mathcal{A} \Rrightarrow \mathcal{B}$ *y* $\mathcal{B} \Rrightarrow \mathcal{A}$.

5.5. Análisis de varias particiones

Un problema que atrae nuestro interés es el del análisis de correspondencias entre más de dos particiones. Pueden darse varias situaciones, que pueden resolverse de diferentes formas. Una solución rápida es la de obtener dos particiones finales, combinando varias de ellas en una sola.

Una forma común de combinar particiones es la siguiente: Dadas las particiones $\mathcal{A}$ y $\mathcal{B}$, llamamos $\mathcal{AB}$ a la partición

$$\mathcal{AB} = \{A_i \cap B_j \mid A_i \in \mathcal{A}, B_j \in \mathcal{B}\} \setminus \{\emptyset\} \tag{5.5}$$

De esta forma, obtenemos una clasificación como el cruce del conjunto de objetos existentes en las particiones originales. Notaremos como A_iB_j al conjunto de objetos $A_i \cap B_j$.

Por simplicidad en la formulación, consideraremos únicamente tres particiones, aunque se podría extender y generalizar muy fácilmente. Sean $\mathcal{A}$, $\mathcal{B}$ y $\mathcal{C}$ tres particiones de O.

Proposición 5.5.1 *Se cumple la correspondencia local $A_iB_j \subseteq C_k$ si y sólo si la regla de asociación $[X_{\mathcal{A}} = A_i, X_{\mathcal{B}} = B_j] \Rightarrow [X_{\mathcal{C}} = C_k]$ se cumple con $CF = 1$.*

Demostración: Trivial. □

La interpretación que damos a esta correspondencia es que, conocidas las clases para un objeto o dado según $\mathcal{A}$ y $\mathcal{B}$, conocemos también la clase para o de acuerdo con $\mathcal{C}$.

Proposición 5.5.2 *La correspondencia parcial $\mathcal{AB} \Rrightarrow \mathcal{C}$ se cumple si y sólo si la dependencia aproximada $X_{\mathcal{A}}X_{\mathcal{B}} \rightarrow X_{\mathcal{C}}$ se cumple con $CF = 1$.*

Demostración: Trivial. □

Usaremos el factor de certeza, CF, de las reglas y las dependencias para medir la bondad de las correspondencias locales y parciales, respectivamente. Se cumplen las siguientes propiedades:

Proposición 5.5.3 $Conf(A_i, B_jC_k) \leq Conf(A_i, C_k)$.

Demostración:

$$Conf(A_i, B_jC_k) = \frac{supp(A_i \cap B_j \cap C_k)}{supp(A_i)} \leq$$

$$\leq \frac{supp(A_i \cap C_k)}{supp(A_i)} = Conf(A_i, C_k)$$

□

Proposición 5.5.4 *Sea $A_i \subseteq B_j$. Entonces*

1. $Conf(A_iB_j, C_k) = Conf(A_i, C_k)$.

2. $Conf(A_i, B_jC_k) = Conf(A_i, C_k)$.

Demostración: $A_i \subseteq B_j$ quiere decir que $o \in A_i$ implica $o \in B_j$. Luego entonces, $supp(A_i \cap B_j) = supp(A_i)$.

1.
$$Conf(A_iB_j, C_k) = \frac{supp(A_i \cap B_j \cap C_k)}{supp(A_i \cap B_j)} =$$
$$= \frac{supp(A_i \cap C_k)}{supp(A_i)} = Conf(A_i, C_k)$$

2.
$$Conf(A_i, B_jC_k) = \frac{supp(A_i \cap B_j \cap C_k)}{supp(A_i)} =$$
$$= \frac{supp(A_i \cap C_k)}{supp(A_i)} = Conf(A_i, C_k)$$

□

Proposición 5.5.5 *Sea* $C_k \subseteq B_j$. *Entonces* $Conf(A_i, C_kB_j) = Conf(A_i, C_k)$.

Demostración: $C_k \subseteq B_j$ significa que $o \in C_k$ implica $o \in B_j$. Y así, $supp(C_k \cap B_j) = supp(C_k)$ y

$$Conf(A_i, C_kB_j) = \frac{supp(A_i \cap B_j \cap C_k)}{supp(A_i)} =$$
$$= \frac{supp(A_i \cap C_k)}{supp(A_i)} = Conf(A_i, C_k)$$

□

Proposición 5.5.6 *Sea* $A_i \subseteq B_j$. *Entonces* $CF(A_iB_j, C_k) = CF(A_i, C_k)$.

Demostración: Trivial por la proposición 5.5.4, la confianza no cambia y el consecuente es el mismo. □

Proposición 5.5.7 *Sea* $C_k \subseteq B_j$. *Se cumple que* $CF(A_i, C_kB_j) = CF(A_i, C_k)$.

Demostración: Trivial ya que en este caso $supp(C_k \cap B_j) = supp(C_k)$ y, por la proposición 5.5.4, la confianza no cambia. □

Proposición 5.5.8 *Dados los atributos X_A, X_B y X_C correspondientes a las particiones A, B y C, $Conf(X_A, X_BX_C) \leq Conf(X_A, X_C)$.*

Demostración: Como se puede ver en [Blanco et al., 2000], la confianza de una dependencia aproximada es la suma ponderada de la confianza de las reglas que conforman su modelo, y todos los pesos son positivos. Así que de la proposición 5.5.3 se extrae que $Conf(X_A, X_BX_C) \leq Conf(X_A, X_C)$. □

Proposición 5.5.9 *Sea $A \Rrightarrow B$. Entonces*

1. $Conf(X_AX_B, X_C) = Conf(X_A, X_C)$.
2. $Conf(X_A, X_BX_C) = Conf(X_A, X_C)$.

Demostración: Similar a la de la proposición 5.5.8 pero haciendo referencia ahora a la proposición 5.5.4. □

Proposición 5.5.10 *Sea $C \Rrightarrow B$. Luego $Conf(X_A, X_BX_C) = Conf(X_A, X_C)$.*

Demostración: Análoga a la proposición 5.5.8 pero haciendo referencia a la proposición 5.5.5. □

Proposición 5.5.11 *Sea $A \Rrightarrow B$. $CF(X_AX_B, X_C) = CF(X_A, X_C)$.*

Demostración: De acuerdo con la proposición 5.5.9 sabemos que la confianza no cambia. Lo mismo ocurre con el soporte del consecuente, así que el factor de certeza tampoco cambia. □

Proposición 5.5.12 *Sea $C \Rrightarrow B$. $CF(X_A, X_BX_C) = CF(X_A, X_C)$.*

Demostración: Similar al caso de la proposición 5.5.11 pero haciendo referencia a la proposición 5.5.10. □

5.6. Análisis de correspondencias difusas

En problemas reales, es muy común encontrarse con situaciones en las que es necesario definir particiones sobre un cierto conjunto de datos, estableciendo cierto grado de incertidumbre sobre ellas. El problema de partida es el de relacionar un conjunto de objetos con un conjunto de clases (una partición), lo que normalmente se conoce como clasificación.

Supongamos que tratamos de encajar un conjunto de síntomas en un conjunto de enfermedades, con el objetivo de definir la base de un sistema de ayuda a la decisión en un entorno médico. Será muy común que ciertos síntomas puedan venir asociados a varias enfermedades simultáneamente. Podemos enriquecer la semántica del modelo estableciendo los grados en que dichos síntomas se asocian a las enfermedades.

Otro ejemplo de este tipo de situaciones lo tenemos en el reconocimiento de texto. Si ahora los objetos son muestras de letras escritas a mano, podemos querer clasificar tales muestras de acuerdo con su semejanza con las letras reales, estableciendo dicha semejanza como un grado entre 0 y 1. Incluso habrá muestras que puedan parecerse a más de una letra.

Para conseguir tales conjuntos de particiones se suelen aplicar herramientas de agrupamiento (o *clustering*) difuso. Algunos estudios interesantes sobre el tema que se pueden encontrar en la literatura son los presentados en [Backer, 1975, Bezdek y Pal, 1992, López de Mantaras y Valverde, 1988]. Por otro lado, en [Dubes y Jain, 1988, López de Mantaras y Valverde, 1988] encontramos definidos algunos algoritmos basados en la técnica de las k-medias, mediante los que es posible obtener particiones difusas sobre un conjunto de datos. Por último, en [Vila et al, 1999] se describe otro algoritmo para obtener eficientemente el conjunto de particiones difusas.

Partiendo de lo anteriormente expuesto, proponemos la siguiente definición para el análisis de correspondencias entre particiones difusas.

5.6.1. Planteamiento del problema

A continuación describiremos formalmente nuestro problema desde el punto de vista más general. Retomando la notación que hemos venido usando hasta ahora, sea $O = \{o_1, \ldots o_n\}$ un conjunto de objetos finito. Sobre el mismo

Tabla 5.3: Tabla $T'_{\mathcal{A}'\mathcal{B}'}$.

	$\tilde{A}_1$	...	$\tilde{A}_p$	$\tilde{B}_1$	...	$\tilde{B}_q$
o_1	$\mu_{\tilde{A}_1}(o_1)$	...	$\mu_{\tilde{A}_p}(o_1)$	$\mu_{\tilde{B}_1}(o_1)$	...	$\mu_{\tilde{B}_q}(o_1)$
o_2	$\mu_{\tilde{A}_1}(o_2)$	...	$\mu_{\tilde{A}_p}(o_2)$	$\mu_{\tilde{B}_1}(o_2)$	...	$\mu_{\tilde{B}_q}(o_2)$
o_3	$\mu_{\tilde{A}_1}(o_3)$	...	$\mu_{\tilde{A}_p}(o_3)$	$\mu_{\tilde{B}_1}(o_3)$	...	$\mu_{\tilde{B}_q}(o_3)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

definiremos dos particiones difusas, $\mathcal{A}' = \{\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_p\}$ y $\mathcal{B}' = \{\tilde{B}_1, \tilde{B}_2, \ldots, \tilde{B}_q\}$.

Sean $o \in O$, $\tilde{A}_i \in \mathcal{A}'$, $\tilde{B}_j \in \mathcal{B}'$. Llamaremos $\mu_{\tilde{A}_i}(o)$ (respectivamente, $\mu_{\tilde{B}_j}(o)$) al grado en que o pertenece a la clase $\tilde{A}_i$ (respectivamente, $\tilde{B}_j$). Cada una de las clases ha de contener al menos un elemento, esto es, $\tilde{A}_i, \tilde{B}_j \neq \emptyset$. Además, todo objeto ha de pertenecer al menos a una clase de cada partición, es decir,

$$\forall o \in O, \exists \tilde{P}_i \in \mathcal{P}'/\tilde{t}_o[\mathcal{P}'] = \tilde{P}_i \text{ y } \mu_{\tilde{t}_o[X_{\mathcal{P}'}]} > 0$$

Dado que estamos trabajando con particiones de carácter difuso, podemos relajar las condiciones originales por las cuales las clases dentro de una misma partición habían de ser disjuntas, es decir, ahora puede ocurrir que $\tilde{A}_{i_1} \cap \tilde{A}_{i_2} \neq \emptyset$ $\forall i_1, i_2 \in \{1, \ldots, p\}$ y $\tilde{B}_{j_1} \cap \tilde{B}_{j_2} \neq \emptyset$ $\forall j_1, j_2 \in \{1, \ldots, q\}$.

Tampoco vamos a considerar, en principio, el hecho de que las particiones deban estar normalizadas, es decir, $\exists \tilde{P} \in \mathcal{P}', \exists o \in O/\tilde{t}_o[\mathcal{P}'] = \tilde{P}$ y $\mu_{\tilde{t}_o[\mathcal{P}']} = 1$. Como trabajo futuro, nos planteamos estudiar si dicha restricción es necesaria o no.

Para ilustrar nuestra definición, usaremos una representación como la que aparece en la tabla 5.3, en la que vemos cómo es posible usar una tabla transaccional, $T'_{\mathcal{A}'\mathcal{B}'}$, para llevar a la práctica nuestra propuesta. Asociaremos cada objeto a una transacción, por lo que obtenemos que $|r'_{\mathcal{A}'\mathcal{B}'}| = |O|$.

Estamos en disposición de definir el siguiente tipo de posible correspondencia entre las particiones difusas $\mathcal{A}'$ y $\mathcal{B}'$:

Definición 5.6.1 Correspondencia difusa local. *Sean $\tilde{A}_i \in \mathcal{A}'$ y $\tilde{B}_j \in \mathcal{B}'$. Diremos que existe una correspondencia difusa local de $\tilde{A}_i$ hacia $\tilde{B}_j$, notada*

como $\tilde{A}_i \Rightarrow \tilde{B}_j$*, si* $\tilde{A}_i \subseteq \tilde{B}_j$*, es decir, si* $\forall o \in O$*,* $\mu_{\tilde{A}_i(o)} \leq \mu_{\tilde{B}_j(o)}$.

De forma análoga a lo que ocurre en el caso "crisp", podemos obtener correspondencias locales difusas en términos de reglas de asociación difusas (las definidas en el apartado 3.3.2).

5.6.2. Obtención de otros tipos de correspondencias difusas

El siguiente objetivo que podemos plantearnos es el de ampliar la metodología propuesta para abarcar también el problema del análisis de correspondencias parciales y globales entre particiones difusas.

Este problema plantea algunos inconvenientes. Hemos de trabajar a nivel de particiones, en lugar de a nivel de clases, como en el caso anterior. Para ello sería necesario definir un grado de pertenencia a la partición, llamémosle $\mu_{\mathcal{A}'}(o)$, para la partición $\mathcal{A}'$ y un cierto objeto o. Dicha medida habría de calcularse como una agregación de los grados de pertenencia de o a cada una de las clases $\tilde{A}_i$ de $\mathcal{A}'$.

Si bien resulta muy sencillo, con las herramientas que tenemos a nuestra disposición, establecer una relación de orden (por ejemplo, $\leq$) entre clases de distintas particiones, no lo es tanto establecerla entre las propias particiones. Sería preciso, en este caso, definir una relación de orden vectorial.

La dimensionalidad que alcanza el problema hace que éste, en principio, parezca bastante complejo tanto desde el punto de vista teórico como práctico. Por este motivo, estudiaremos con más detenimiento este problema en futuros trabajos, mientras que en la presente memoria restringiremos el modelo al caso particular de considerar el que un objeto únicamente pueda hallarse en una clase de cada partición, aquélla a la que pertenezca con mayor grado.

Formalizaremos el modelo restringido, representando las particiones $\mathcal{A}'$ y $\mathcal{B}'$ por medio de una tabla, similar a la utilizada en nuestra propuesta de análisis de correspondencias para el caso "crisp". Sea $r'_{\mathcal{A}'\mathcal{B}'}$ esta tabla, una relación difusa en el siguiente sentido: cada fila (tupla) y columna (atributo) de $r'_{\mathcal{A}'\mathcal{B}'}$ vendrán asociadas a un objeto y a una partición, respectivamente, a lo que habremos de añadir un cierto grado de cumplimiento. Asumiremos también que $|r'_{\mathcal{A}'\mathcal{B}'}| = |O|$.

Notaremos por $\tilde{t}_o$ a la tupla asociada al objeto o, y $X_{\mathcal{A}'}$ como el atributo

Tabla 5.4: Tabla $r'_{\mathcal{A}'\mathcal{B}'}$.

Objeto	**tupla**	$X_{\mathcal{A}'}$	$X_{\mathcal{B}'}$
o_1	$\tilde{t}_{o_1}$	$\tilde{A}_{i1}, \mu_{\mathcal{A}'}(o_1)$	$\tilde{B}_{j1}, \mu_{\mathcal{B}'}(o_1)$
o_2	$\tilde{t}_{o_2}$	$\tilde{A}_{i2}, \mu_{\mathcal{A}'}(o_2)$	$\tilde{B}_{j2}, \mu_{\mathcal{B}'}(o_2)$
o_3	$\tilde{t}_{o_3}$	$\tilde{A}_{i3}, \mu_{\mathcal{A}'}(o_3)$	$\tilde{B}_{j3}, \mu_{\mathcal{B}'}(o_3)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

asociado a la partición $\mathcal{A}'$. El valor correspondiente a la tupla $\tilde{t}_o$ para el atributo $X_{\mathcal{A}'}$, $\tilde{t}_o[X_{\mathcal{A}'}]$, será la clase para o de acuerdo con la partición $\mathcal{A}'$, es decir, $\tilde{t}_o[X_{\mathcal{A}'}] \in \mathcal{A}'$. Denotaremos por $\mu_{\tilde{t}_o[X_{\mathcal{A}'}]}$ el grado en el que el objeto o se halla en la clase $\tilde{t}_o[X_{\mathcal{A}'}]$. Abusando de la notación, expresaremos esto mismo por medio de $\mu_{\mathcal{A}'}(o)$. Mostramos un ejemplo de la estructura de la relación difusa utilizada en la tabla 5.4.

Sobre este modelo, podemos definir las siguientes correspondencias difusas.

Definición 5.6.2 Correspondencia difusa parcial. *Diremos que existe una correspondencia difusa parcial de $\mathcal{A}'$ hacia $\mathcal{B}'$, que notaremos por $\mathcal{A}' \Rrightarrow \mathcal{B}'$, si $\forall \tilde{A}_i \in \mathcal{A}'$ $\exists \tilde{B}_j \in \mathcal{B}'$ tal que $\tilde{A}_i \subseteq \tilde{B}_j$, lo cual expresaremos como, si $\forall o \in O / \tilde{t}_o[\mathcal{A}'] = \tilde{A}_i$ implica que $\tilde{t}_o[\mathcal{B}'] = \tilde{B}_j$ y $\mu_{\tilde{t}_o[\mathcal{A}']} \dot{\leq} \mu_{\tilde{t}_o[\mathcal{B}']}$.*

$\dot{\leq}$ define una relación de orden vectorial que, en nuestro caso particular, se corresponde con una relación de orden clásica.

Definición 5.6.3 Correspondencia difusa global. *Existe una correspondencia difusa global entre $\mathcal{A}'$ y $\mathcal{B}'$, denotada por $\mathcal{A}' \equiv \mathcal{B}'$, cuando ocurre que $\mathcal{A}' \Rrightarrow \mathcal{B}'$ y $\mathcal{B}' \Rrightarrow \mathcal{A}'$.*

Como ocurriera con las correspondencias *"crisp"*, las correspondencias parciales y globales asocian particiones, en este caso difusas, mientras que las correspondencias locales hacen lo mismo entre clases de dichas particiones. Adicionalmente, encontramos la ventaja de que, en el caso de trabajar con particiones difusas, hasta donde alcanza nuestro conocimiento no hemos hallado su equivalente en la Estadística tradicional, con lo cual podemos considerar que estamos ante una nueva herramienta.

En el siguiente apartado, dedicado a la descripción de algunos casos prácticos, mostraremos uno en particular a través del cual ejemplificar la aplicación del análisis de correspondencias sobre particiones de carácter difuso.

5.7. Casos prácticos

De acuerdo con nuestro planteamiento, la extracción de correspondencias entre clases puede verse como la fase inicial de un caso especial de fusión de conocimiento, comúnmente denominado mezcla de bases de datos (*database merging*). Específicamente, la búsqueda de correspondencias globales entre clasificaciones puede aplicarse al problema de integración de esquemas [Cholvy y Moral, 2001] basado en la identificación de correspondencias entre atributos [Li y Clifton, 2000]. En este caso, la idea es la de determinar si un conjunto de atributos tiene la misma semántica y contiene la misma información, aunque expresada de forma distinta.

Por otro lado, el análisis de correspondencias parciales puede verse como un problema de ajuste de campos (*field matching*) [Monge y Elkan, 1996, Cholvy y Moral, 2001]. En este caso, se trata de encontrar valores de atributo que juegan el mismo papel en la base de datos. Este problema se asocia con el de encontrar correspondencias entre atributos, desde el momento en que una correspondencia entre atributos puede verse como un conjunto de correspondencias entre valores.

Con el objetivo de probar la bondad del modelo, lo hemos aplicado sobre un caso real, consistente en un conjunto de resultados, obtenidos en un trabajo previo, de un primer análisis exploratorio sobre información acerca del cultivo del olivar en la provincia de Granada. El proyecto original se financió con fondos FEDER[1], y tenía como objetivo principal construir un modelo de ayuda a la decisión para empresarios agrícolas. De cara a conseguir esto, se realizaron una serie de encuestas que recogían información sobre localización geográfica, manejo del cultivo, producción y datos de suelo de un conjunto de fincas. Remitimos al lector a [Serrano et al., 2001] para una descripción más detallada de las encuestas y del proceso seguido para realizarlas.

En este apartado, nuestra intención es la de dejar claras las propiedades

[1]Proyecto FEDER 1FD97-0244-C03-2.

más interesantes de las reglas de asociación y las dependencias aproximadas aplicadas a nuestro problema y, por otro lado, descubrir cómo difieren los criterios de clasificación de los tres tipos distintos de información que se han usado para describir el componente suelo de cultivo y que se describen a continuación. Esta segunda parte es la más interesante de cara a los usuarios finales, que en este caso son expertos edafólogos y agricultores.

5.7.1. Descripción del problema

El suelo de cultivo es un elemento natural, que se genera a través de la interacción entre diferentes características ambientales, como el clima, el material geológico, relieve, vegetación y, en mayor medida, la actuación del hombre [FAO, 1998]. Desde la perspectiva de los usuarios (agricultores), el suelo es el medio por excelencia de producción para el cultivo, de olivar en el caso que nos ocupa. De esta forma, el así llamado "Conocimiento de usuario" está esencialmente basado en criterios utilitarios. Los sistemas taxonómicos, como el introducido en [FAO, 1968], que constituye la base para el Mapa 1:200.000 de Unidades de Suelo de la provincia de Granada [Pérez-Pujalte y Prieto, 1980], tratan de compensar el vacío existente entre la visión del suelo como elemento natural o como medio de producción. Por esta razón, se han de añadir un conjunto de parámetros relativos al uso agrario, basados en características utilitarias, al sistema taxonómico de la FAO junto con los criterios genéticos ya existentes. De esta forma obtendríamos el llamado "Conocimiento taxonómico-cartográfico" o "Conocimiento de mapa de suelo". Por último, hemos considerado un tercer tipo de conocimiento, "Conocimiento experto", obtenido a partir de los estudios de los expertos edafólogos consultados. Este conocimiento tiene un carácter local y se basa esencialmente en características de la génesis del suelo. Las tablas C.51, C.52, C.53 y C.54 (páginas 282–285) muestran los tres tipos de clasificaciones de acuerdo a los usuarios, los mapas de suelo y los expertos, respectivamente.

5.7.2. Búsqueda de correspondencias locales y parciales

En [Aranda et al., 2002], introdujimos una primera clasificación de las parcelas, basándonos en los atributos relativos al suelo. Se realizó a partir del

conocimiento de usuario, es decir, a partir de las encuestras recogidas. Las parcelas consideradas fueron distribuidas sobre un total de 19 clusters. Llamaremos *grupos* a esta clasificación. Previamente, ya contábamos con una clasificación científica, presentada en [Pérez-Pujalte y Prieto, 1980], que se basaba en las claves propuestas en [FAO, 1968]. En ésta, se definían un total de 21 tipos de suelo, denominados *unidades de mapas de suelos*, pero de los que sólo 19 eran apropiados para el cultivo del olivar. Llamaremos *codunida* a esta otra clasificación. Finalmente, con objeto de completar los tipos de conocimiento sobre el suelo, necesitamos considerar un tercer tipo, definido con anterioridad como conocimiento experto. Llamaremos *exp13* a la partición obtenida a partir de esta clasificación, en la que se dan un total de 13 clases de suelo.

Más tarde, las clasificaciones consideradas se vieron modificadas para reducir la granularidad presente en las mismas, y por ende el número de clases,

- Por un lado, las 21 clases originales de la clasificación científica se reagruparon en 6, atendiendo a las semejanzas en la estructura del suelo (regosoles cálcicos, regosoles dístricos, etc.). Representaremos esta nueva clasificación mediante la variable *grunida*.
- Con respecto a la clasificación definida por el usario, obtenida por medio de un clustering jerárquico, se estudió el dendrograma resultante y se consiguió reducir a 4 el número de clases. A esta nueva partición la llamaremos *grupos4*.
- Por último, llamaremos *exp5* a la reducción de 13 a 5 clases en el caso de la partición basada en conocimiento experto.

Como podemos ver, cada clasificación consta de dos niveles jerárquicos, teniendo claramente uno de ellos un mayor número de clases que el otro. En consecuencia, es de experar que se encuentre una fuerte relación entre el nivel original y el reducido. Por ejemplo, las siguientes correspondencias parciales nos lo confirman,

$codunida \Rrightarrow grunida$, CF = 1.0, y a la inversa

$grunida \Rrightarrow codunida$, CF = 0.199,

$exp13 \Rrightarrow exp5$, CF = 1.0, pero

$exp5 \Rrightarrow exp13$, CF = 0.16, y

$grupos \Rrightarrow grupos4$, CF = 1.0, junto con

$grupos4 \Rrightarrow grupos$, CF = 0.10

La aplicación de dependencias aproximadas nos permite hallar correspondencias parciales entre estos niveles jerárquicos. Como esperábamos, las dependencias recíprocas tienen un factor de certeza bajo. De acuerdo con la proposición 2.3.1, a partir de este conjunto de dependencias aproximadas podemos inferir tres dependencias funcionales, $codunida \rightarrow grunida$, $exp13 \rightarrow exp5$, y $grupos \rightarrow grupos4$.

Volviendo a los experimentos, nuestro primer paso fue tratar de saber cuán importante era el conocimiento experto sobre génesis del suelo con respecto a la clasificación basada en perfiles dada por la FAO, y necesaria para la elaboración del mapa de suelos 1:200.000. Podemos destacar la siguiente correspondencia parcial,

$grunidad \Rrightarrow exp5$, CF = 0.45

A través de la misma nos es posible establecer un nivel de correspondencia (basado en el factor de certeza) relativamente alto entre conocimiento experto y de mapas de suelos. La interpretación que los expertos nos dan es que los criterios de génesis de suelos dados por el mapa corresponden parcialmente con los criterios expertos, exclusivamente genéticos. Puede observarse cómo esta correspondencia se establece entre dos particiones al mismo nivel jerárquico, con un número similar de clases (ver tablas C.56,C.57).

En el siguiente ejemplo podemos ver cómo se ve incrementado el factor de certeza cuando ascendemos por el nivel jerárquico en el antecedente pero dejamos constante el consecuente. Esto se debe a la dependencia funcional existente entre particiones entre las que existe una relación jerárquica,

$codunidad \Rrightarrow exp5$, CF = 0.80

Por el contrario, las siguientes dependencias aproximadas,

$exp5 \Rrightarrow grupos4$, CF = -0.04, y

$exp13 \Rrightarrow grupos4$, CF = -0.05

nos muestran que cuando el factor de certeza es muy bajo, este efecto transitivo no puede apenas apreciarse.

No se encontraron correspondencias parciales ni globales entre los conocimientos de usuario y experto ni entre los de usuario y mapas de suelos. Era de esperar, en boca de los expertos, debido a que se trabajó sobre datos de una provincia ambientalmente compleja como es Granada, con múltiples y posibles combinaciones entre variables ambientales y factores genéticos. Esto puede deberse a la particularidad del cultivo, perfectamente adaptado a los diferentes entornos mediterráneos. Por tanto, resulta difícil relacionar la predicción de idoneidad del suelo con los criterios cartográficos de la FAO [Sys et al., 1991], o los puramente genéticos.

Pero aún así podemos destacar algunos resultados locales. Si recordamos la definición de correspondencias, el hecho de que no pudieran hallarse correspondencias de tipo parcial o global no nos debía hacer descartar el que pudieran existir correspondencias a un nivel más local, como es el caso. He aquí una ventaja más del modelo propuesto. Por ejemplo,

$$[grunida = clase_4] \Rightarrow [grupos4 = clase_1],\ \text{CF} = 1.0,$$
$$[grunida = clase_6] \Rightarrow [grupos4 = clase_1],\ \text{CF} = 1.0$$

Las anteriores reglas de asociación constituyen un ejemplo de correspondencias locales entre estas clases. Cabe destacar la fuerte dependencia que aparece entre *grunida* y *grupos4*, aunque sólo en sea unos pocos casos y en una única dirección. En *grupos4*, la clase 1 corresponde a suelos pardos y de baja pendiente. La interpretación que se le podría dar a esto es que ambos tipos de suelos (*Xerosoles* para la clase 4 y *Luvisoles* para la clase 6) se hallan incluídos en el mismo grupo (de acuerdo con la proposición 5.3.4).

A pesar del hecho de que el número de correspondencias parciales encontradas fue muy reducido, en tales casos deben destacarse las correspondencias locales asociadas entre valores de las particiones, como las siguientes,

$$[grunida = clase_2] \Rightarrow [exp5 = clase_2],\ \text{CF} = 0.95,$$

$$[exp5 = clase_2] \Rightarrow [grunida = clase_2],\ \text{CF} = 1.0,$$
$$[codunida = clase_3] \Rightarrow [exp5 = clase_2],\ \text{CF} = 0.95,$$
$$[codunida = clase_3] \Rightarrow [grunida = clase_2],\ \text{CF} = 1.0,$$

En este nuevo ejemplo, la cuarta regla de asociación resulta trivial, dado que conocemos que se cumple la dependencia funcional $codunida \rightarrow grunida$, como ya indicamos más arriba. Por otra parte, la fusión de conocimiento entre el conocimiento experto y de mapas de suelos es total, ya que gracias a los expertos sabemos que los *Fluvisoles* se encuentran siempre localizados en vegas llanas junto a los ríos, fácilmente identificables por medio de criterios expertos. Por tanto, la diagnosis de esta unidad es muy certera sin necesidad de recurrir a los criterios analíticos de la FAO.

Otro interesante ejemplo que nos muestra como una unidad de mapas de suelo amplia implica a una clase experta es el siguiente,

$$[grunida = clase_1] \Rightarrow [exp5 = clase_3],\ \text{CF} = 1.0$$

Esto es significativo porque todas las parcelas que aparecen sobre *Litosoles* se asocian a suelos que cualquier edafólogo no experto en los suelos de Granada ni en el cultivo del olivo asociaría inmediatamente a la clase 1 en *exp5*. Suena lógico porque el olivar no puede aparecer sobre *Litosoles* "sensu stricto", ya que se trata de suelos no aptos para el cultivo de una especie vegetal arbórea, como es el olivo. Sin embargo, en esta unidad cartográfica se presentan suelos minoritarios denominados "inclusiones" (en la terminología del experto), que podrian corresponder a la clase 3 en *exp5*. El problema antes comentado de la finalidad cartográfica de la clasificación FAO provoca este tipo de situaciones.

Los dos ejemplos antes comentados nos permitirían establecer un flujo de información entre ambas clasificaciones. Sin embargo, la situación se complica en el caso de las clases numéricamente más importantes de ambas clasificaciones. Los *Regosoles* constituyen el 52 % de suelos de olivar en la provincia de Granada. Por su propia naturaleza, son suelos de difícil clasificación, más aún en clima mediterráneo y en sustratos poco compactos y ricos en bases. Muestra de ello son las continuas alteraciones en los criterios clasificatorios en distintas ediciones de la FAO [FAO, 1968, FAO, 1998]. Esta imprecisión o provisionalidad podemos inferirla, por ejemplo, de las siguientes asociaciones:

$$[exp5 = clase_3] \Rightarrow [grunida = clase_3],\ \text{CF} = 0.39$$
$$[grunida = clase_3] \Rightarrow [exp5 = clase_3],\ \text{CF} = 0.67$$
$$[exp13 = clase_4] \Rightarrow [grunida = clase_3],\ \text{CF} = 0.87$$
$$[exp13 = clase_5] \Rightarrow [grunida = clase_3],\ \text{CF} = 0.41$$
$$[exp13 = clase_6] \Rightarrow [grunida = clase_3],\ \text{CF} = 0.65$$
$$[exp13 = clase_7] \Rightarrow [grunida = clase_3],\ \text{CF} = \text{-}0.31$$

Los suelos de la clase 3 (*exp5*) se asocian a los *Regosoles*, sin embargo, no completamente. Dentro de las subunidades de la clase 3 (*exp5*), la subunidad 4 (*exp13*) es la que mejor se asocia al concepto de *Regosoles* en la provincia, y esto es debido a la gran extensión que alcanza en la provincia este tipo de ambiente genético. Estudiando la tabla C.54 (página 285)hallamos más detalles al respecto.

La subunidades 5 y 6 en *exp13* tampoco se asocian mal. Vemos que el carácter, común a la subunidad 4 y 6 (las de mayor certeza), más que el grado de evolución y la pendiente, es el sustrato margoso, deleznable (esto no es de extrañar pues ambas unidades presentan relaciones topográficas denominadas "catenales").

De las subunidades asociadas a otros sustratos geológicos, la de menos evolución y mayor pendiente (clase 5 en *exp13*) es la única que se percibiría con relativa fiabilidad por el experto como *Regosol*, mientras que la de mayor evolución y menor pendiente (clase 7 en *exp13*) no sería clasificada como *Regosol*, como muestra el signo negativo del factor de certeza.

Con respecto a los *Cambisoles*, suelos que representan el 29 % de la superficie de olivar, obtuvimos las siguientes correspondencias locales, por medio de reglas de asociación,

$$[exp5 = clase_4] \Rightarrow [grunida = clase_5],\ \text{CF} = 0.85$$
$$[exp13 = clase_7] \Rightarrow [grunida = clase_5],\ \text{CF} = 0.34$$
$$[exp13 = clase_9] \Rightarrow [grunida = clase_5],\ \text{CF} = 0.34$$

Podemos comprobar como los suelos pertenecientes a la clase 7 (*exp13*) que no se percibían como *Regosoles*, se entienden sin embargo como *Cambisoles*.

Los suelos de otras clases como la clase 4 en *exp5* y la clase 9 en *exp13* se incluyen de forma más o menos ambigua en *Cambisoles*.

A modo de resumen, la fusión del conocimiento del mapa de suelos y experto no es posible a nivel global, sino sólo en algunas categorías que representan un escaso porcentaje del total (aproximadamente un 17 % de las encuestas recogidas), con suelos poco predominantes como *Litosoles* y *Fluvisoles*. En el resto de los casos es sólo moderadamente buena y en la mayor parte las correspondencias son bajas y locales.

Cuando intentamos fusionar conocimiento del mapa de suelos o de experto con el de usuario, los resultados son aún menos dependientes, lo que era predecible dado el criterio de clasificación completamente divergente. Entre el conocimiento del mapa y del usuario podemos destacar las siguientes reglas,

$$[grunida = clase_2] \Rightarrow [grupos4 = clase_1],\ \mathrm{CF} = 0.40$$
$$[codunida = clase_2] \Rightarrow [grupos4 = clase_1],\ \mathrm{CF} = 0.38$$

Los *Fluvisoles* siempre son suelos de este tipo, es decir, muy aptos para el cultivo del olivar ($grupos4 = clase_1$, como puede verse en la tabla C.55). En este punto coinciden ambas clasificaciones, lo que refleja un caso particular del aspecto utilitario de la clasificación FAO.

$$[grupos4 = clase_3] \Rightarrow [grunida = clase_3],\ \mathrm{CF} = 0.28$$
$$[grupos = clase_1 1] \Rightarrow [grunida = clase_3],\ \mathrm{CF} = 0.23$$

Los *Regosoles*, por el contrario, son un grupo heterogéneo que incluyen grupos y subgrupos de usuario, un poco menos aptas para el cultivo del olivo, como por otra parte era de esperar.

Las parcelas encuestadas donde predominan los *Cambisoles*, considerados bastante aptos para el cultivo, se asocian al grupo de suelos más aptos de la clasificación de usuario. La coincidencia es parcial y ambigua, pero sí es posible un cierto nivel de fusión de conocimiento.

Por último, los intentos de fusión entre conocimiento experto y de usuario arrojan resultados muy similares, como en el caso de las siguientes reglas:

$$[exp5 = clase_2] \Rightarrow [grupos4 = clase_1],\ \mathrm{CF} = 0.38$$
$$[exp13 = clase_2] \Rightarrow [grupos4 = clase_1],\ \mathrm{CF} = 0.35$$

Ambas correspondencias locales son buenas, dado que tanto el antecedente como el consecuente son clases de suelos considerados muy apropiados. Por otra parte, se corresponden en todas las características excepto en que la clase 2 de *exp5* es "llana" y la clase 1 de *grupos4* es "moderadamente inclinada" (ver tablas C.55,C.57, en las páginas 286 y 287, respectivamente).

$$[exp5 = clase_3] \Rightarrow [grupos = clase_1], \text{ CF} = \text{-}0.23$$
$$[grupos = clase_3] \Rightarrow [exp5 = clase_3], \text{ CF} = 0.24$$

Ambos subgrupos (clases 1 y 3 en *grupos*) conforman suelos adecuados para el cultivo. Sin embargo, la clase 1, de suelos pardos, tiende a una no asociación con la clase 3 de *exp5*, como nos indica el factor de certeza negativo. El color pardo de un suelo representa un mayor grado de evolución que el color característico de la clase 3 en *exp5*, luego es un carácter importante para fusionar el conocimiento experto y de usuario, en el que coinciden aspectos genéticos y utilitarios.

$$[exp13 = clase_6] \Rightarrow [grupos4 = clase_1], \text{ CF} = 0.41$$

Una vez más, esta regla de asociación (interpretada como una correspondencia local) nos muestra cómo los suelos de la clase 3 en *exp5* (la clase 6 en *exp13* está incluída en la clase 3 de *exp5*), relativamente poco evolucionados, se corresponden con el grupo de usuario con mayor aptitud.

Podemos decir entonces, que el grupo de usuario de mayor aptitud tiende a asociarse a suelos que el experto clasificaría como de poca evolución, carbonatados y espesos, lo que concuerda con la peculiaridad de la zona estudiada, donde los suelos poco evolucionados son *Fluvisoles* o *Regosoles* sobre materiales margosos, blandos y con una elevada capacidad de retención de agua (limitante en el estío) [Pérez-Pujalte y Prieto, 1980]. No obtante, los relativamente bajos factores de certeza no permiten establecer con fiabilidad estas afirmaciones.

5.7.3. Correspondencias globales en la práctica

Por último, daremos un vistazo a un ejemplo en el que casi es posible definir una correspondencia global. Hemos de considerar unas nuevas clasificaciones,

- *kmcluster*. Partiendo de la clasificación de usuario original, *grupos*, redujimos el número de clases por medio de un algoritmo de k-medias, obteniendo un total de 4 clases.
- *hcluster*. Aquí aplicamos un clustering jerárquico, el método del vecino más cercano, usando una distancia euclídea y una normalización de las medias por puntuaciones Z, sobre la clasificación original. También fue 4 el número de clusters obtenidos.

Como hemos visto, ambos casos parten de la misma clasificación pero aplican distintos métodos. Las dependencias aproximadas que pudieron obtenerse fueron las siguientes,

$kmcluster \Rrightarrow hcluster$, CF = 0.827, y

$hcluster \Rrightarrow kmcluster$, CF = 0.893

Para este ejemplo, el factor de certeza es muy cercano a 1 para ambas dependencias. Considerando un cierto grado de relajación, podríamos afirmar que $kmcluster \equiv hcluster$, o lo que es lo mismo, tenemos una correspondencia global entre ambos atributos, con un factor de certeza $CF = 0{,}827$, de acuerdo a lo expuesto en la definición 5.4.1.

Estudiando las correspondencias locales, podemos encontrar las siguientes reglas de asociación,

$[hcluster = clase_1] \Rightarrow [kmcluster = clase_1]$, CF 1.0,

$[kmcluster = clase_1] \Rightarrow [hcluster = clase_1]$, CF 0.945,

$[hcluster = clase_2] \Rightarrow [kmcluster = clase_2]$, CF 0.859,

$[kmcluster = clase_2] \Rightarrow [hcluster = clase_2]$, CF 0.717,

$[hcluster = clase_3] \Rightarrow [kmcluster = clase_3]$, CF 0.947,

$[kmcluster = clase_3] \Rightarrow [hcluster = clase_3]$, CF 0.947,

$[hcluster = clase_4] \Rightarrow [kmcluster = clase_4]$, CF 0.611,

$[kmcluster = clase_4] \Rightarrow [hcluster = clase_4]$, CF 0.904

Podemos apreciar como existe una correspondencia casi perfecta entre ambas clasificaciones, debido a que para su obtención se aplicaron métodos muy similares (k-medias y k-vecinos más cercanos).

5.7.4. Correspondencias entre más de dos conjuntos de particiones

Dado que contamos con más de dos particiones en nuestro ejemplo, podríamos plantearnos el estudiar algunas de las propiedades propuestas en la sección 5.5. Por ejemplo, echemos un vistazo a las siguientes dependencias aproximadas,

$$codunida, grunida \Rrightarrow exp5, \text{ CF} = 0.80, \text{ y}$$

$$codunida \Rrightarrow exp5, \text{ CF} = 0.80,$$

Dado que $codunida \subseteq grunida$, podemos concluir en que se cumple la proposición 5.5.6.

Otro ejemplo es el resultante de estudiar estas otras dependencias aproximadas,

$$exp5 \Rrightarrow codunida, grunida, \text{ CF} = 0.10, \text{ y}$$

$$exp5 \Rrightarrow grunida, \text{ CF} = 0.10,$$

De acuerdo con la proposición 5.5.7, podemos afirmar que, como era de esperar, $codunida \subseteq grunida$.

5.7.5. Estudio de un caso mediante análisis de correspondencias clásico

Sin atrevernos a confirmar qué tipo de análisis de correspondencias es el mejor, lo expuesto hasta ahora en este capítulo debe verse como una metodología alternativa al análisis clásico, expresada en términos de minería de datos. Una futura aportación sobre esta memoria podría ser un estudio comparativo entre ambos tipos de análisis de correspondencias.

Por ahora, nos limitaremos a mostrar que ambas técnicas son equivalentes en los casos que hemos estudiado. Nos centraremos en el estudio de las correspondencias existentes entre las particiones *codunida* y *grunida*, siendo la segunda una generalización de la primera, como ya vimos.

Analizando las correspondencias existentes mediante el método clásico, podemos comprobar que existe una correspondencia muy clara. Para ello sólo

Tabla 5.5: Examen de los puntos de fila[a]

	Masa	Puntuación en la dimensión		Inercia	Contribución				
		1	2		De los puntos a la inercia de la dimensión		De la dimensión a la inercia del punto		
grunida					1	2	1	2	Total
1	.053	.000	-1.394	.864	.000	.102	.000	.118	.118
2	.115	2.131	.717	.885	.521	.059	.589	.067	.656
3	.522	-.625	.717	.478	.204	.268	.425	.561	.986
4	.005	3.541	.717	.995	.060	.002	.060	.002	.063
5	.287	.000	-1.394	.698	.000	.558	.000	.800	.800
6	.019	3.353	.717	.981	.215	.010	.219	.010	.229
Total activo	1.000			4.901	1.000	1.000			
a Normalización Simétrica									

hay que comparar las puntuaciones en la dimensión (fila o columna) de las tablas 5.5 y 5.7. En las celdas de dichas tablas hallamos las coordenadas asociadas a cada valor, que luego se verán representadas en el mapa perceptual (figura 5.1). Por ejemplo, entre los puntos $codunida = 1$ y $grunida = 1$ existe una correspondencia perfecta. De igual forma, el punto $grunida = 5$ coincidiría con los puntos $codunida = 13, 15, 16, 17, 19$ y 20. Estas coincidencias se aprecian considerablemente en el mapa perceptual, donde puede verse que la correspondencia entre valores es perfecta para este caso en particular.

Esta misma información podemos obtenerla por medio de la metodología propuesta, estudiando las correspondencias a nivel parcial y local, expresadas en términos de dependencias aproximadas y reglas de asociación, respectivamente. Como ya indicáramos al inicio del apartado de experimentación, existe una dependencia funcional (que se traduce en una correspondencia parcial total) de *codunida* a *grunida* (con CF = 1). Además, obtenemos las correspondencias locales exactas que aparecen en la tabla 5.6, donde el valor de la fila determina al de la columna con el factor de certeza que aparece en la intersección de las mismas (ver las tablas C.53 y C.56 en la página 284 para la codificación de los valores),

Con estos resultados mostramos que, para este caso concreto, nuestra metodología para el análisis de correspondencias resulta equivalente al análisis clásico de correspondencias, siempre que éste se realice sobre particiones "crisp", claro. En el siguiente apartado veremos cómo trabajar sobre un ejemplo en el que se cuenta con particiones de carácter difuso.

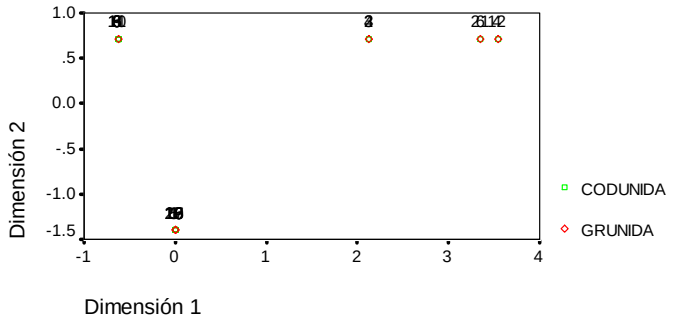


Figura 5.1: Mapa perceptual con los puntos de columna y de fila

Tabla 5.6: Correspondencias locales entre *codunida* (filas) y *grunida* (columnas)

	1	3	4	5	6	8	9	10	11	12	13	15	16	17	19	20	21
1	1																
2		1	1														
3				1	1	1	1	1	1								
4										1							
5											1	1	1	1	1	1	
6																	1

Tabla 5.7: Examen de los puntos de columna[a]

	Masa	Puntuación en la dimensión		Inercia	Contribución				
		1	2		De los puntos a la inercia de la dimensión		De la dimensión a la inercia del punto		
codunida					1	2	1	2	Total
1	.057	.000	-1.394	.861	.000	.112	.000	.130	.130
2	.000	.	.	.	.	.	.	.	.
3	.110	2.131	.717	.848	.500	.057	.589	.067	.656
4	.005	2.131	.717	.037	.022	.002	.589	.067	.656
5	.172	-.625	.717	.158	.067	.089	.425	.561	.986
6	.033	-.625	.717	.031	.013	.017	.425	.561	.986
7	.000	.	.	.	.	.	.	.	.
8	.024	-.625	.717	.022	.009	.012	.425	.561	.986
9	.177	-.625	.717	.162	.069	.091	.425	.561	.986
10	.057	-.625	.717	.053	.022	.030	.425	.561	.986
11	.057	-.625	.717	.053	.022	.030	.425	.561	.986
12	.005	3.541	.717	.995	.060	.002	.060	.002	.063
13	.067	.000	-1.394	.166	.000	.130	.000	.783	.783
14	.000	.	.	.	.	.	.	.	.
15	.110	.000	-1.394	.273	.000	.214	.000	.783	.783
16	.043	.000	-1.394	.107	.000	.084	.000	.783	.783
17	.019	.000	-1.394	.048	.000	.037	.000	.783	.783
18	.000	.	.	.	.	.	.	.	.
19	.010	.000	-1.394	.024	.000	.019	.000	.783	.783
20	.033	.000	-1.394	.083	.000	.065	.000	.783	.783
21	.019	3.353	.717	.981	.215	.010	.219	.010	.229
Total activo	1.000			4.901	1.000	1.000			
a Normalización Simétrica									

5.7.6. Ejemplos sobre particiones difusas

Hasta ahora hemos contemplado el caso en el que deseamos obtener correspondencias entre atributos (en nuestro ejemplo, particiones) "crisp". Por el apartado anterior, hemos visto que tanto el análisis clásico de correspondencias como nuestra propuesta alternativa resultan válidos para resolver el problema de partida.

Sin embargo, pueden darse situaciones, como las descritas anteriormente, en las que los atributos que vayamos a considerar estén afectados por cierta imprecisión o incertidumbre en sus valores. La Estadística clásica no provee, hasta donde nosotros sabemos, de ninguna herramienta con la que abordar dicho problema. En este apartado vamos a aplicar las definiciones aportadas en la sección 5.6 sobre un problema cuyos datos son de carácter difuso.

Vamos a reutilizar los datos que definíamos al inicio del apartado 5.7. Originalmente, el conjunto de clases definido por el atributo *grupos* se obtuvo por medio de un clustering en el que combinamos métodos de agrupación jerárquicos y no jerárquicos. Contábamos con 211 casos de partida, de los cuales inicialmente sólo podían usarse 157, al tener el resto algún valor nulo entre los atributos que lo describían, algo que las técnicas clásicas no admiten.

En una primera etapa se intentó determinar el número de conglomerados mediante un método jerárquico, sin indicar el número de éstos con el que debía de detenerse el algoritmo. Se utilizó un clustering jerárquico mediante el método del vecino más cercano, usando una distancia euclídea y estandarizando medias mediante puntuaciones Z.

Como resultado de este agrupamiento, obtuvimos 20 clases para los 157 casos. A continuación, este conjunto de clases se refinó aplicando un algoritmo basado en las k medias difusas, como los descritos en [Dubes y Jain, 1988, López de Mantaras y Valverde, 1988]. Tras esto, obtuvimos un nuevo conjunto de 19 clases, que ahora ya sí podía usarse para clasificar los 211 casos iniciales. Todo este proceso, junto con los resultados obtenidos, se detallan en [Aranda et al., 2002].

El resultado del clustering difuso es una matriz de 211 casos por 19 posibles clases, en donde cada celda contiene un grado de pertenencia del caso concreto a la clase correspondiente. Dicha matriz fue la que usamos para definir nuestra

primera partición difusa, a la que llamaremos $grupos_d$.

La otra partición sobre la que trataremos de hallar correspondencia seguirá siendo la definida por el atributo *codunida* [Pérez-Pujalte y Prieto, 1980].

Aplicando el algoritmo de extracción de dependencias aproximadas difusas propuesto en el capítulo 4 (algoritmo B.11, en la página 245), obtenemos el siguiente par de correspondencias parciales difusas,

$codunida \Rrightarrow grupos_d, CF$ 0,12, y

$grupos_d \Rrightarrow codunida, CF$ 0,18

Aunque ambos factores de certeza tengan un valor muy bajo, merece la pena destacar el hecho de que en el caso "crisp" los valores para el factor de certeza eran mucho más bajos, por lo que, en cierto modo, ganamos algo más en cuanto a contenido informativo, ya que es posible obtener más dependencias por encima del umbral mínimo prefijado. De estas dos correspondencias parciales difusas podríamos extraer una correspondencia global difusa, $codunida \equiv grupos_d$, con un CF = 0.12, aunque este valor es tan bajo que quizá no mereciera la pena en este caso.

Estudiando las correspondencias difusas a un nivel local, extraemos las mismas relaciones que se podían obtener en el caso crisp, junto con algunas nuevas que se hubieran perdido de otro modo. En la tabla 5.8, recogemos algunos ejemplos de estas correspondencias parciales difusas, obtenidas mediante reglas de asociación difusas. El contenido de la tabla se interpreta de la siguiente forma: cada una de las filas es una clase de la partición *codunida*, cada una de las columnas es una clase de la partición $grupos_d$, y la intersección entre ellas nos proporciona el factor de certeza de la correspondencia de la fila hacia la columna. Se ha restringido la búsqueda a aquellas correspondencias con $CF > 0{,}68$.

A la vista de los resultados, parece claro para este ejemplo concreto que el análisis de correspondencias difusas proporciona más juego de cara a la interpretación de los resultados, al ser éstos menos restrictivos. Como futura aportación, queda pendiente el estudio de sus aplicaciones sobre otros problemas existentes.

Tabla 5.8: Correspondencias locales difusas entre las clases de *codunida* (filas) y $grupos_d$ (columnas)

	1	2	3	4	5	8	10	11	15	16
3	0.76	0.75	0.70			0.68		0.69		
5			0.69							
6	0.82	0.83	0.72			0.72		0.73	0.75	
8		0.87								
11								0.71		
15			0.74							
16		0.76	0.73						0.72	
20			0.70	0.76	0.76		0.70			0.72

5.8. Discusión

El estudio de correspondencias locales, parciales y globales es especialmente útil para ciertos procesos de fusión de información. En este capítulo hemos propuesto una medida de la precisión de dichas correspondencias basándonos en el factor de certeza de reglas de asociación y dependencias aproximadas, por medio de una representación de particiones sobre tablas relacionales. Además, proporcionamos algunas propiedades que relacionan los valores de certeza con posibles situaciones entre correspondencias.

Una aplicación donde resulta interesante el uso de técnicas de minería de datos para la fusión de conocimiento está en el campo de las ciencias del suelo. Ello puede deberse a la heterogeneidad e imprecisión asociadas a la diferentes fuentes de conocimiento sobre suelos. Las posibilidades de interpretación del conocimiento de usuario en términos científicos con el objeto de ser empleado en la toma de decisiones son de lo más interesante, gracias a la facilidad de obtención y a la gran extensión de esta fuente de conocimiento.

Hemos presentado un caso particular en el que el análisis de correspondencias clásico y nuestra propuesta nos proporcionan resultados análogos. Las técnicas de data mining obtienen unos resultados expresados mediante reglas, más fácilmente interpretables, en opinión de los expertos consultados, para un usuario no experimentado con el estudio de mapas perceptuales.

También proponemos una primera aproximación al problema de manejar particiones de carácter difuso sobre un conjunto de datos. En este caso, no tenemos constancia de ninguna técnica estadística semejante, por lo que nos hallamos ante una nueva herramienta.

Por otra parte, pensamos que el algoritmo de búsqueda de dependencias aproximadas y sus modelos, propuesto en [Blanco et al., 2000] será de suma utilidad para la comparación de varias particiones al mismo tiempo. En especial, el caso de la comparación de jerarquías nos resulta particularmente interesante, y será el tema principal de futuros trabajos.

6. Aspectos prácticos

En este capítulo presentamos la aplicación FuzzyQueries 2+, un gestor de consultas flexibles que además cuenta con un amplio conjunto de herramientas para operar con los resultados de dichas consultas, entre las que nos interesan especialmente, de cara a los objetivos de esta memoria, las destinadas a la minería de datos. Junto con la principal ventaja que supone la integración de distintas herramientas, contamos con un módulo de fácil ampliación y mejora.

Éste es también un capítulo eminentemente práctico. Detallaremos los aspectos de implementación de los resultados teóricos que se han ido desarrollando a lo largo de esta memoria, y que se ven consumados en una aplicación software completamente funcional, como la ya citada FuzzyQueries 2+.

FuzzyQueries 2+ continúa con la filosofía iniciada por FQ. FQ (Fuzzy Queries) fue la primera versión de una aplicación cliente F-SQL capaz de conectarse a un Sistema de Bases de Datos Relacionales Difusas basado en GEFRED [Medina et al, 1994, Medina, 1994], y operar sobre tablas con contenido difuso. Implementado en Visual Basic por José Galindo Gómez para

complementar su trabajo de Tesis Doctoral [Galindo, 1999], ha supuesto la base sobre la que se apoya el actual FuzzyQueries 2+.

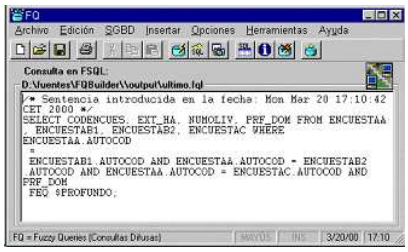


Figura 6.1: Fuzzy Queries original

Hemos de volver a incidir en el hecho de que este capítulo se complementa con el contenido del apéndice A, donde incluimos un completo manual de usuario de la aplicación que aquí presentamos, y que esperamos que el lector encuentre de utilidad.

6.1. Elección del lenguaje de programación

El primer aspecto a tener en cuenta antes de entrar en más detalle es el de considerar la posibilidad de que la aplicación final que implementamos estuviese escrita en un lenguaje de programación con una gran potencia y una amplia perspectiva de futuro, vistas sus aplicaciones sobre multitud de medios, desde su aplicación en electrodomésticos hasta la más conocida, en Internet. Concretamente, nos estamos refiriendo al lenguaje Java, desarrollado originalmente por Sun Microsystems, aunque en la actualidad podamos contar con otras varias implementaciones como la proporcionada por Microsoft, Visual J++. Sin embargo, ésta última no deja de presentar ciertos problemas de compatibilidad con la versión original de Sun, por lo que nos decidimos a usar el entorno de programación Java Development Kit (JDK) ofrecido por Sun, contando en añadidura con la ventaja que supone su libre distribución vía Internet.

Java es un lenguaje relativamente reciente, aunque eso no ha sido obstáculo para su rápida extensión y aplicación en multitud de campos, debido a su versatilidad y potencia. Parte de una sintaxis muy similar a la del lenguaje C

y, más en concreto, a su versión orientada a objetos, C++. De ahí que resulte muy sencillo y rápido de aprender tanto por expertos programadores como por neófitos en la materia.

Como principal inconveniente de este lenguaje, encontramos el hecho de que aún no sea posible generar archivos ejecutables a partir del código Java, ya que presenta una arquitectura basada en el establecimiento de una máquina virtual (Java Virtual Machine, JVM), sobre la que se ejecutan los programas compilados o "byte-codes". Este hecho provoca el que, por lo general, el mismo programa escrito y compilado en Java se ejecute más lentamente que si se hubiera programado en C++, debido a que lo que realmente hace Java es interpretar los archivos compilados. Pero lo que por un lado es una desventaja, desde otro punto de vista resulta una gran baza a su favor, ya que el código Java generado es totalmente portable de una máquina a otra, aún teniendo sistemas operativos diferentes, con más que dispongan de una máquina virtual Java implementada sobre ellos. Esta importante característica es una de las más conocidas de este lenguaje y su principal ventaja frente a otros, motivo por el cual se ha extendido tanto y tan rápidamente en los últimos años.

No obstante, el anterior problema promete ser subsanado por la propia Sun, quien pretende sacar al mercado en un futuro no muy lejano una utilidad denominada *java2c*, con la que, para casos puntuales donde fuera necesario, se permitiría traducir el código compilado Java a un código objeto C, que pudiera enlazarse posteriormente y generar así un archivo ejecutable. Con esto perderíamos la característica anterior de portabilidad, frente a la ganancia en velocidad que proporcionaría la ejecución directa del programa sobre un sistema operativo concreto.

Java ha evolucionado mucho desde sus primeras versiones, debido por un lado a la necesidad de aplicarlo sobre nuevos campos y, por otro, a la facilidad para extender el lenguaje de acuerdo a dichas necesidades, junto con la mejora de la tecnología actual en ordenadores. La versión del lenguaje Java que se utilizó para la implementación de los resultados prácticos presentados en esta memoria fue la del JDK 1.3.1.02. Aunque por entonces ya se encontraba disponible una versión más reciente, el JDK 1.4.0.01, los cambios introducidos en ésta no influían especialmente en la mejora del resultado final. En particular, la aplicación puede seguir ejecutándose sin problemas con la versión

actual del lenguaje. Destacaremos los paquetes para la conexión con sistemas de bases de datos, por medio de JDBC (Java Data Base Connectivity), con un papel crucial para nuestras necesidades de acceso a la información, así como la nueva y mejorada API (Application Programming Interface, Interfaz de Programación de Aplicaciones) de Java basada en tecnología Swing, que proporciona un amplio abanico de componentes gráficos con los que presentar al usuario final los resultados de una forma más amena y eficiente.

Para más información sobre el lenguaje, acceso a sus últimas versiones, a la posibilidad de consultar completas guías y manuales de uso, y acceso a interfaces gráficas de usuario orientadas a la programación visual mediante Java, junto con innumerables recursos relacionados con este lenguaje de programación, recomendamos la visita a la página oficial de Java, `http://java.sun.com`.

6.2. Módulo de consulta

FuzzyQueries 2+ nace originalmente como una aplicación cliente para la confección y ejecución de consultas basadas en SQL (o en una de sus extensiones difusas, FSQL, del que incluimos una breve introducción en el apartado A.1). Con el paso del tiempo, se hace necesaria la inclusión de nuevos módulos en la aplicación para abordar nuevos problemas, como se verá a continuación.

6.2.1. Prototipo original

La primera versión del prototipo aparece en [Galindo, 1999], está programada en Visual Basic y permite escribir sentencias basadas en los lenguajes SQL y FSQL [Galindo et al, 1998], actuando también como herramienta cliente para permitir al usuario comunicarse con el Sistema Gestor de Bases de Datos Relacionales (SGBDR), que en nuestro caso concreto nos viene suministrado por Oracle®.

Sobre dicha herramienta se implementa un asistente, FQBuilder, cuya ventana principal se muestra en la figura 6.2, para la construcción de las consultas flexibles, que permite acceder al catálogo de la base de datos para obtener información útil sobre las tablas, sus columnas y el tipo de éstas, prestando especial atención al caso de elementos difusos. Además, cuenta con la opción de mostrar, para las columnas de tipo difuso, una representación gráfica de la

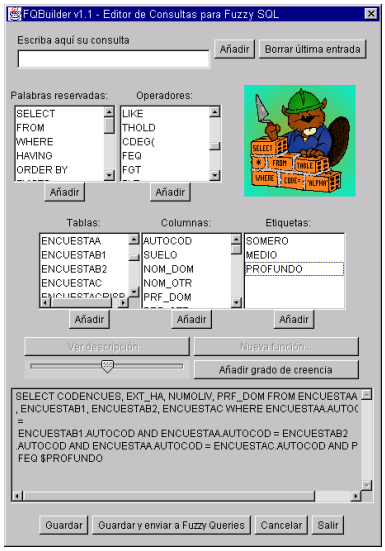


Figura 6.2: Ventana principal del FQBuilder original

distribución de posibilidad asociada a la columna. Mediante este mismo interfaz, el usuario puede definir sus propias etiquetas lingüísticas a través de una distribución trapezoidal.

La principal desventaja de esta arquitectura estribaba en que las dos aplicaciones estaban escritas en lenguajes de programación distintos, y la interacción entre ellas era lenta e incompleta. De ahí que finalmente se optara por integrar ambas herramientas en una única aplicación escrita en Java, que pasó a denominarse FuzzyQueries 2. Respetando el interfaz gráfico del original, e incorporando las ventajas del asistente de consultas a la construcción de las mismas y la comunicación con el SGBDR, también permite la adición de nuevas herramientas. A esto unimos uno de los puntos fuertes del lenguaje Java, al que ya hemos hecho mención con anterioridad, y que no es otro que la portabilidad, esto es, la posibilidad de poder ejecutar el mismo código en distintas plataformas, tanto software (distintos sistemas operativos, como Windows o Linux) como hardware (distintas arquitecturas, como un PC o un Macintosh).

6.2.2. Operaciones disponibles

Posteriormente, la aplicación ha ido ampliándose con algunas funcionalidades que hemos considerado interesantes. En el momento de redactar esta memoria, las capacidades de FuzzyQueries 2 son, entre otras, las siguientes:

- **Edición y modificación de consultas, flexibles o no, sobre el servidor (F)SQL.** A tal fin, la aplicación cuenta con las posibilidades básicas de cualquier editor de texto como, por ejemplo, las herramientas Cortar, Copiar y Pegar, así como la necesaria posibilidad de abrir y guardar los archivos de texto con los que trabajemos. Otra función más específica de cara a la confección de consultas esla inclusión de una lista de palabras reservadas del lenguaje (F)SQL, a las que el usuario puede acceder navegando a través de los menús de la herramienta.

- **Posibilidad de mostrar la traducción de la sentencia FSQL a SQL.** Dado que, en realidad, el servidor FSQL actúa como una capa sobre el servidor SQL (recordemos la figura 3.5, página 72), toda sentencia escrita con la sintaxis de FSQL tiene una correspondiente traducción al lenguaje SQL. Esta función, de carácter didáctico, tiene como objetivo mostrar dicha traducción. También se facilita al usuario la posibilidad de almacenarla en disco.

- **Posibilidad de almacenar los resultados de la consulta en distintos formatos (texto simple, objeto Java, tabla HTML).** Una vez que el usuario ha redactado la consulta y se ha conectado al servidor, puede ejecutar dicha consulta y recoger en una tabla los resultados de la misma. Con objeto de resultar lo más portable posible, nuestra herramienta brinda la posibilidad de almacenar dichos resultados en distintos formatos. Como un archivo de texto simple, editable por cualquier procesador de textos, como un objeto Java, que puede ser recuperado más tarde por la misma aplicación (para, por ejemplo, realizar algún tipo de operación sobre dichos resultados sin tener que volver a conectarse a la base de datos y ejecutar de nuevo la consulta), o bien como un archivo HTML, que podremos visualizar en un navegador web cualquiera.

Realmente, y exceptuando la posibilidad de almacenar y posteriormente recuperar un conjunto de resultados, estas características ya aparecían en el Fuzzy Queries original. Pero sobre éstas se fueron añadiendo nuevas funcionalidades como las siguientes:

- **Consulta interactiva del catálogo FMB de la base de datos: tablas y vistas, columnas y etiquetas lingüísticas asociadas.** Esta extensión constituye un vestigio de la aplicación FQBuilder que, como comentábamos en el apartado anterior, fue la primera extensión sobre Fuzzy Queries, una aplicación por separado para facilitar la confección de la consulta por medio de accesos al catálogo del sistema y, en caso de trabajar sobre un servidor FSQL, a la base de metaconocimiento difuso (FMB, ver apartado 3.2.2), donde el usuario puede encontrar información sobre qué tipo de columnas difusas se encuentran accesibles, y sobre si existen o no etiquetas lingüísticas definidas sobre sus dominios. Resulta una función bastante didáctica, en el sentido de que aporta más información al usuario sobre la consulta que desea realizar. Cuando se desarrolló FuzzyQueries 2 en su primera versión, esta posibilidad de consulta del catálogo fue plenamente integrada en la aplicación.

- **Posibilidad de visualizar la correspondencia entre etiquetas lingüísticas y distribuciones de posibilidad asociadas.** Una etiqueta lingüística, tal y como se definió en el apartado 3.1.1, suele estar asociada a una distribución de posibilidad definida sobre un dominio, normalmente numérico. El uso de etiquetas lingüísticas resulta bastante interesante de cara a proporcionar una información más acesible y comprensible para el usuario, ya que suelen venir expresadas en lenguaje natural. Como operación suplementaria a la anterior, en el momento de consultar las etiquetas lingüísticas definidas sobre el dominio de una columna difusa, si éste es numérico, la aplicación muestra en una ventana aparte el conjunto de distribuciones asociadas a las etiquetas. Asimismo, es posible crear una nueva distribución trapezoidal sobre el dominio numérico subyacente, usando el ratón para definir sus parámetros. Si el usuario así lo desea, FuzzyQueries 2 permite que éste deje constancia de la nueva distribución trapezoidal definida, asignándole una etiqueta

lingüística y almacenándola en el catálogo del sistema, desde donde podrá reutilizarla en una sesión posterior. Hemos de observar, no obstante, que, para que la operación tenga éxito, el usuario ha de contar con los privilegios necesarios para realizarla sin problemas.

- **Cálculo de diversas medidas sobre los resultados de la consulta.** Por ejemplo, podemos calcular una estimación de la imprecisión del atributo, mediante la medida not-null, que encontramos definida en [Marín et al, 2003]. Mucho más interesante es la posibilida de obtener, gráficamente, algunos de los cardinales difusos más comunes, entre los que se encuentran los definidos en el apartado 3.1.5, junto con la propuesta de un cardinal lingüístico difuso que aparece también en [Marín et al, 2003].

- **Fuzzy Deductor.** Por último, en lo que respecta al tratamiento de imprecisión e incertidumbre en una base de datos, la herramienta incluye un módulo con la capacidad de manejar predicados difusos intensivos, del que podemos encontrar más información sobre sus fundamentos teóricos en [Blanco, 2001].

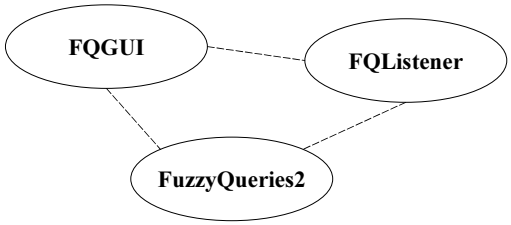
La aplicación ya ha sido usada como herramienta de consulta flexible en proyectos como el descrito en [Serrano et al., 2001], donde se aborda el modelado en una base de datos difusa, definida siguiendo el modelo GEFRED [Medina et al, 1994], en la que encontramos información de diversos tipos sobre parcelas agrícolas dedicadas al cultivo del olivar.

6.2.3. Detalles de implementación

La principal característica con la que dotamos a nuestro módulo fue la de facilitar futuras extensiones y ampliaciones en previsión de, por ejemplo, cambios en la implementación del servidor difuso de bases de datos subyacente (basado en la arquitectura FIRST, que ya comentamos con anterioridad, en el apartado 3.2.2), o bien por la inclusión de nuevas funcionalidades y operaciones sobre los datos. Con este objetivo en mente, y uniendo a ello las ventajas ya comentadas del lenguaje de programación Java, dicho lenguaje fue el elegido para llevar a la práctica nuestro modelo.

Entrando más en detalle, la aplicación software en sí se puede descomponer en tres submódulos bien diferenciados, tales como los que se muestran en la figura 6.3.

Figura 6.3: Submódulos principales en FuzzyQueries 2+



- Por un lado, está el módulo principal, FuzzyQueries2, que contiene los subprogramas básicos para el establecimiento de la conexión con el servidor de bases de datos, el intercambio de mensajes entre éste y la aplicación, así como toda la parte de gestión de sistema, lo que incluye tanto el manejo de archivos como la detección y solución, en la medida de lo posible, de los posibles errores que puedan aparecer en tiempo de ejecución, sean éstos debidos al sistema o bien al usuario. En definitiva, el módulo FuzzyQueries2 comprende el núcleo de la aplicación, con todo lo necesario para un buen funcionamiento de ésta.

- Pero el funcionamiento de la aplicación puede verse enormemente mejorado con la inclusión de una interfaz gráfica de usuario que proporcione un entorno visual y amigable y que facilite la interacción de una persona con la aplicación software. De este apartado se encarga el módulo FQGUI, que contiene toda la parte de programación del interfaz gráfico de la aplicación. En este módulo, hemos intentado aprovechar al máximo las potentes funcionalidades que ofrece el paquete Java Swing, cuya especialidad es la implementación de todos aquellos componentes gráficos necesarios en una aplicación software, desde componentes de contenido (ventanas, marcos y recuadros) hasta componentes interactivos (botones, campos de texto, diales, etc.). La portabilidad y la potencia que nos aporta el lenguaje Java nos facilita el que, sin necesidad de tocar

una sola línea de código de este módulo, podamos lanzar la aplicación en diferentes entornos gráficos sin que ésta pierda un ápice de funcionalidad. La propia máquina virtual Java se encargará de dotar de la apariencia habitual del sistema a todos los componentes definidos en el módulo FQGUI.

- Aún es necesario un tercer módulo que sirva de puente entre los dos anteriores. Si FuzzyQueries2 proporciona la funcionalidad de la aplicación y FQGUI la apariencia e interactividad de la misma, FQListener se apoya en la potente y versátil filosofía de gestión de eventos de Java para establecer las conexiones entre las acciones del usuario y las funciones del programa. FQListener implementa varias *interfaces* de Java para captar los eventos generados por pulsaciones de teclado, movimientos de ratón, etc. por parte del usuario, a través de la interfaz gráfica proporcionada por el módulo FQGUI, y los convierte en llamadas a los submódulos correspondientes en FuzzyQueries2.

Así pues, de cara a añadir nuevas funcionalidades a las ya existentes en nuestra aplicación, el programador correspondiente tan sólo tendría que delimitar qué aspectos habría de incluir en cada uno de los módulos comentados y obrar en consecuencia. Así, los submódulos específicos de implementación habrían de ser incluidos en FuzzyQueries2. En caso de necesitar apoyo gráfico para su puesta en marcha, debería añadir la definición de las estructuras necesarias en el módulo FQGUI. Y, por último, debería enlazar unos con otros mediante la detección de los eventos correspondientes y las subsiguientes operaciones necesarias, que tendrían su lugar en el módulo de gestión de eventos, FQListener.

6.3. Nueva extensión para minería de datos

Se hacía patente la necesidad de una herramienta de minería de datos a través de la cual implementar los resultados teóricos que aquí hemos presentado. Con dicho objetivo en mente, desarrollamos un nuevo módulo que fue integrado en la aplicación anteriormente descrita, que pasó a denominarse FuzzyQueries 2+. El nuevo módulo cuenta con la capacidad de ex-

traer reglas de asociación [Agrawal et al., 1993] y dependencias aproximadas [Blanco et al., 2000] a partir de la información recogida en una base de datos, con el añadido de que también está preparado para extraer las correspondientes extensiones al caso difuso de las herramientas mencionadas, esto es, reglas de asociación difusas [Delgado et al., 2003a] y dependencias aproximadas difusas (como las definidas en el capítulo 4).

6.3.1. Precedentes

Este módulo de minería de datos tiene su origen en una aplicación independiente, Data Miner, desarrollada para el estudio de las dependencias aproximadas, y de cómo era posible extraer las mismas a partir de la misma metodología empleada en la obtención de reglas de asociación (como vimos en el apartado 2.3.6). A través del interfaz de dicha aplicación, el usuario podía configurar las opciones de entrada, estableciendo los umbrales mínimos para las medidas de soporte y confianza (de las que ya hablamos en el apartado 2.3.4), seleccionando el conjunto de atributos que se quería involucrar en la extracción de reglas o dependencias, etc.

Figura 6.4: Algunas ventanas del módulo original de minería de datos

Dado que en más de una ocasión se puede usar el resultado de una consulta como entrada para un proceso de extracción de conocimiento, optamos por integrar esta aplicación como un módulo más de la herramienta FuzzyQueries 2+. El producto final constituye algo más que eso, puesto que el módulo Data Miner permite ejecutarse tanto dentro de FuzzyQueries 2+ como fuera de la misma, de forma independiente, en lo que constituye una puesta al día de la versión original.

6.3.2. Características y funcionalidades

La herramienta Fuzzy Data Miner, ejecutada de forma independiente con respecto a FuzzyQueries 2+, nos permite elegir como fuente alternativa de datos de origen un archivo de texto, siempre y cuando se mantengan unas sencillas reglas de sintaxis según las cuales el archivo de texto debe de seguir una estructura concreta. Dicha estructura no es otra que la de incluir una línea de texto por cada fila de datos. Dentro de cada línea, separaremos los atributos mediante tabuladores.

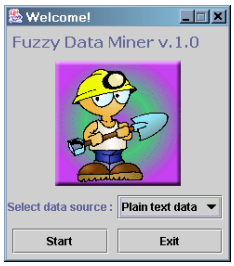


Figura 6.5: Ventana de entrada a FDMiner

En definitiva, el módulo Fuzzy Data Miner cuenta con las siguientes características:

- **Extracción de Reglas de Asociación, Dependencias Aproximadas, Reglas de Asociación Difusas y Dependencias Aproximadas Difusas.** Esto se explicará con más detalle en el apéndice con el manual de usuario de la aplicación software.
- **Fase de configuración inicial.** Fuzzy Data Miner ofrece la posibilidad de configurar los siguientes parámetros de entrada: Mínimos umbrales de soporte, confianza y factor de certeza y máximo número de iteraciones (lo cual redundará en el tamaño final de las reglas).
- **Origen de los datos.** La aplicación permite usar como origen de los datos el resultado de una consulta (F)SQL, para lo que habremos de estar conectados a un servidor de bases de datos, o bien utilizar un archivo de texto con formato (separando las columnas por tabuladores). Esta última opción puede resultarnos más interesante para experimentos

rápidos o sencillos, que no requieran la complejidad que puede acarrear la conexión a una base de datos.

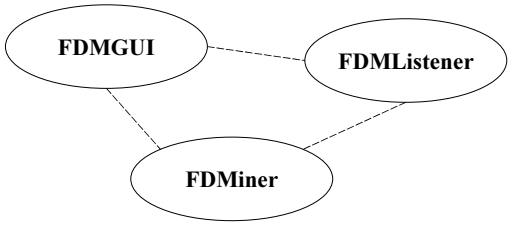
- **Reusabilidad de los resultados.** La herramienta nos ofrece la posibilidad de almacenar en disco informes con los resultados parciales (conjuntos de ítemsets frecuentes) y los finales (reglas o dependencias, según el caso). Al igual que ocurre con el módulo de consulta, el usuario puede elegir el formato en el que quiere almacenar los resultados, bien sea éste un simple archivo de texto, o bien un archivo de texto en un formato especial en el que se delimitan los elementos por tabulaciones, facilitando la importación de dicho archivo desde otra aplicación. Por ejemplo, podemos usar alguna aplicación comercial de gestión de bases de datos para generar una base de reglas, más fácil de manejar que un simple listado.

Para más detalles sobre la herramienta, se incluye un apéndice al final de la memoria con un completo manual de usuario de la misma.

6.3.3. Implementación

A nivel general, el módulo Fuzzy Data Miner se compone de tres submódulos, de acuerdo a las funciones específicas de cada uno de ellos:

Figura 6.6: Submódulos principales en Fuzzy Data Miner



- En primer lugar, toda la parte de programación se encuentra en el módulo principal, FDMiner. Desde aquí se controla todo lo relativo a la configuración y ejecución de los algoritmos de minería de datos, así como también los aspectos más internos de ejecución, aquellos relativos a la

gestión de archivos y estructuras de memoria. Se encarga también de la comunicación con los objetos mediante los cuales representamos el tipo de conocimiento que podemos extraer, y, por supuesto, de la comunicación con el servidor de bases de datos, en caso de que precisemos conectarnos con uno.

- Para facilitar una cómoda interacción con el usuario, también hemos incluido en un módulo aparte todo lo relativo con la definición y la gestión de la interfaz gráfica de usuario. Dicho módulo es el que hemos denominado FDMGUI. También en este caso aprovechamos la potencia proporcionada por el paquete Swing de Java.

- Por último, con la función de enlazar los dos módulos anteriores, gestionando los eventos generados desde la interfaz gráfica y haciendo que el módulo principal opere en consecuencia, hemos de contar de nuevo con un tercer módulo que herede los métodos proporcionados por Java para la gestión de eventos tales como pulsaciones de teclado, acciones sobre ventanas, etc. El módulo FDMListener es el encargado de llevar a cabo todo esto.

La figura 6.6 representa la interacción entre estos tres módulos, y salta a la vista la analogía existente entre esta figura y la figura 6.3, ya que hemos aplicado la misma filosofía de funcionamiento que para la aplicación principal. En un primer momento, se pensó en integrar completamente el módulo de minería de datos dentro de la aplicación, pero con vistas a permitir la ejecución por separado de éste, se optó finalmente por separarlo también a nivel de programación. En vista de los buenos resultados que nos proporcionó la división en tres submódulos de la aplicación original, decidimos continuar con esa misma filosofía a la hora de programar el módulo de minería de datos.

6.3.3.1. Tipos de conocimiento extraíble

Siguiendo la filosofía expuesta por Java y por la Programación Orientada a Objetos en general, establecemos la funcionalidad de una aplicación software apoyándonos en clases y objetos. No vamos a entrar más en detalle en

cuestiones teóricas referidas al paradigma de la Programación Orientada a Objetos, sino que nos reduciremos a describir las clases que hemos implementado para representar los distintos tipos de conocimiento que podemos extraer por medio de nuestra aplicación.

Por un lado, hemos definido una clase abstracta, DMResultado, de la cual heredan las siguientes subclases:

- **AssocRule.** Por medio de esta clase implementamos la extracción de Reglas de Asociación clásicas. Definimos la representación de las reglas mediante clases auxiliares, ItemSet y Regla (mediante la cual representamos una regla en sí), mientras que la clase AssocRule contiene aquellos métodos necesarios para el cálculo de los ítemsets frecuentes. Dado que nuestra elección, por el momento, fue la de implementar únicamente el algoritmo Apriori [Agrawal y Srikant, 1995], podemos implementar éste dentro de esta clase.

- **ApproxDep.** El funcionamiento y la filosofía son básicamente los mismos que en la clase anterior, salvo que el proceso seguido para la extracción de ítemsets frecuentes va ahora orientado a la obtención de dependencias aproximadas. Para ello, se ha de extender el algoritmo básico (que en nuestro caso se trata de Apriori) tal y como se expone en [Blanco et al., 2000].

- **FAssocRule.** Esta clase hereda las funcionalidades de la clase AssocRule para posibilitar la extracción de reglas de asociación difusas, tal y como se definen en [Delgado et al., 2003a]. Tal y como allí se expone, hemos de extender la clase ItemSet para manejar soportes en transacciones difusas, definiendo una nueva clase, FItemSet, donde substituimos la variable que usamos como contador por un vector de contadores para cada α-corte.

- **FApproxDep.** Por último, mediante esta clase llevamos a la práctica el algoritmo que definimos en el capítulo 4 para extraer ítemsets frecuentes en el proceso de obtención de dependencias aproximadas difusas. Nos apoyamos en estructuras previamente definidas como algunas de las ya comentadas, al tiempo que hemos de implementar algunas nuevas clases

para manejar nuevas estructuras, como por ejemplo, las clases de equivalencia que ya comentamos que usamos para establecer las relaciones de similitud entre valores de atributos.

6.3.3.2. Tipos de fuentes de datos

Una de las características introducidas en este módulo es la posibilidad de trabajar contra un servidor de bases de datos (para lo cual podemos utilizar las funcionalidades que los paquetes JDBC de Java nos ofrecen), o bien trabajar a partir de archivos de texto, para conjuntos de datos pequeños. Antes de comenzar a utilizar la aplicación, elegiremos el tipo de fuente de datos.

- En el caso de trabajar contra una base de datos, deberemos conectarnos al servidor de bases de datos en primer lugar, para después abrir el archivo de texto con la consulta en SQL a partir de la cual trabajaremos. Si nos encontramos usando la aplicación FuzzyQueries 2+, y es desde ella desde donde realizamos la llamada al módulo Fuzzy Data Miner, éste tomará como parámetro de entrada la consulta que se esté editando actualmente en FuzzyQueries 2+. Hemos de hacer algunas distinciones en función del tipo de conocimiento que deseamos extraer, ya que algunos de ellos precisan de estructuras auxiliares que habrán de ser definidas sobre la misma base de datos.

  - En el caso de trabajar con reglas de asociación y dependencias aproximadas, no hay ningún problema. Podemos trabajar a partir de una tabla o de una vista, que puede construirse como resultado de una consulta. No se precisa de ninguna restricción en especial sobre el formato de los datos.
  - La extracción de reglas de asociación difusas, tal y como fueron descritas en el apartado 3.3.2, sí que exige, por el momento, que la tabla o vista con la que trabajemos esté constituida por columnas exclusivamente numéricas, donde cada columna vendrá asociada a un ítem, y cada celda al grado de cumplimiento de un ítem en una transacción.

- Por último, el caso más complicado es el de la extracción de dependencias aproximadas difusas. Con objeto de mantener el formato expuesto en la definición de las mismas, que aparece en el apartado 4.4, la tabla o vista se compondrá de pares de columnas, representando la primera de ellas el atributo a considerar, y el grado de cumplimiento de un valor de dicho atributo en una tupla, la segunda columna. No obstante, con objeto de eliminar información trivial, si se omite la segunda columna, se dará por hecho que el grado de cumplimiento será siempre 1 para cada tupla de la relación.

  Aparte de esto, hemos de almacenar por separado en otra tabla los valores asociados a las relaciones de similitud. La tabla en cuestión ha de tener la siguiente estructura, en notación SQL,

  ```
  create table nombre(
  attrib varchar2,
  val1 varchar2,
  val2 varchar2,
  degree number(4,2))
  ```

  El usuario, responsable de preparar esta tabla, tiene absoluta libertad para establecer el nombre de la misma (que será un parámetro de entrada más de la aplicación) y la longitud de las tres primeras columnas, alfanuméricas, en virtud de lo que considere necesario. Esta tabla habrá de ser convenientemente rellenada de acuerdo con las relaciones de similitud definidas. Al incluir la columna `attrib`, podemos usar esta misma tabla para representar todas las relaciones de similitud definidas sobre un conjunto de atributos. Por último, para ahorrar espacio, no es necesario incluir aquellos pares de valores entre los que no existe relación (se entiende que, por defecto, si una pareja de valores no aparece en la tabla, el grado de similitud entre ambos es 0), ni tampoco es necesario especificar los casos triviales (es decir, aquellas parejas en las que `val1` y `val2` son el mismo valor, en cuyo caso el grado de similitud es siempre 1).

  Como vimos en el apartado 4.10.2, uno de los factores que más influyen en la eficiencia final del algoritmo es el número de pares

de valores en las relaciones de similitud. Unido a esto, otro factor importante que encarece mucho el coste en tiempo de ejecución del algoritmo es el acceso a la base de datos. Una posible mejora que proponemos es la de, para no incrementar excesivamente el número de accesos a la base de datos, almacenar las relaciones de similitud en memoria principal. Frente a la necesidad de más espacio que esto supone, hemos observado cómo los costos en tiempo se reducen significativamente.

- Por el contrario, para trabajar con archivos de texto, hemos de dar un formato determinado a dichos archivos, así como preparar los archivos necesarios por separado. En un archivo aparte, han de encontrarse los nombres de los atributos, uno por línea. Este primer archivo ha de tener el mismo nombre que el archivo usado como fuente, pero substituyendo la extensión por ".dic". El formato del archivo con los datos dependerá del tipo de conocimiento que queramos extraer. Si bien, en general, se tratará de valores separados por tabulaciones, en una línea por cada transacción o tupla.

  - Para la extracción de reglas de asociación y dependencias aproximadas clásicas, el archivo fuente se compondrá de secuencias de valores separados por tabulaciones. Cada línea del archivo equivaldrá a una transacción o tupla, y todas las líneas han de contener el mismo número de elementos, esto es, el mismo número de columnas. El tipo de éstos, numérico o alfanumérico dependerá del significado que queramos dar a la columna.
  - Para la extracción de reglas de asociación difusas, el formato del archivo fuente ha de ser el mismo que en el caso anterior, pero ahora los elementos han de ser valores reales entre 0 y 1, representando cada uno el grado de pertenencia del ítem a la transacción, tal y como vimos en el apartado 3.3.2.
  - Por último, para extraer dependencias aproximadas difusas seguiremos el mismo formato pero alternando valores de atributo y grados de cumplimiento, siguiendo la representación presentada en el capítulo 4. En este caso, además, hemos de preparar un archivo de

texto adicional para almacenar las relaciones de similitud entre valores. El formato de este archivo será de una línea por cada posible par, indicado así:

*atributo tab valor*1 *tab valor*2 *tab grado*

Este archivo ha de tener el mismo nombre que el archivo fuente, pero con la extensión ".sim". Al igual que comentábamos para el caso de trabajar con una base de datos remota, se entiende que por defecto, si un par de valores no aparece en el archivo, su grado de similitud es 0, por lo que no es necesario incluir tales pares en el archivo. De igual forma, tampoco es necesario incluir los casos triviales, en los que ambos valores son el mismo y el grado de similitud entre ellos, 1.

Como en el caso anterior, las relaciones de similitud se almacenan en memoria al inicio del proceso para disminuir el acceso a disco y tener un acceso más rápido a las mismas.

6.4. Resumen

Este capítulo se ha dedicado a la descripción de una aplicación software completa y de fácil manejo, orientada en principio a la confección y procesamiento de consultas sobre un servidor de bases de datos, admitiendo la posibilidad de trabajar con información afectada por imprecisión o incertidumbre. La arquitectura de la herramienta es lo suficientemente flexible como para permitir añadir nuevos módulos a la misma, aumentando de esa forma sus funcionalidades, e integrando herramientas relacionadas entre sí para dar una mayor comodidad al usuario que precise del uso simultáneo de varias de las mismas.

Concretamente, hemos visto cómo, con vistas a la experimentación necesaria durante la confección de esta memoria, ha sido posible añadir un módulo de minería de datos a la aplicación. El módulo en cuestión admite también la posibilidad de ejecutarse de forma independiente, en caso de que no sea necesario, por ejemplo, conectarse a ningún servidor de bases de datos, ya que además permite trabajar con archivos de texto como origen de los datos.

El módulo de minería de datos cuenta con un conjunto de herramientas basadas en las reglas de asociación, admitiendo además las extensiones al caso difuso de las mismas. El proceso de extracción de conocimiento se ha cuidado para reducir en la medida de lo posible los requerimientos en tiempo de ejecución y espacio de memoria necesarios.

Discutiremos algunas futuras ampliaciones sobre la herramienta en el siguiente capítulo.

7. Conclusiones y líneas futuras

Por último, este capítulo nos servirá para resumir los objetivos que se han logrado y que hemos descrito en la presente memoria. A continuación de los mismos, expondremos un cierto número de líneas futuras que extenderán los resultados obtenidos y aquí presentados.

7.1. Resumen de los resultados obtenidos

Comenzamos el apartado de conclusiones resumiendo y enumerando los objetivos que nos proponíamos resolver y que hemos desarrollado a lo largo de esta memoria.

Definición de Dependencia Aproximada Difusa. Partiendo de dos conceptos clave: la obtención de dependencias aproximadas a partir de reglas de asociación, y la extensión al caso difuso de las citadas reglas de asociación, hemos definido el nuevo concepto de dependencia aproximada difusa. Los ítemsets mediante los cuales obtendremos las dependencias

aproximadas difusas se calculan a su vez por medio de ítemsets de reglas de asociación difusas. Al igual que ocurre en el caso clásico, existen situaciones particulares en las que esto nos resulta de gran ayuda, pudiendo explicar una dependencia aproximada por medio de un conjunto de reglas de asociación difusas. Desarrollamos dicha obtención por medio de una metodología que nos permite extraer dependencias aproximadas en una base de datos relacional difusa de una forma sencilla y eficiente. Este último aspecto ha sido tratado con especial cuidado, minimizando en la medida de lo posible los requisitos en tiempo de ejecución y espacio de memoria empleados durante el cálculo. Prueba de ello es el algoritmo que presentamos en esta memoria.

Las dependencias aproximadas difusas, como la gran mayoría de las herramientas y técnicas extendidas mediante el uso de la lógica difusa, nos permiten enriquecer la idea original al tiempo que nos facilitan el análisis de información afectada de imprecisión o incertidumbre, que de otra forma nos resultaría intratable. Así, aparte de constituir un caso más general de las dependencias aproximadas clásicas, nos permiten trabajar con casos especiales en los que, por ejemplo, nos interesa disminuir la granularidad de un cierto dominio, para lo cual se podrá definir un conjunto de etiquetas entre las que podrá existir cierto grado de parentesco (o, para ser más exactos, de similitud).

Minería de datos aplicada al Análisis de Correspondencias. Con respecto al segundo objetivo que nos proponíamos al comienzo de esta memoria, hemos propuesto una alternativa al análisis de correspondencias estadístico clásico, en términos de reglas de asociación y dependencias aproximadas. A partir de unas ciertas particiones sobre un mismo conjunto de objetos, nos resulta posible estudiar hasta qué punto pueden corresponderse, por medio de una sencilla representación tabular. Mediante dependencias aproximadas, podemos hallar correspondencias entre atributos a un nivel parcial o global (p.e., hasta qué punto pueden solaparse varias particiones obtenidas a partir de diferentes criterios). Por otro lado, y en caso de necesitar entrar en más detalle, las reglas de asociación pueden informarnos sobre las correspondencias que puedan

existir a un nivel local. Nuestra metodología proporciona una alternativa frente al análisis de correspondencias clásico, al contar además con una medida de la precisión de los resultados como es el factor de certeza de reglas de asociación y dependencias aproximadas. Una serie de experimentos realizados a partir de distintas clasificaciones sobre el elemento suelo, que se describen en profundidad dentro de la memoria, nos han servido para ilustrar este enfoque y sus posibilidades.

De forma adicional, permitimos ampliar la metodología existente por medio de la aplicación de dependencias aproximadas difusas al análisis de correspondencias sobre particiones difusas. Dado que hasta donde alcanza nuestro conocimiento no hay precedentes sobre dicha materia, consideramos esto un factor importante y enriquecedor de cara a los resultados.

Estas nuevas metodologías tienen aplicación en la fusión de conocimiento, entre otras áreas. Conocer el grado en que se solapan dos o más particiones de un mismo conjunto de datos puede resultar una información clave a la hora de realizar tareas como la fusión de bases de datos, o lo que se conoce tambien como *database merging*.

Desarrollo de la herramienta FuzzyQueries 2+. Como último objetivo, pero no por ello menos importante, presentamos una herramienta software, desarrollada e implementada durante el transcurso de la investigación que se describe en la memoria. Dicha herramienta estaba ideada inicialmente para facilitar la experimentación y la puesta en práctica de los resultados teóricos obtenidos. Pero consideramos que la funcionalidad de la aplicación resultante va más allá de su mero uso como herramienta para esta memoria.

FuzzyQueries 2+ no es sólo una aplicación de consulta en bases de datos (difusas o no), sino que además nos permite realizar una serie de cálculos y operaciones sobre los resultados de las consultas, destacando la implementación de una herramienta de minería de datos, Fuzzy Data Miner, que puede ser ejecutada como un módulo más de FuzzyQueries 2+, o bien de forma independiente.

La aplicación cuenta con la ventaja que supone la portabilidad entre

sistemas, al estar escrita completamente en lenguaje Java. Además, permanece abierta a futuras extensiones de cara a cubrir nuevas necesidades que vayan surgiendo. A nivel ergonómico, creemos que el interfaz resulta muy fácil de utilizar, por su comodidad y sencillez de diseño, así como también por la posibilidad de presentar documentos de ayuda siempre que el usuario así lo requiera. Permite trabajar con datos procedentes de distintas fuentes y también almacenar los resultados en varios formatos diferentes.

El análisis de correspondencias mediante técnicas minería de datos (restringida al caso "crisp") ha sido aplicado a un problema concreto en el entorno agrícola, cuyos resultados se encuentran en [Aranda et al., 2003]. Por otro lado, una primera versión del prototipo FuzzyQueries 2+ fue presentada en [Blanco et al., 2002]. Por último, el trabajo en el que presentamos la nueva definición de dependencias aproximadas difusas [Berzal et al., 2003] se encuentra actualmente sometido para su publicación en un número especial de Fuzzy Sets and Systems.

7.2. Futuras aportaciones

El trabajo descrito en la presente memoria nos deja un campo abierto sobre el que investigar, extendiendo los resultados previamente resumidos en las conclusiones. A continuación presentamos algunas de las ideas que hemos considerado más interesantes de cara a futuras aportaciones.

Ampliación del concepto de dependencia aproximada difusa. Se pretenden estudiar nuevos campos de aplicación para la técnica definida. Por un lado, en el aspecto teórico, ya hemos visto algunas propuestas, extendiendo el modelo clásico con las funcionalidades que el nuevo modelo nos ofrece. Un ejemplo es el uso de dependencias aproximadas difusas como herramientas de extracción de conocimiento para establecer relaciones entre atributos con dominios imprecisos, algo que ya planteamos en el capítulo 5. El problema de manejar atributos sobre dominios no atómicos no se ha contemplado, debido a la complejidad asociada, y será objeto de un análisis más detallado en el futuro.

Por otro lado, en esta memoria hemos formulado el problema de extracción de dependencias aproximadas difusas en unos términos muy concretos de representación, restringiéndonos al caso en el que cada celda de la relación difusa contenga un par $< valor, grado >$. Otra futura aportación sería estudiar la representación de dicho problema por medio del modelo GEFRED (apartado 3.2.1) para bases de datos relacionales difusas.

Análisis experimental. El algoritmo que se ha implementado es una extensión y adaptación del algoritmo básico para extraer reglas de asociación. Podemos plantearnos la posibilidad de adaptar otro (o varios) de los muchos algoritmos existentes en el campo de la extracción de conocimiento, siempre con el objetivo en mente de optimizar en lo posible el uso de los recursos necesarios para la ejecución.

Optimización en el manejo de las relaciones de similitud. Con respecto a los aspectos de implementación y en alusión al uso de relaciones de similitud en la extracción de dependencias aproximadas, en nuestro enfoque tales relaciones han de ser proporcionadas por el usuario, de acuerdo con los criterios que pueda considerar convenientes, o al conocimiento que éste tenga sobre las posibles relaciones entre los valores del dominio. Pero también podría ser interesante el poder contar con un procedimiento mediante el que se permitiera particionar de manera automática los dominios de los atributos, de forma que con ello se consiguiera optimizar la obtención de determinadas dependencias difusas, sobre las que el usuario podría tener un especial interés. Como una línea futura, proponemos estudiar qué sería necesario para desarrollar un algoritmo con estos objetivos en mente.

Mejora del interfaz de usuario. Aunque éste sea más bien un aspecto propio de la minería de datos en general, cuando el número de reglas y dependencias se haga inmanejable, sería conveniente contar con una interfaz mediante la que se pueda acceder a los resultados de forma sencilla y práctica, aplicando filtros cuando el número de resultados que se desea obtener sea muy inferior al que en verdad se ha obtenido. Algunos

avances al respecto se dirigen hacia las llamadas bases de datos inductivas, donde el objetivo es el de englobar en la misma base de datos la información que resulta de su análisis. Al respecto de este tema, encontramos trabajos como [Meo et al, 1996] o [Boulicaut et al, 1998], donde se propone un operador `MINE RULE` sobre SQL para extraer reglas de asociación de una base de datos. Otro trabajo relacionado con la extensión del lenguaje SQL lo podemos encontrar en [Rasmussen, 1997], donde se introduce el lenguaje SummarySQL, con algunas funcionalidades interesantes sobre medidas de agregación y resumen.

Estudio comparativo de técnicas para análisis de correspondencias. Otro trabajo futuro que consideramos interesante consiste en la realización de un estudio pormenorizado de las posibilidades que nos ofrece la definición del análisis de correspondencias difusas que propusimos al final del capítulo 5. En este caso, no nos es posible comparar la bondad de nuestra metodología con la técnica estadística asociada, al no tener constancia actualmente de la existencia de ninguna herramienta que contemple la posibilidad de trabajar con particiones difusas sobre un conjunto de datos. Es por esta razón que consideramos especialmente interesante nuestra definición, y por lo que abordaremos en futuros trabajos sus aplicaciones potenciales en, por ejemplo, la comparación de particiones obtenidas a partir de un agrupamiento (*clustering*) difuso.

Ampliación del conjunto de herramientas. La aplicación software presentada en la memoria permite, por su propia arquitectura, que las extensiones y las inclusiones de nuevos módulos puedan llevarse a cabo de forma sencilla y eficiente. Algunas de las tareas pendientes son, por ejemplo, la inclusión de operaciones sobre los resultados, como medidas difusas de resumen, como las presentadas en [Blanco et al., 2003], o la comparación con algunos trabajos sobre evaluación de sentencias cuantificadas [Díaz-Hermida et al., 2003, Glöckner, 2003]. Tanto esta futura aportación como la que describimos a continuación podrían englobarse como resultados en el proyecto *Fuzzy-KIM*[1], actualmente en desarrollo y cuya finalidad principal es la del diseño de un sistema de minería de

[1]Proyecto CICYT TIC2002-04021-C02-02

datos con ayuda inteligente basado en técnicas de *Soft Computing*.

Desarrollo de una herramienta multiagente. Otro aspecto de especial interés de cara a la implementación estriba en la posibilidad de generar una aplicación orientada a la minería de datos distribuida. Por medio de varios agentes remotos, se podría mejorar la eficiencia en el cálculo de los ítemsets frecuentes y en la posterior obtención de las reglas de asociación (o dependencias aproximadas en su caso), derivando ambas tareas en distintos agentes coordinados e intercomunicados entre sí, de manera que disminuyeran los requisitos en tiempo y en espacio gracias al reparto de tareas.

Además, integrando el interfaz en una página web, se permitiría a distintos clientes remotos formular peticiones a un servidor de bases de datos desde un simple navegador web. Teniendo en cuenta la rapidez con la que Java se extiende por Internet, donde ya encontramos multitud de aplicaciones *on-line* basadas en este lenguaje, y considerando que ya tenemos implementado el prototipo en dicho lenguaje, resulta cuando menos una interesante propuesta sobre la que debatir en un futuro no muy lejano.

A. FuzzyQueries 2+. Manual de usuario

Dedicamos este apéndice a la presentación de un manual de usuario básico y actualizado de la aplicación FuzzyQueries 2+, que ya fue introducida en el capítulo 6, correspondiente a la parte práctica de los resultados presentados en esta memoria.

A través de este manual se describirán las capacidades y funcionalidades de la herramienta y se expondrán los pasos necesarios para sacar el máximo partido de sus posibilidades. Deseamos que el lector encuentre interesante el contenido de este apéndice y que le resulte de utilidad en el caso de que precise del uso de nuestra aplicación. Así también merece la pena destacarse el hecho de que este mismo apéndice puede hallarse en forma de ayuda online dentro de la misma herramienta, a través de un sencillo navegador HTML inserto en la aplicación.

A.1. El lenguaje FSQL

FSQL [Galindo et al, 1998, Galindo, 1999] extiende el lenguaje estándar de consulta SQL para permitir sentencias con imprecisión, flexibles o difusas. Uno de los componentes de la arquitectura FIRST, como recordamos en el apartado 3.2.2, es el Servidor FSQL, mediante el cual es posible ejecutar las sentencias escritas en dicho lenguaje. Está programado en PL/SQL sobre la plataforma suministrada por Oracle®, y está diseñado para trabajar tanto con bases de datos difusas como con bases de datos tradicionales, lo cual permite una rápida incorporación de las ventajas de las consultas flexibles sobre bases de datos tradicionales preexistentes.

El lenguaje FSQL extiende algunas de las sentencias clásicas de SQL, como es el caso de la sentencia `SELECT` para la consulta flexible, o la sentencia `CREATE TABLE` para la creación de tablas que admitan columnas difusas. Además, ha de introducir algunas nuevas sentencias para manejar y almacenar otras estructuras específicas del caso difuso, como por ejemplo, etiquetas lingüísticas o relaciones de semejanza. Se pueden consultar otros trabajos como [Galindo et al, 1998, Blanco, 2001], para más detalles sobre la sintaxis y el funcionamiento interno del lenguaje.

Tabla A.1: Comparadores difusos en FSQL

Posibilidad	**Necesidad**	**Significado**
`FEQ`	`NFEQ`	Posiblemente/Necesariamente Igual Difuso
`FGT (FGEQ)`	`NFGT (NFGEQ)`	Posiblemente/Necesariamente Mayor (o Igual) Difuso
`FLT (FLEQ)`	`NFLT (NFLEQ)`	Posiblemente/Necesariamente Menor (o Igual) Difuso
`MGT (MLT)`	`NMGT (NMLT)`	Posiblemente/Necesariamente Mucho Mayor (o Menor)

A.1.0.3. Datos almacenados en un SGBDRD

En el modelo FIRST, la información que se maneja se divide en dos categorías: la contenida en la base de datos tradicional y la incluída en la FMB.

Tabla A.2: Constantes difusas usadas en FSQL

Constante difusa	**Significado**
`UNKNOWN`	Valor desconocido, pero el atributo es aplicable.
`UNDEFINED`	El atributo no es aplicable o carece de sentido.
`NULL`	Desconocimiento total (no se sabe si es o no aplicable).
$\$[a, b, c, d]$	Trapecio Difuso ($a \leq b \leq c \leq d$).
$\$label$	Etiqueta Lingüística, asociada a un valor difuso numérico o escalar.
$[n, m]$	Intervalo ($n \leq m$), "Entre n y m".
$\#n$	Valor difuso "Aproximadamente n".
a	Valor numérico clásico (crisp).

Éstos pueden ser de tres tipos:

- **Tipo 1 o FTYPE1:** Se corresponden con los atributos "crisp", esto es, sin imprecisión, pero con la salvedad de que pueden ser utilizados en una consulta flexible. Es posible definir etiquetas lingüísticas sobre su dominio, asociadas a algún valor difuso.

- **Tipo 2 o FTYPE2:** Definidos sobre un dominio ordenado (comúnmente numérico), admiten tanto valores "crisp" como difusos, éstos últimos en forma de distribuciones de posibilidad o asociados a una etiqueta lingüística previamente definida.

- **Tipo 3 o FTYPE3:** Por último, este tipo de atributos corresponde a los definidos sobre un dominio subyacente no ordenado (o escalar), como por ejemplo, "color del pelo". De cara a manejar este tipo de datos, es necesario definir un conjunto de etiquetas como posibles valores, y sobre éstas, una relación de semejanza que nos indique en qué medida se parecen entre sí.

Para operar adecuadamente con valores difusos, FSQL define un nuevo conjunto de comparadores para estos tipos de datos, que se muestra en la tabla A.1. Asímismo, para trabajar más cómodamente con atributos de tipo

1 o 2, FSQL cuenta con un conjunto de constantes como las que aparecen en la tabla A.2.

Por otro lado, en la FMB hemos de almacenar la información relativa a la gestión de la base de datos difusa y sus atributos, también en un formato relacional. Para cada atributo de tipo 1 o 2, se han de almacenar las etiquetas lingüísticas que pueda tener asociadas (y su correspondiente distribución de posibilidad). Para los atributos de tipo 3, necesitamos almacenar las etiquetas lingüísticas y las relaciones de semejanza entre ellas. Todo este proceso se realiza de forma transparente al usuario, a través de las sentencias que conforman el DDL (Data Definition Language, Lenguaje de Definición de Datos) de FSQL.

A.1.1. Consultas flexibles en FSQL

Las consultas flexibles se construyen en FSQL de forma análoga a como se harían en SQL estándar, es decir, usando la misma y ampliamente conocida notación

`SELECT ... FROM ... WHERE ...`,

pero con la salvedad y la ventaja añadida de que ahora es posible utilizar la sintaxis extendida que nos proporciona FSQL para trabajar con expresiones difusas como las mostradas en la tabla A.2, y con los operadores difusos de la tabla A.1. Así mismo, de especial interés es la función `CDEG(*)`, a la que podemos llamar dentro de la consulta flexible para obtener el grado de compatibilidad asociado a las tuplas devueltas como resultado, de acuerdo con las condiciones incluidas en la clausula `WHERE`.

A.2. FuzzyQueries 2+

FuzzyQueries 2+ es una aplicación cliente para F-SQL (la extensión difusa del lenguaje SQL que describimos en el apartado A.1) pensada inicialmente para manejar consultas difusas. Sobre esta funcionalidad básica, se han ido añadiendo nuevas aportaciones para manejar otros tipos de información difusa y facilitar el uso del interfaz gráfico.

FuzzyQueries 2+ está completamente implementado mediante el lenguaje de programación Java, a través del cual se comunica con el Sistema Gestor

de Bases de Datos, proporcionado en nuestro caso por Oracle®, aunque como futura extensión proponemos la posibilidad de usar esta herramienta sobre otros sistemas comerciales. El hecho de estar programado íntegramente en Java facilita la portabilidad de la aplicación para poder ser ejecutada sobre distintos sistemas operativos (p.e., Windows, Linux, etc.) o sobre distintas plataformas (p.e., PC, Macintosh, etc.), siempre y cuando se disponga de una máquina virtual Java para ello. Se puede encontrar más información al respecto en la página oficial de Java, `http://java.sun.com`.

A.2.1. Ventana principal

Tras lanzar el programa, lo primero que veremos será su ventana principal. En la ventana principal de FuzzyQueries 2+ distinguimos tres zonas. La barra de menús, la barra de herramientas, con los comandos más comúnmente utilizados, y el área de texto donde editar la consulta.

Figura A.1: Ventana principal de FuzzyQueries 2+

A.2.1.1. Barra de menús

En la parte superior de la ventana principal (ver figura A.1) nos encontramos con una barra de menús, desde la cual acceder a todas las posibilidades que nos ofrece la aplicación. Los menús se desglosan en los siguientes conjuntos de operaciones, las cuales nos limitaremos a transcribir únicamente, ya que serán debidamente detalladas más adelante:

- Menú “File” (Archivo)

- New query (Nueva consulta).
- Open query (Abrir archivo de consulta).
- Open result (Abrir archivo de resultados).
- Save query (Guardar consulta en disco).
- Save query as (Guardar consulta especificando un nombre de archivo).
- Print query (Imprimir el texto de la consulta).
- Exit (Salir al sistema y terminar la sesión).

▪ Menú "Edit" (Edición)

- Cut (Corta el texto seleccionado).
- Copy (Copia el texto seleccionado).
- Paste (Pega el texto almacenado en el portapapeles a partir de la posición actual del cursor).
- Select All (Selecciona todo el texto).

▪ Menú "DB" (Base de datos)

- Connect to DB (Conexión con la Base de datos).
- Disconnect from DB (Desconexión de la Base de datos).
- Execute SQL Query (Ejecutar una consulta no difusa, en SQL).

▪ Menú "F-SQL"

- View SQL Translation (Ver la traducción a SQL de una sentencia FSQL).
- Execute F-SQL Query (Ejecutar una consulta flexible, en FSQL).
- FQBuilder Wizard (Llamada al submódulo asistente FQBuilder Wizard).

▪ Menú "D-(F)SQL"

- View SQL Translation (Ver la traducción a SQL de una sentencia escrita en DFSQL).

  - Execute D(F)SQL Sentence (Ejecutar una sentencia deductiva, en D(F)SQL).

- Menú "Insert" (Inserción)
  - DDL Sentence (Inserción de la cabecera de una sentencia del DDL, Lenguaje de Definición de Datos).
  - DML Sentence (Inserción de la cabecera de una sentencia del DML, Lenguaje de Manejo de Datos).
  - Logic operator (Inserción de un operador lógico).
  - Set operator (Inserción de un operador de conjuntos).
  - Classic crisp comparator (Inserción de un comparador clásico, "crisp").
  - F-SQL fuzzy comparator (Inserción de un comparador difuso, específico de FSQL).
  - Fuzzy constant (Inserción de una constante difusa, específica de FSQL).
- Menú "Data Miner"
  - Run (Lanzar el módulo Data Miner).
- Menú "Help" (Ayuda)
  - Help contents (Llamada a la ayuda en formato HTML).
  - About FuzzyQueries 2+ (Créditos de la aplicación).

A.2.1.2. Gestión de archivos

En la gestión de consultas, FuzzyQueries 2+ nos permite comenzar desde cero con una consulta nueva  , abrir una consulta existente  , o bien almacenar la consulta actual en modo texto  . Siempre que nos encontremos editando una consulta y vayamos a crear una nueva o a abrir una ya existente, el programa nos indicará si deseamos almacenar las últimas modificaciones realizadas, con objeto de no perderlas por equivocación.

También, desde el menú "Archivo", podemos abrir un archivo de resultados que ya hubiera sido generado en una sesión anterior. Esto nos puede servir para

agilizar cálculos sobre los datos, si sabemos que éstos no han variado en la Base de Datos desde la última consulta que se realizó sobre ellos.

A.2.1.3. Edición de consultas

La edición de la consulta se realiza como en cualquier otro procesador de texto simple. Teniendo siempre en mente la sintaxis usada por el lenguaje SQL, podemos escribir nuestra consulta, flexible (usando comandos y términos exclusivos de F-SQL) o clásica. Como en muchos otros editores, contamos con las herramientas de Cortar  , Copiar  y Pegar  para facilitar nuestra labor en la escritura. Para detalles más cercanos al lenguaje F-SQL y a la estructura de los datos almacenados en la base de datos sobre los que estamos consultando, FuzzyQueries 2+ incorpora algunas novedosas funcionalidades.

Palabras reservadas A través del menú "Inserción."[en] la barra de menús, accedemos a una colección de operadores y palabras reservadas de SQL (y de F-SQL) para su uso en las consultas. Esta herramienta nos puede resultar útil como recordatorio de las operaciones que se pueden realizar en una consulta.

Las palabras reservadas se encuentran agrupadas de acuerdo a su significado, para una navegación más intuitiva. He aquí una lista de dichas palabras:

- DDL Sentence (Sentencias del DDL, Lenguaje de Definición de Datos).
  - `CREATE TABLE`
  - `CREATE LABEL`
  - `CREATE NEARNESS`
  - `CREATE INTENSIONAL TABLE`
  - `CREATE RULE`
- DML Sentence (Sentencias del DML, Lenguaje de Manejo de Datos).
  - `INSERT INTO`
  - `DELETE FROM`
  - `SELECT`

- Logic operator (Operadores lógicos).
  - `NOT`
  - `AND`
  - `OR`
- Set operator (Operadores de conjuntos).
  - `UNION`
  - `UNION ALL`
  - `INTERSECT`
  - `MINUS`
- Classic crisp comparator (Comparadores clásicos, "crisp").
  - Igual, `=`
  - Distinto de, `<>`
  - Menor que, `<`
  - Menor o igual que, `<=`
  - Mayor que, `>`
  - Mayor o igual que, `>=`
  - `BETWEEN . AND .`
  - `NOT BETWEEN . AND .`
  - `IS NULL`
  - `IS NOT NULL`
  - `IN`
  - `NOT IN`
  - `LIKE`
  - `NOT LIKE`
  - `EXISTS`
  - `NOT EXISTS`
- F-SQL fuzzy comparator (Comparadores difusos, específicos de FSQL).
  - "Igual" difuso, `FEQ`
  - "Mayor que" difuso, `FGT`
  - "Mayor o igual que" difuso, `FGEQ`

- "Menor que" difuso, `FLT`
- "Menor o igual que" difuso, `FLEQ`
- "Mucho mayor que" difuso, `MGT`
- "Mucho menor que" difuso, `MLT`
- "Necesariamente igual" difuso, `NFEQ`
- "Necesariamente mayor que" difuso, `NFGT`
- "Necesariamente mayor o igual que" difuso, `NFGEQ`
- "Necesariamente menor que" difuso, `NFLT`
- "Necesariamente menor o igual que" difuso, `NFLEQ`
- "Necesariamente mucho mayor que" difuso, `NMGT`
- "Necesariamente mucho menor que" difuso, `NMLT`
- `IS (NOT) NULL` (en el sentido difuso)
- `IS (NOT) UNDEFINED`
- `IS (NOT) UNKNOWN`

▪ Fuzzy constant (Constantes difusas, específicas de FSQL).

- `UNKNOWN`
- `UNDEFINED`
- `NULL`
- Distribución trapezoidal, `$[a,b,c,d]`
- Intervalo, `[m,n]`
- Valor aproximado, `#n`
- Etiqueta, `$nombre` (debe estar previamente definida en la FMB).

Ventana FQWizard Mediante la herramienta FQWizard, accesible a través del menú "F-SQL"→"FQBuilder Wizard" o con el botón  en la barra de herramientas, podemos obtener información directa sobre el contenido del Catálogo del Sistema, y más concretamente de la Base de Metaconocimiento Difuso, FMB, que ya fue descrita en el apartado 3.2.2. Se puede encontrar una descripción más detallada del proceso en el apartado A.2.2, que veremos más adelante.

A.2.1.4. Conexión con el Servidor de Bases de datos

Antes de obtener resultados del Sistema Gestor de Bases de Datos, habremos de estar conectados a él. Si pulsamos sobre el botón  , o en su defecto elegimos la opción "Connect to DB" dentro del menú "DB", accederemos a una ventana modal desde la que introducir los campos necesarios para la conexión. Éstos son: login, password, servidor, puerto y servicio al que se desea acceder.

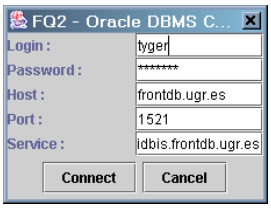


Figura A.2: Conexión con la base de datos

Para facilitar en la medida de lo posible un uso continuado del programa, los campos necesarios para conectarnos a la base de datos aparecen rellenos con unos valores iniciales por defecto. En cualquier caso, éstos siempre pueden ser fácilmente modificados.

Si la conexión tiene éxito, a partir de ese instante podremos ejecutar consultas y acceder a otros tipos de información sobre la Base de Datos. Para esto último siempre podemos usar la opción "Info", en el menú "Help" (botón ).

Si en algún momento deseamos desconectarnos de la Base de Datos, bien para volver a conectarnos como un usuario distinto o simplemente por terminar la sesión con FuzzyQueries 2+, podemos hacerlo de dos formas. A través del menú "DB" o mediante el botón  .

Mientras se concluye la sesión con la aplicación, y antes de que ésta nos devuelva al sistema, se comprueba si aún seguimos conectados a la base de datos. De ser así, se realiza la desconexión de forma automática. Aún así, se recomienda al usuario que sea él mismo quien realice este paso antes de terminar su sesión con FuzzyQueries 2+.

A.2.1.5. Traducción de una sentencia FSQL a SQL

Antes de ejecutar una consulta flexible F-SQL puede interesarnos comprobar si la sintaxis es correcta, o simplemente ver cuál es la sentencia SQL final

en la que se traduce la anterior. Recordemos, por el apartado A.1, que toda sentencia escrita en F-SQL tiene su equivalente en SQL estándar, aunque es el servidor difuso de bases de datos quien se encarga de realizar esta traducción, de forma transparente al usuario. Pues bien, también desde el menú "F-SQL", con la opción "View SQL translation", o mediante el botón  en la barra de herramientas, accederemos a una ventana desde la cual podremos visualizar la sentencia SQL final que se enviará al Sistema Gestor de Bases de Datos.

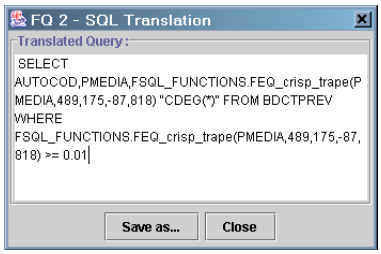


Figura A.3: Ejemplo de traducción de una sentencia FSQL a SQL

Desde esta ventana podemos modificar la consulta, aunque los cambios sólo tendrán valor en el caso de que deseemos almacenar la consulta en un archivo de texto. La consulta original no se verá afectada desde aquí.

A.2.1.6. Ejecución de una sentencia

Una vez conectados a la Base de Datos y con nuestra sentencia (F)SQL lista para ser enviada al Servidor de Bases de Datos, debemos pulsar el botón  (en el menú "F-SQL", "Execute F-SQL Query"). Tras un breve intervalo de tiempo, durante el cual la sentencia es procesada, enviada al Servidor y ejecutada, los resultados obtenidos se recuperarán y formatearán antes de ser presentados en pantalla en la ventana de Resultados (ver apartado A.2.3). Si durante la ejecución de la sentencia se produjo algún error, nos aparecerá una ventana modal para indicarnos la naturaleza del mismo y darnos la opción de corregirlo si es posible, siempre que éste sea debido al usuario y no dependa del servidor de bases de datos (por ejemplo, que se haya encontrado un error sintáctico en la consulta).

FuzzyQueries 2+ permite ejecutar tanto consultas flexibles (F-SQL) como clásicas (sólo SQL). Mediante el proceso descrito, podemos ejecutar ambos

tipos. Sin embargo, y para agilizar el proceso, si la consulta en curso sólo opera dentro de los marcos del lenguaje SQL, FuzzyQueries 2+ nos proporciona un mecanismo para ejecutar consultas SQL. Este proceso es significativamente más rápido, ya que evitamos el coste en tiempo que supone realizar una llamada innecesaria al traductor de sentencias F-SQL. Para ello, desde el menú "DB", debemos escoger la opción "Execute SQL Query", o bien acceder a dicha opción a través del botón  .

A.2.2. Ventana FQWizard

Mediante la herramienta FQWizard, a la que puede accederse a través del menú "F-SQL"→"FQBuilder Wizard" o con el botón  en la barra de herramientas, podemos obtener información directa sobre el contenido del Catálogo del Sistema.

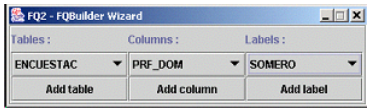


Figura A.4: Ventana FQWizard

Pensado inicialmente como un asistente para la construcción de consultas flexibles, tal y como ya comentamos en el apartado 6.2.1, FQWizard acabó formando parte de FuzzyQueries 2+ como un módulo adicional. A través de FQWizard podemos:

- Obtener el total de tablas y vistas de usuario accesibles desde la sesión actual.
- Obtener el conjunto de columnas asociadas a una tabla concreta.
- Si la columna seleccionada por el usuario admite tratamiento difuso y, además, se ha definido previamente sobre ella un conjunto de etiquetas, podremos consultar éstas. Si además, están definidas como un conjunto de distribuciones de posibilidad sobre un dominio subyacente numérico, FQWizard nos mostrará una representación gráfica de las mismas.

A.2.2.1. Tablas disponibles

Previa conexión con el Servidor de Bases de Datos, al iniciar FQWizard se recogerá desde el servidor el total de tablas o vistas accesibles por el usuario. Eligiendo una de ellas, el programa obtendrá el conjunto de columnas asociado. Al mismo tiempo, pulsando sobre el botón "Add table", el nombre de la tabla será añadido en el texto de la consulta, en la posición actual del cursor.

A.2.2.2. Columnas

Elegida la tabla en el paso anterior, FQWizard consultará el conjunto total de columnas que contiene. También podremos añadir el nombre de esa columna a la consulta en curso mediante el botón "Add column" bajo la lista de columnas.

A.2.2.3. Etiquetas asociadas

Si la columna que el usuario ha seleccionado admite tratamiento difuso, esto es, es de alguno de los tipos difusos de F-SQL (FTYPE1, FTYPE2, FTYPE3), descritos en el apartado A.1, FQWizard consultará la Base de metaconocimiento difuso en busca de algún conjunto de etiquetas lingüísticas definido sobre esa columna. En caso afirmativo, las mostrará por pantalla en una lista y el usuario podrá añadirlas (botón "Add label") a su consulta si así lo cree necesario. Paralelamente, y como se describe a continuación, si la columna tiene un dominio numérico subyacente (lo cual ocurre en los tipos FTYPE1 y FTYPE2), se nos mostrará mediante una gráfica en una ventana aparte las distribuciones de posibilidad asociadas a las etiquetas.

A.2.2.4. Representación gráfica

Si el conjunto de etiquetas que FQWizard ha encontrado se encuentra definido sobre un dominio subyacente numérico, FQWizard mostrará una representación gráfica de ese dominio, con el conjunto de las etiquetas lingüísticas dispuesto en él.

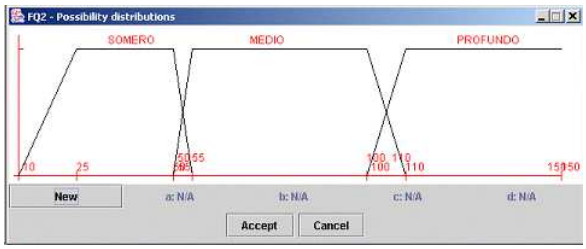


Figura A.5: Conjunto de etiquetas lingüísticas sobre un dominio

A.2.2.5. Definición de una nueva distribución trapezoidal de probabilidad

Apoyándonos en esa representación gráfica, FQWizard nos permite definir una distribución de posibilidad "in situ", que podrá ser añadida a la consulta actual como un literal, esto es, un valor constante. Para ello, debemos pulsar el botón "New", tras lo cual FQWizard creará por nosotros una distribución general. Mediante el ratón, podremos ajustar los puntos de la figura a nuestra conveniencia. La ventana modal nos mostrará en la zona inferior los valores numéricos actuales. Pulsando sobre "Accept", añadiremos la distribución como una constante F-SQL a nuestra consulta. Para volver atrás sin realizar ningún cambio, bastará con pulsar "Cancel".

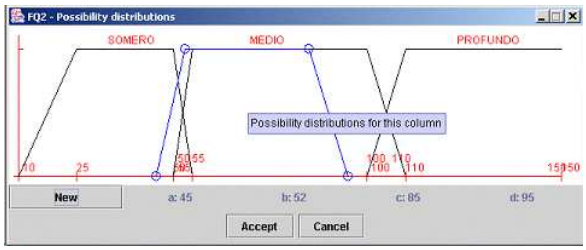


Figura A.6: Creación de una nueva etiqueta

Si el usuario así lo desea, FuzzyQueries 2+ permite que éste deje constancia de la nueva distribución trapezoidal definida, asignándole una etiqueta lingüística, que deberá ser introducida por teclado (y que no deberá existir previamente) y almacenándola en el catálogo del sistema, desde donde podrá reutilizarla en una sesión posterior. Para ello, la aplicación presentará una

ventana modal en la que se dará opción de asignar un nombre de etiqueta a la distribución. Si el usuario pulsa "Yes", la etiqueta se almacenará en la base de datos. Si pulsa "No", tan solo se añadirá a la consulta en curso la distribución como una constante FSQL (ver tabla A.2). Por último, si el usuario pulsa "Cancel", se le devolverá a la ventana de visualización y definición de etiquetas.

Figura A.7: Definición de un nombre para una nueva etiqueta

Nótese que, para que dicha operación tenga éxito, el usuario ha de contar con los privilegios necesarios para poder realizarla sin problemas.

A.2.2.6. Fuzzy Deductor

Fuzzy Deductor es una herramienta para el manejo de predicados difusos intensivos, los cuales podemos representar en términos de consultas sobre tablas difusas mediante F-SQL. FuzzyQueries 2+ permite dos operaciones básicas al respecto. Por un lado, podemos construir uno de estos predicados (para lo cual también podemos hacer uso del conjunto de palabras reservadas al que se accede a través del menú "Insert"), y ejecutarlo, bien pulsando sobre el botón  , o bien seleccionando la opción "Execute D(F)SQL sentence", en el menú "D-(F)SQL". Análogamente a como ocurría con las consultas flexibles en FSQL, también aquí podemos visualizar previamente la traducción a SQL estándar de la consulta deductiva, seleccionando la opción "View SQL Translation" del menú anterior, o a través del botón  .

Más información sobre los fundamentos teóricos en los que se basa este módulo puede hallarse en [Blanco, 2001].

A.2.3. Ventana Resultados

La ventana de resultados de FuzzyQueries 2+ nos muestra una tabla con el resultado de una consulta, flexible o no. Desde aquí se da opción al usuario

para que realice diversas operaciones, que van desde efectuar algunos cálculos sobre los resultados, como los que se verán más adelante, hasta simplemente almacenar en disco la tabla de valores.

A.2.3.1. Presentación de los resultados

Tras la ejecución de una consulta, sea ésta flexible o no, FuzzyQueries 2+ devuelve una tabla con los resultados, dentro de una ventana como la que nos muestra la figura A.8. Para una navegación más cómoda a través de la tabla de resultados, además de la aparición de barras de scroll vertical u horizontal cuando sea preciso, podemos ordenar los resultados de una columna alfabéticamente (o en sentido inverso). Para ello, basta con pulsar con el botón izquierdo del ratón sobre el nombre de la columna que deseamos usar como índice. Los valores que aparezcan en dicha columna se ordenarán ascendentemente, permutando las filas de resultados en consecuencia. Si, por el contrario, el usuario desea ordenar los valores en sentido descendente, bastará con pulsar simultáneamente la tecla "shift" (mayúsculas) y el botón izquierdo del ratón sobre el nombre de la columna correspondiente.

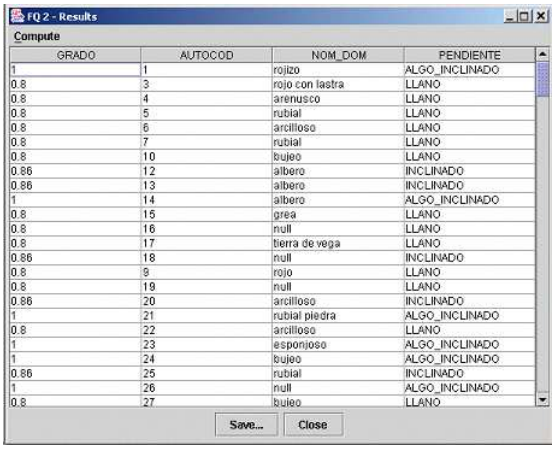


Figura A.8: Ejemplo de presentación de los resultados de una consulta

Desde esta misma ventana de resultados, se da pie a que el usuario almacene dichos resultados en disco de distinta forma :

- Texto simple. La tabla se almacenará en un archivo común de texto, con

la extensión ".txt".

- Tabla HTML. FuzzyQueries 2+ permite almacenar la tabla en un archivo de extensión ".html", para poder ser visualizada directamente en un navegador web.

- Objeto java. Por último, como una funcionalidad añadida, podemos almacenar los resultados en un archivo binario, como un objeto, de forma que luego pueda ser recuperado desde el mismo FuzzyQueries 2+. Esto nos puede servir, por ejemplo, para aplicar alguna de las operaciones sobre los resultados que se comentan a continuación sin tener que ejecutar de nuevo la consulta que los generó.

  Para acceder de nuevo a estos valores, hemos de seleccionar la opción "Open Result" en el menú "File".

A.2.3.2. Operaciones sobre los resultados

FuzzyQueries 2+ admite que sobre los resultados generados a través de una consulta se puedan efectuar algunos cálculos experimentales. Este apartado se ha diseñado de la forma más general posible, para que se permita ampliarlo en un futuro cuando sea necesario. Por el momento, las medidas implementadas están accesibles desde el menú "Compute".

A.2.3.2.1. Cálculo del Cardinal Difuso (Fuzzy Cardinal) El cardinal de un conjunto se define como el número de elementos contenidos en ese conjunto. En el caso de un conjunto difuso, es razonable pensar que el cardinal será también difuso. Diversos enfoques han sido estudiados en la bibliografía e implementados en FuzzyQueries 2+. Para más detalles, remitimos al lector al apartado 3.1.5, donde ya dedicamos un espacio a definir los cardinales difusos más conocidos.

A.2.3.2.2. Cálculo de la Medida de Importancia Not Null (Fuzzy Not Null) También relacionada con el cálculo del cardinal difuso, la medida de importancia Not Null nos indica el grado de interés o importancia de un valor difuso. Esta medida nos puede resultar interesante de cara a tener o no en

cuenta dicho valor difuso en el cálculo del cardinal, así como en cualquier otra operación que lo precise. Para más información sobre la medida Not Null y cómo se calcula por FuzzyQueries 2+, remitimos al lector a [Marín et al, 2003].

A.2.4. Ventana FuzzyCardinal

Podemos operar sobre el resultado de la ejecución de una consulta difusa como si de un conjunto difuso se tratara. Para ello, nos bastará con haber incluído entre las columnas la función CDEG(*), que nos devuelve el grado de cumplimiento de la condición para toda la tupla (como ya fue descrito en el apartado A.1. Dicho valor puede ser interpretado como el grado de pertenencia de la tupla completa al conjunto de resultados, y usado en consecuencia.

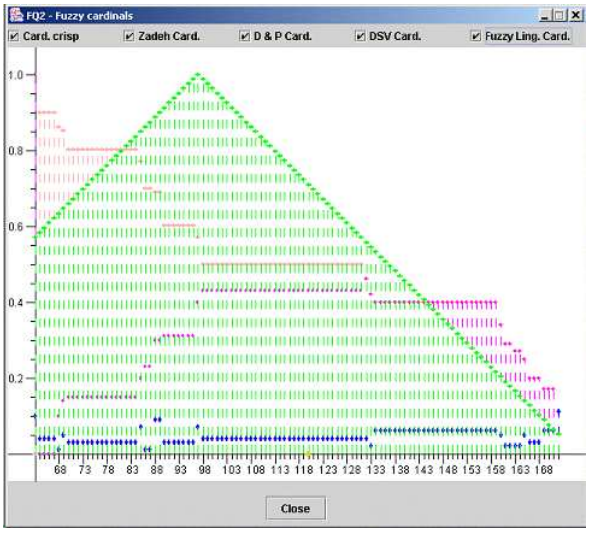


Figura A.9: Comparación de varios cardinales difusos

Si FuzzyQueries 2+ detecta que dicho valor se encuentra entre el conjunto de columnas, nos permitirá obtener el cardinal difuso del conjunto resultado y la opción correspondiente dentro del menú "Compute" se verá resaltada. El cardinal de un conjunto difuso ha sido objeto de muchos estudios y de ahí que existan varias definiciones sobre el mismo, las cuales ya fueron recordadas en el apartado 3.1.5. FuzzyQueries 2+ calcula algunas de ellas, concretamente el cardinal crisp de un conjunto difuso según DeLuca y Ter-

mini [DeLuca y Termini, 1972], el cardinal difuso de Zadeh [Zadeh, 1983], el cardinal difuso de Dubois y Prade [Dubois y Prade, 1985], el cardinal difuso probabilístico de Delgado, Sánchez y Vila [Delgado et al., 2000b], y el cardinal lingüístico definido por Vila et al. [Marín et al, 2003], mostrándolas en pantalla. FuzzyQueries 2+ calcula todos esos cardinales, y el usuario puede decidir cuáles mostrar y cuáles no, en una representación gráfica del dominio numérico subyacente.

A.2.5. Ventana NotNull

A partir de la ventana de resultados de FuzzyQueries 2+ podemos acceder al cálculo de diversas medidas difusas sobre dichos resultados. Una de ellas es la comentada a continuación, la medida de importancia Not Null. Para su obtención, el programa nos dará a escoger la columna difusa sobre la cual se efectuará el cálculo tras lo cual se nos mostrará por pantalla el resultado del mismo, como una tabla en la que se indica el valor de la medida para cada ocurrencia en la columna.

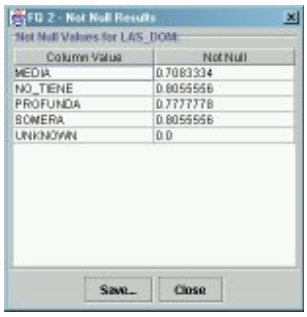


Figura A.10: Ventana NotNull

También se nos permite almacenar estos resultados en distintos formatos, ya sea en modo texto o como una tabla HTML. Algunos fundamentos teóricos sobre esta medida se discuten en [Marín et al, 2003].

A.3. (Fuzzy) Data Miner

Implementada originalmente como una aplicación independiente, la herramienta Data Miner para extracción de Reglas de Asociación y Dependencias Aproximadas en bases de datos relacionales ha sido finalmente integrada en la arquitectura de FuzzyQueries 2+, tal y como ya anticipábamos en el capítulo 6. Una vez conectados a la base de datos, a través de los menús o la barra de herramientas de FuzzyQueries 2+, (botón ), accederemos a una nueva ventana desde configurar nuestro módulo de minería de datos.

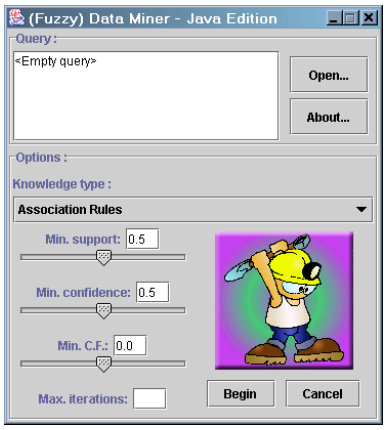


Figura A.11: Ventana de configuración de Data Miner

Desde aquí, se nos mostrará la consulta a partir de la cual se lanzará el proceso de extracción de conocimiento. Dicha consulta habremos de haberla construido o editado previamente desde FuzzyQueries 2+. En la ventana de configuración, procederemos a establecer los umbrales para los parámetros de soporte, confianza y factores de certeza, así como elegir entre los distintos tipos de conocimiento a nuestra disposición. También es posible establecer un número máximo de iteraciones en el algoritmo. Este número equivale también al número total de ítems involucrados en las reglas resultantes. Por defecto, se considera que el algoritmo se ejecutara mientras sea capaz de extraer nuevos ítemsets.

A.3.1. Ejecución independiente de Fuzzy Data Miner

Fuzzy Data Miner también puede ser ejecutado en solitario, sin necesidad de lanzar previamente la aplicación FuzzyQueries 2+. En este caso, obviamente, contaremos sólo con las herramientas de minería de datos. Antes de comenzar dicho proceso, se dará opción al usuario a que elija su fuente de datos. Por un lado, podemos usar un conjunto de archivos de texto, que habrán de seguir un formato determinado, separando en cada línea los valores por medio de tabulaciones. Este proceso está más indicado para conjuntos de datos sencillos y pequeños, y al no necesitar de una conexión con un servidor remoto de bases de datos, resulta más rápido.

Por otro lado, podemos seguir obteniendo los datos de origen a partir de una tabla o conjunto de tablas dentro de una base de datos. En tal caso, habremos de conectarnos previamente a ella, en un proceso análogo al descrito en el apartado A.2.1.4. Después, el usuario deberá cargar una consulta (a partir de un archivo, normalmente con la extensión ".sql"), para usar el resultado de la misma como conjunto de datos.

El resto del proceso de minería de datos no variará, tanto si ejecutamos Fuzzy Data Miner como submódulo de FuzzyQueries 2+ como si lo ejecutamos independientemente, y tanto si la fuente de origen de los datos es un archivo de texto o el resultado de una consulta sobre una base de datos. Para una descripción más completa de dicho proceso, remitimos al apartado 2.3 de esta misma memoria, donde recordamos los conceptos y procesos fundamentales asociados a la minería de datos y a la extracción de conocimiento en bases de datos.

A.3.2. Cálculo de ítemsets frecuentes

En el primer paso del algoritmo de minería de datos, tal y como ya conocemos por los capítulos preliminares de esta memoria, se obtiene un conjunto con los ítemsets de mayor frecuencia de aparición en la base de datos considerada. Estos ítemsets están agrupados de acuerdo al número de elementos (ítems) que contienen. El cálculo del conjunto de ítemsets frecuentes suele ser la fase más lenta y costosa en el proceso global de extracción de conocimiento, debido a que se han de realizar llamadas remotas a un servidor de bases de

datos, y a que normalmente se deberán de realizar varias pasadas sobre el conjunto de datos. Comúnmente, la duración está en función del tamaño del conjunto de datos y de la longitud de los ítemsets que estamos procesando. Tras la finalización del cálculo, se muestra al usuario el conjunto de ítemsets frecuentes y el número de éstos, en una ventana aparte, como la que aparece en la figura A.12.

A partir de este conjunto, se le dará la oportunidad al usuario de continuar con la segunda y última fase de la extracción de conocimiento, así como también de guardar los ítemsets en un archivo de texto o en formato HTML, con el objeto de facilitar la generación de posibles informes.

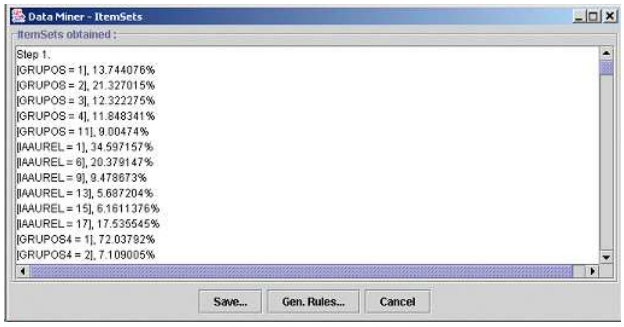


Figura A.12: Presentación de los ítemsets obtenidos

A.3.3. Extracción de reglas o dependencias

Por último, el algoritmo de minería de datos toma los conjuntos de ítemsets frecuentes y, mediante un proceso recursivo que ya fue descrito en el apartado 2.3.4.5, va obteniendo las reglas (del tipo que se haya especificado en la etapa de configuración, esto es, reglas de asociación o dependencias funcionales, difusas o no), realizando una criba de los resultados en función de los umbrales de soporte, confianza y factor de certeza considerados. El conjunto de reglas resultante se ofrece por pantalla (en una ventana como la que muestra la figura A.13) y también en esta ocasión se le da al usuario la posibilidad de almacenar los resultados en disco, bien en modo texto o como una tabla HTML.

Al igual que ocurría con las tablas de resultados de una consulta difusa, podemos ordenar ascedente o descendentemente las reglas o dependencias

obtenidas pulsando sobre los nombres de las columnas, que en este caso corresponden a *antecedente*, *consecuente*, *soporte*, *confianza* y *factor de certeza*.

Adicionalmente, y con objeto de facilitar el manejo del conjunto de reglas (o dependencias) resultante, que en ocasiones puede llegar a ser excesivamente grande, podemos almacenar dicho conjunto en un archivo de texto formateado, esto es, separando los componentes de la regla (antecedentes, consecuentes, soporte, confianza y factor de certeza) mediante tabulaciones. Esto nos permite importar este archivo desde otras aplicaciones o facilitar el almacenamiento del conjunto de resultados en una base de datos, con objeto de facilitar el tratamiento y la visualización de los mismos.

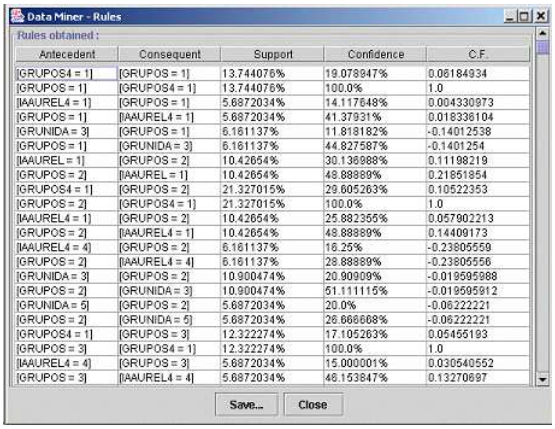


Figura A.13: Presentación de las reglas resultantes

FuzzyQueries 2+ es un proyecto abierto, dentro del seno del grupo de investigación IdBIS, y se encuentra en continuo periodo de actualización y adaptación a los nuevos resultados obtenidos en nuestro trabajo. Estaremos abiertos a cualquier tipo de comentario y sugerencia.

Para más información, recomendamos al lector que visite la página web del grupo de investigación IdBIS, `http://frontdb.ugr.es`.

B. Algoritmos utilizados en la memoria

En pro de facilitar una mejor lectura global de la presente memoria, dedicamos este apéndice a la recopilación de los distintos algoritmos a los que se ha hecho referencia a lo largo de la misma. Entrarían en esta categoría tanto aquellos algoritmos ya existentes, que simplemente hemos citado en momentos puntuales donde era necesario recordarlos, así como los algoritmos que fue necesario definir para llevar a la práctica los resultados teóricos obtenidos y presentados en la memoria de esta tesis doctoral.

Algoritmo B.1 Algoritmo para transformar una relación r en una relación transaccional T

Entrada: r, una relación, $RE_r = \{At_1, At_2, \ldots, At_k\}$ el subconjunto de atributos en r de un esquema relacional RE.

Salida: T, una relación transaccional.

1: $RE_T \leftarrow \emptyset$
2: **para cada** $At \in RE_r$ **hacer**
3: **para cada** $a \in dom(At)$ **hacer**
4: $RE_T \leftarrow RE_T \cup \{< At, a >\}$
5: **fin para**
6: **fin para**
7: **para cada** transacción $t' \in T$ **hacer**
8: Inicializar todas sus columnas a 0.
9: **fin para**
10: **para cada** tupla $t \in r$ **hacer**
11: **si** $t[At] = a$ **entonces**
12: $t'[At = a] = 1$
13: **fin si**
14: **fin para**
15: Devolver T; Fin.

Algoritmo B.2 [[Agrawal y Srikant, 1995]] Algoritmo Apriori para el cálculo de ítemsets frecuentes a partir un T-set (primera etapa de la Extracción de Reglas de Asociación)

Entrada: I, un conjunto de ítems, T, un conjunto de transacciones sobre I.
Salida: F, conjunto de ítemsets frecuentes.

{Inicialización}
Definir un contador $c_{\{i\}}$ para cada $i \in I$
$L_1 \leftarrow \{\{i\} | i \in I\}$
$F \leftarrow \emptyset$
$l \leftarrow 1$
mientras $l \leq |I|$ y $L_l \neq \emptyset$ **hacer**
  para cada $t \in T$ **hacer**
    para cada $I_* \in L_l$ **hacer**
      si $I_* \subseteq t$ **entonces**
        $c_{I_*} \leftarrow c_{I_*} + 1$
      fin si
    fin para
  fin para
  para cada $I_* \in L_l$ **hacer**
    si $c_{I_*} < minsupp \times |T|$ **entonces**
      $L_l \leftarrow L_l \backslash \{I_*\}$
      Liberar la memoria usada por c_{I_*}
    fin si
  fin para
  {Actualización de variables}
  $F \leftarrow F \cup L_l$
  $L_{l+1} \leftarrow CrearNivel(l+1, L_l)$
  $l \leftarrow l + 1$
fin mientras
Devolver F; Fin.

Algoritmo B.3 Algoritmo para la generación de Reglas de Asociación a partir de un conjunto de ítemsets frecuentes, extensión del aparecido en [Agrawal et al., 1993]

Entrada: F, conjunto de ítemsets frecuentes, obtenido por medio del algoritmo B.2 o alguna extensión del mismo.

Salida: AR, conjunto de reglas de asociación.

{Llamada al algoritmo recursivo}

1: $RA \leftarrow \emptyset$
2: **para cada** k-ítemset $I_k \in F$, $k \geq 2$ **hacer**
3: $GenReglas(I_k, I_k)$
4: **fin para**

{$GenReglas$ genera reglas de la forma $\tilde{a} \Rightarrow (I_k - \tilde{a}), \forall \tilde{a} \subset a_m$}

1: **Procedimiento** $GenReglas(I_k : k - itemset, a_m : m - itemset)$
2: $A \leftarrow \{(m-1) - itemsets\ a_{m-1} | a_{m-1} \subset a_m\}$
3: **para cada** $a_{m-1} \in A$ **hacer**
4: $conf \leftarrow supp(I_k)/supp(a_{m-1})$
{Nuestra propuesta: Uso del factor de certeza}
5: **si** $conf > supp(I_k - a_{m-1})$ **entonces**
6: $cf \leftarrow \frac{conf - supp(I_k - a_{m-1})}{1 - supp(I_k - a_{m-1})}$
7: **si no si** $conf < supp(I_k - a_{m-1})$ **entonces**
8: $cf \leftarrow \frac{conf - supp(I_k - a_{m-1})}{supp(I_k - a_{m-1})}$
9: **si no**
10: $cf \leftarrow 0$
11: **fin si**
12: **si** $cf \geq mincf$ **entonces**
13: $RA \leftarrow RA \cup \{[a_{m-1}] \Rightarrow [I_k - a_{m-1}]\}$
14: **si** $m - 1 > 1$ **entonces** {Llamada recursiva}
15: $GenReglas(I_k, a_{m-1})$
16: **fin si**
17: **fin si**
18: **fin para**
19: Devolver RA; Fin.

Algoritmo B.4 [[Blanco et al., 2000]] Algoritmo para el cálculo del soporte de un ítemset crisp I_V

1: $S(I_V) \leftarrow 0$
2: **para cada** $i \in \{1, .., K\}$ **hacer**
3: $\quad N(V, v_i) \leftarrow 0$
4: **fin para**
5: **para cada** $t \in r$ **hacer**
6: $\quad N(V, t[V]) \leftarrow N(V, t[V]) + 1$
7: $\quad S(I_V) \leftarrow S(I_V) + 2N(V, t[V]) - 1$
8: **fin para**
9: Salida: $S(I_V)/n^2$ es el soporte del ítemset I_V

Algoritmo B.5 [[Blanco et al., 2000]] Cálculo de ítemsets frecuentes en la extracción de dependencias aproximadas a partir de reglas de asociación

Entrada: RE, un conjunto de atributos, r, una relación sobre RE.
Salida: F, conjunto de ítemsets frecuentes.

{Inicialización}
1: $F \leftarrow \emptyset, l \leftarrow 1, L_1 \leftarrow \emptyset$
2: **para cada** $At_k \in RE$ **hacer**
3: Reservar memoria para $S(\{it_{At_k}\})$
4: $S(\{it_{At_k}\}) \leftarrow 0$
5: $L_1 \leftarrow L_1 \cup \{\{it_{At_k}\}\}$
6: **para cada** $a \in dom(At_k)$ **hacer**
7: Reservar memoria para $N(At_k, a)$
8: $N(At_k, a) \leftarrow 0$
9: **fin para**
10: **fin para**
11: **mientras** $l \leq m$ y $L_l \neq \emptyset$ **hacer**
12: **para cada** $t \in r$ **hacer**
13: **para cada** $I_V \in L_l$ **hacer**
14: $N(V, t[V]) \leftarrow N(V, t[V]) + 1$
15: $S(I_V) \leftarrow S(I_V) + (2N(V, t[V]) - 1)/n^2$
16: **fin para**
17: **fin para**
18: **para cada** $I_V \in L_l$ **hacer**
19: Liberar la memoria ocupada por $N(V, v)$ para cada $v \in dom(V)$
20: **si** $S(I_V) < minsupp$ **entonces**
21: $L_l \leftarrow L_l \backslash \{I_V\}$
22: Liberar la memoria usada por $S(I_V)$
23: **fin si**
24: **fin para**
25: $F \leftarrow F \cup L_l$
26: $L_{l+1} \leftarrow CrearNivel(l + 1, L_l)$
27: $l \leftarrow l + 1$
28: **fin mientras**
29: Devolver F; Fin.

Algoritmo B.6 [[Delgado et al., 2000b]] Algoritmo para calcular $GD_Q(D/A)$ a partir de V_A y $V_{A\cup D}$

1: $j \leftarrow k; GD \leftarrow 0; nf(A)^* \leftarrow 0; acum_A \leftarrow 0; acum_{A\cup D} \leftarrow 0$
   {Cálculo $nf(A)^* = nf(A)^* \times k$}
   {Éste es el factor de normalización}
2: **mientras** $nf(A)^* > 0$ y $V_A(nf(A)^*) = 0$ **hacer**
3: $nf(A)^* \leftarrow nf(A)^* - 1$
4: **fin mientras**
5: **si** $nf(A)^* = 0$ **entonces**
6: devolver ("Error"); Fin
7: **fin si**
8: **mientras** $j > 0$ **hacer**
9: $acum_{A\cup D} \leftarrow acum_{A\cup D} + V_{A\cup D}(j)$
10: $acum_A \leftarrow acum_A + V_A(j)$
11: **si** $j \leq nf(A)^*$ **entonces**
12: $GD \leftarrow GD + Q(\frac{acum_{A\cup D}}{acum_A})$
13: **fin si**
14: $j \leftarrow j - 1$
15: **fin mientras**
   {Normalización}
16: $GD \leftarrow \frac{GD}{nf(A)^*}$
17: Devolver(GD); Fin

Algoritmo B.7 Modificación del algoritmo B.6, para calcular $GD_Q(D/A)$ a partir de V_A y $V_{A\cup D}$ en el caso de dependencias aproximadas difusas

1: $j \leftarrow k; GD \leftarrow 0; nf(A)^* \leftarrow 0$
   {Cálculo $nf(A)^* = nf(A)^* \times k$}
   {Éste es el factor de normalización}
2: **mientras** $nf(A)^* > 0$ y $V_A(nf(A)^*) = 0$ **hacer**
3: $nf(A)^* \leftarrow nf(A)^* - 1$
4: **fin mientras**
5: **si** $nf(A)^* = 0$ **entonces**
6: devolver ("Error"); Fin
7: **fin si**
8: **mientras** $j > 0$ **hacer**
9: **si** $j \leq nf(A)^*$ **entonces**
10: $GD \leftarrow GD + Q(\frac{V_{A\cup D}(j)}{V_A(j)})$
11: **fin si**
12: $j \leftarrow j - 1$
13: **fin mientras**
   {Normalización}
14: $GD \leftarrow \frac{GD}{nf(A)^*}$
15: Devolver(GD); Fin

Algoritmo B.8 [[Kandel y Yelowitz, 1974]]Cálculo de la clausura transitiva de una matriz asociada a una relación difusa

1: Etiquetar todos los posibles valores por medio de los enteros $1, \ldots, N$
2: Construir la matriz primitiva de semejanzas ρ, donde la entrada ij denota el grado de semejanza entre los valores i y j
3: **para** $K = 1$ hasta N **hacer**
4: **para** $I = 1$ hasta N **hacer**
5: **si** $\rho(I, K) \neq 0$ **entonces**
6: **para** $J = 1$ hasta N **hacer**
7: $\rho(I, J) = max(\rho(I, J), min(\rho(I, K), \rho(K, J)))$
8: **fin para**
9: **fin si**
10: **fin para**
11: **fin para**

Algoritmo B.9 Algoritmo para el cálculo del conjunto de clases de equivalencia para un ítemset difuso determinado I_X

Entrada: I_X un ítemset difuso de atributos, S_X una relación difusa de similitud, deg_X, un α-corte.

Salida: Υ, un conjunto de clases de equivalencia.

1: **para cada** $x \in dom(X)$ **hacer**
2: $\quad \overline{x} \leftarrow \{x' | x' \in dom(X)\ and\ S_X(x, x') \geq deg_X\}$
3: $\quad \Upsilon \leftarrow \Upsilon \cup \{\overline{x}\}$
4: **fin para**
5: Devolver Υ, el conjunto de clases de equivalencia para el ítemset I_X en el α-corte deg_X.

Algoritmo B.10 Algoritmo para calcular el soporte de un determinado ítemset difuso de atributos, I_X

1: $j \leftarrow k$
2: **mientras** $j > 0$ **hacer**
3: $\quad \Upsilon \leftarrow CalculaClasesEquiv(I_X, S_X, j)$ (ver Algoritmo B.9)
4: $\quad$ **para cada** $\overline{x} \in \Upsilon$ **hacer**
5: $\quad\quad acum_{\overline{x}} \leftarrow 0$
6: $\quad$ **fin para**
7: $\quad$ **para cada** $x \in dom(X)$ **hacer**
8: $\quad\quad acum_{\overline{x}} \leftarrow acum_{\overline{x}} + N_{(X,x)}[j]$
9: $\quad$ **fin para**
10: $\quad V_{(I_X)}[j] \leftarrow V_{(I_X)}[j] + \sum_{\overline{x} \in \Upsilon} acum_{\overline{x}}^2$
11: $\quad j \leftarrow j - 1$
12: **fin mientras**
13: Devolver $V_{(I_X)}$; Fin.

Algoritmo B.11 Algoritmo para el cálculo de ítemsets frecuentes a partir de T'_r, primera etapa de la Extracción de Dependencias Aproximadas Difusas

Entrada: RE, un conjunto de atributos; r una relación difusa sobre RE; S_{RE}, un conjunto de relaciones de similitud para cada atributo de RE.

Salida: F, el conjunto de todos los ítemsets difusos frecuentes.

1: $F \leftarrow \emptyset; l \leftarrow 1; L_1 \leftarrow \emptyset$
2: **para cada** atributo $At \in R$ **hacer**
3: Reservar memoria para $V_{(\{it_{At}\})}$, un array de $k+1$ posiciones inicializadas a 0
4: $L_1 \leftarrow L_1 \cup \{\{it_{At}\}\}$
5: **para cada** $a \in dom(At)$ **hacer**
6: Reservar memoria para $N_{(At,a)}$, un array de $k+1$ posiciones inicializadas a 0
7: **fin para**
8: **fin para**
9: **mientras** $l \leq m$ y $L_l \neq \emptyset$ **hacer**
10: **para cada** tupla $\tilde{t} \in r$ **hacer**
11: **para cada** ítemset $I_X \in L_l$ **hacer**
12: $N_{(X,\tilde{t}(x))}[\rho(\mu_{\tilde{t}}(X), k)] \leftarrow N_{(X,\tilde{t}(x))}[\rho(\mu_{\tilde{t}}(X), k)] + 1$
13: **fin para**
14: **fin para**
15: **para cada** ítemset $I_X \in L_l$ **hacer**
16: Calcular $V_{(I_X)}$ (ver Algoritmo B.10)
17: Liberar memoria de cada $N_{(X,x)}, \forall x \in dom(X)$
18: Calcular $GD_Q(\tilde{\Gamma}_{I_X}, T'_r)$ (ver Algoritmo B.7)
19: **si** $GD_Q(\tilde{\Gamma}_{I_X}, T'_r) < minsupp$ **entonces**
20: $L_l \leftarrow L_l \backslash \{I_X\}$
21: Liberar memoria de $V_{(I_X)}$
22: **fin si**
23: **fin para**
24: $F \leftarrow F \cup L_l; L_{l+1} \leftarrow CrearNivel(l+1, L_l); l \leftarrow l+1$
25: **fin mientras**
26: Devolver F, el conjunto de todos los ítemsets frecuentes

C. Descripción de los datos para experimentación

Para evitar entorpecer la lectura de la presente memoria, traspasamos a un apéndice separado las tablas asociadas a los experimentos que se llevaron a cabo en el capítulo 4. A continuación, se incluyen las tablas descriptivas de los grupos cuyas relaciones se analizaron en el capítulo 5.

C.1. Experimentos sobre STULONG

La base de datos facilitada por el proyecto STULONG consta de varias tablas, pero nos vamos a limitar a la tabla de ingresos, *Entry*. Incluimos las descripciones de los atributos involucrados, separando éstos en grupos de acuerdo a su semántica. Seguidamente, mostramos las tablas con las relaciones de similitud que se definieron sobre algunos de ellos. Por último, ampliamos y discutimos brevemente los resultados obtenidos en las cuestiones que se plantearon sobre los datos.

Tabla C.1: Factores sociales

Atributo	**Significado**	**Valores**
ROKNAR (ANAC)	Año de nacimiento	numérico
ROKVSTUP (AESTUD)	Año en el que fue estudiado	numérico
STAV (STATUS)	Estatus	1: casado/a 2: divorciado/a 3: soltero/a 4: viudo/a 5: no establecido
VZDELANI (EDUCAL)	Estudios realizados	1: primaria 2: formación profesional 3: secundaria 4: universitarios 5: no establecido
ZODPOV (RESLAB)	Responsabilidad laboral	1: directivo 2: independiente en parte 3: otros 4: jubilado por problemas cardíacos 5: jubilado - otros motivos 6: no establecido

Tabla C.2: Actividades físicas

Atributo	**Significado**	**Valores**
TELAKTZA (FISENT)	Actividad física en el trabajo	1: ppalmente sentado 2: ppalmente de pie 3: ppalmente caminando 4: cargas pesadas 5: no establecido
AKTPOZAM (FISTRT)	Actividad física tras el trabajo	1: ppalmente sentado 2: actividad moderada 3: gran actividad 4: no establecido
DOPRAVA (TRANST)	Medio de transporte para ir a trabajar	1: a pie 2: bicicleta 3: transporte público 4: coche 9: no establecido
DOPRATRV (TTRANST)	Tiempo en llegar al trabajo	5: alrededor de 1/2 hora 6: alrededor de 1 hora 7: alrededor de 2 horas 8: más de 2 horas 5: no establecido

Tabla C.3: Tabaco

Atributo	**Significado**	**Valores**
KOURENI (INTENSF)	Intensidad	1: no fumador 2: fumador (1-4)cig./día 3: fumador (5-14)cig./día 4: fumador (15-20)cig./día 5: fumador >21 cig./día 6: fumador puro/pipa 13: no establecido
DOBAKOUR (TFUM)	Tiempo como fumador	7: hasta 5 años 8: 6-10 años 9: 11-20 años 10: 21 años o más
BYVKURAK (TSINFUM)	Tiempo sin fumar como ex-fumador	11: menos de 1 año 12: más de un año vacío: dejó de fumar en un tiempo anterior/posterior a 1 año

Tabla C.4: Alcohol

Atributo	**Significado**	**Valores**
ALKOHOL (ALCOHOL)	consumición habitual	1: no bebe 2: ocasionalmente 3: regularmente 13: no establecido
PIVO7 (CERV7)	cerveza 7º	8: sí vacío: no
PIVO10 (CERV10)	cerveza 10º	9: sí vacío: no
PIVO12 (CERV12)	cerveza 12º	10: sí vacío: no
VINO	vino	11: sí vacío: no
LIHOV (LICOR)	licores	12: sí vacío: no
PIVOMN (CERVDI)	Cantidad de cerveza diaria	1: no bebe 2: hasta 1 litro 3: más de 1 litro 10: no establecido vacío: sin alcohol
VINOMN (VINODI)	Cantidad de vino diaria	4: no bebe 5: hasta medio litro 6: más de medio litro 10: no establecido vacío: sin alcohol
LIHMN (LICDI)	Cantidad de licor diaria	7: no bebe 8: hasta 100cc 9: más de 100cc 10: no establecido vacío: sin alcohol

Tabla C.5: Reconocimiento físico

Atributo	**Significado**	**Valores**
VYSKA	altura (cm)	numérico
VAHA	peso (kg)	numérico
SYST1	presión sanguínea I sistólica (mm Hg)	numérico
DIAST1	presión sanguínea I diastólica (mm Hg)	numérico
SYST2	presión sanguínea II sistólica (mm Hg)	numérico
DIAST2	presión sanguínea II diastólica (mm Hg)	numérico
TRIC	pliegues dérmicos sobre músculo tríceps (mm)	numérico
SUBSC	pliegues dérmicos sobre músculo subescapular (mm)	numérico

Tabla C.6: Reconocimiento bioquímico

Atributo	**Significado**	**Valores**
CHLST	colesterol en mg %	numérico
TRIGL	triglicéridos en mg %	numérico

Tabla C.7: Relaciones de similitud difusa definidas sobre atributos categóricos de la tabla *Entry*

STATUS	2	3	4	5
1	0.4	0.2	0.4	0.0
2		0.2	0.6	0.0
3			0.2	0.0
4				0.0

EDUCAL	2	3	4	5
1	0.5	0.4	0.3	0.0
2		0.4	0.3	0.0
3			0.3	0.0
4				0.0

RESLAB	2	3	4	5	6
1	0.3	0.2	0.1	0.1	0.0
2		0.2	0.1	0.1	0.0
3			0.1	0.1	0.0
4				0.3	0.0
5					0.0

FISENT	2	3	4	5
1	0.3	0.3	0.2	0.0
2		0.4	0.2	0.0
3			0.2	0.0
4				0.0

INTENSF	2	3	4	5	6	13
1	0.1	0.1	0.1	0.1	0.1	0.0
2		0.4	0.4	0.4	0.4	0.0
3			0.4	0.4	0.4	0.0
4				0.4	0.4	0.0
5					0.4	0.0
6						0.0

FISTRT	2	3	4
1	0.3	0.2	0.0
2		0.2	0.0
3			0.0

TTRANST	6	7	8	9
5	0.3	0.3	0.3	0.0
6		0.3	0.3	0.0
7			0.3	0.0
8				0.0

TRANST	2	3	4	9
1	0.4	0.3	0.3	0.0
2		0.3	0.3	0.0
3			0.4	0.0
4				0.0

TFUM	8	9	10
7	0.4	0.4	0.4
8		0.4	0.4
9			0.4

CERVDI	2	3	10	*vacío*
1	0.1	0.1	0.0	0.0
2		0.1	0.0	0.0
3			0.0	0.0
10				0.0

Tabla C.8: Etiquetas lingüísticas (Atributo BMI)

Etiqueta	α	β	γ	δ
Delgado	0	0	24.75	25
Con sobrepeso	25	25.12	50	50

Tabla C.9: Etiquetas lingüísticas (Atributo SYST1)

Etiqueta	α	β	γ	δ
Baja	90	90	118	121
Media	121	124	138	141
Alta	141	144	225	225

Tabla C.10: Etiquetas lingüísticas (Atributo DIAST1)

Etiqueta	α	β	γ	δ
Baja	50	50	78	81
Media	81	84	88	91
Alta	91	94	145	145

Tabla C.11: Etiquetas lingüísticas (Atributo SYST2)

Etiqueta	α	β	γ	δ
Baja	70	70	118	121
Media	121	124	133	136
Alta	136	139	220	220

Tabla C.12: Etiquetas lingüísticas (Atributo DIAST2)

Etiqueta	α	β	γ	δ
Baja	50	50	78	81
Media	81	84	88	91
Alta	91	94	140	140

Tabla C.13: Etiquetas lingüísticas (Atributo TRIC)

Etiqueta	α	β	γ	δ
Bajo	1	1	8	9
Medio	9	10	11	12
Alto	12	13	35	35

Tabla C.14: Etiquetas lingüísticas (Atributo SUBSC)

Etiqueta	α	β	γ	δ
Bajo	4	4	14	16
Medio	16	18	20	22
Alto	22	24	70	70

Tabla C.15: Etiquetas lingüísticas (Atributo CHLST)

Etiqueta	α	β	γ	δ
Bajo	112	112	211	214
Medio	214	217	249	252
Alto	252	255	530	530

Tabla C.16: Etiquetas lingüísticas (Atributo TRIGL)

Etiqueta	α	β	γ	δ
Bajo	28	28	117	120
Medio	120	123	173	176
Alto	176	179	1197	1197

C.2. Cuestiones planteadas y resultados sobre STULONG

Utilizaremos este apartado para ampliar en más detalle las cuestiones que se plantearon sobre el conjunto de datos, así como la discusión de los resultados obtenidos por medio de nuestra metodología.

Restringiéndonos a la tabla de ingresos (*Entry*), hay varias cuestiones que nos resulta interesante formular. Por ejemplo, es de esperar que existan relaciones entre atributos pertenecientes a un mismo grupo semántico.

- **Hábitos de fumador**. Aplicando nuestra metodología para la extracción de dependencias aproximadas difusas, encontramos ciertas relaciones destacables entre atributos de este conjunto. Por ejemplo,

  $[TFUM] \rightarrow [INTENSF]$, CF 0,318,
  $[INTENSF] \rightarrow [TFUM]$, CF 0,523, y
  $[TSINFUM] \rightarrow [TFUM]$, CF 0,07,
  $[TFUM] \rightarrow [TSINFUM]$, CF 0,38

  El primer par recíproco de dependencias presenta un factor de certeza relativamente alto, algo razonable si nos atenemos a que nos informan sobre la relación entre el número de cigarrillos diarios (atributo $INTENSF$) y el tiempo en años como fumador del individuo (atributo $TFUM$). En cuanto a la segunda pareja de dependencias, la primera de ellas nos aporta información sobre la posible independencia entre atributos en ese sentido: resulta contraintuitivo el que se intente establecer una relación entre el tiempo que se lleva sin fumar (atributo $TSINFUM$) y el número de cigarrillos diarios; no así al contrario, ya que nos encontramos un factor de certeza medianamente alto.

- **Consumo de alcohol**. Sobre los atributos que se engloban dentro de este grupo encontramos algunas dependencias triviales, con un factor de certeza igual o muy cercano a 1,

  $[VINODI] \rightarrow [VINO]$, CF 1,0,
  $[VINO] \rightarrow [VINODI]$, CF 0,64, y

$[LICDI] \rightarrow [LICOR]$, CF 0,99,
$[LICOR] \rightarrow [LICDI]$, CF 0,58

Vemos como en los casos en los que se contempla la cantidad de bebida ingerida diariamente frente a la bebida en cuestión, el factor de certeza nos indica un casi total cumplimiento de las dependencias. De hecho, la primera de ellas es una dependencia funcional entre la cantidad diaria de vino (atributo $VINODI$) y la confirmación de que éste se ingiere (atributo $VINO$). Sin embargo, el caso recíproco presenta algunas excepciones más, de forma que disminuye el valor del factor de certeza. Algo similar ocurre en el segundo par de reglas, que miden los mismos factores pero asociados al consumo de licores (atributos $LICOR$ y $LICDI$).

Hemos de hacer hincapié en el hecho de por qué en esta ocasión hemos considerado únicamente dependencias entre pares de atributos, sin tener en cuenta más atributos. La razón radica en que no ha sido posible hallar ninguna otra dependencia con más atributos que no fuera ya una generalización de una dependencia ya considerada.

Por otro lado, podemos tratar de hallar relaciones entre atributos de distintos grupos semánticos. Como información inicial adicional, contamos también con el hecho de que los pacientes con los que trabajamos fueron clasificados en tres grupos básicos:

- Grupo Normal (el atributo $GRUPOP$ toma los valores 1 o 2).
- Grupo de Riesgo (donde $GRUPOP$ puede valer 3 o 4).
- Grupo Patológico (valor 5 para $GRUPOP$).

En los siguientes subapartados, mostraremos algunos de los resultados más interesantes que conseguimos obtener. Al mismo tiempo, trataremos de, desde nuestro punto de vista, dar una interpretación a los mismos. Con el objeto de optimizar el número de dependencias útiles, consideramos que un umbral adecuado de mínimo factor de certeza podría ser el de $minCF = 0{,}4$, por lo que no mostraremos ninguna dependencia con factor de certeza por debajo de este valor.

C.2.1. Relaciones en las que intervienen factores sociales

Para empezar, debemos destacar el hecho de que hallamos muchas dependencias con un alto valor para el factor de certeza y en las que al mismo tiempo nos aparece el atributo $STATUS$ (status) en el consecuente. Estudiando la distribución de valores en el dominio de dicho atributo, nos encontramos con que existe un gran desequilibrio en dicha distribución (de entre 1417 sujetos estudiados, más de 1200 están casados). Debido a esto, concluimos en que las dependencias en las que interviene este atributo nos van a resultar desinformativas y poco van a aportar sobre las relaciones totales. De ahí que optáramos por omitir el atributo $STATUS$ en los sucesivos análisis.

C.2.1.1. Relaciones entre factores sociales y actividades físicas.

Algunas de las dependencias obtenidas que hemos considerado interesantes son las que se muestran a continuación,

$$[EDUCAL] \rightarrow [FISTRT], supp\ 17{,}47\,\%, CF\ 0{,}53$$
$$[RESLAB] \rightarrow [FISTRT], supp\ 19{,}81\,\%, CF\ 0{,}51$$

con un valor para el factor de certeza similar para todos los grupos de pacientes. Interpretamos que estas dependencias revelan una posible relación entre el nivel de educación alcanzado (atributo $EDUCAL$) y la responsabilidad en el puesto de trabajo (atributo $RESLAB$) con la actividad física tras la jornada laboral (atributo $FISTRT$). Además, para los pacientes dentro de los grupos Normal y de Riesgo, hallamos algunas relaciones adicionales entre los ya citados atributos $EDUCAL$ y $RESLAB$ y el tiempo empleado en llegar al lugar de trabajo (atributo $TTRANST$),

$$[EDUCAL] \rightarrow [TTRANST], supp\ 14{,}78\,\%, CF\ 0{,}47$$
$$[RESLAB] \rightarrow [TTRANST], supp\ 18{,}63\,\%, CF\ 0{,}44$$

C.2.1.2. Relaciones entre factores sociales y tabaco.

En este campo, aparecen fuertes relaciones involucrando aspectos como el nivel educativo alcanzado o la responsabilidad laboral con el tiempo que lleva un ex-fumador sin probar el tabaco (atributo $TSINFUM$). Los valores para

el factor de certeza son más altos en pacientes del grupo de Riesgo, indicando tal vez que las personas con un alto estatus social tienden a llevar una vida más saludable. Basamos esta hipótesis en el claro desequilibrio existente entre ex-fumadores y fumadores dentro de la población estudiada, destacando un mayor número de los primeros.

$[EDUCAL] \rightarrow [TSINFUM]$, *supp* 21,36 %, CF 0,74
$[RESLAB] \rightarrow [TSINFUM]$, *supp* 27,81 %, CF 0,74

C.2.1.3. Relaciones entre factores sociales y consumo de alcohol.

Por un lado, nos encontramos con un atributo, $CERV7$ (que nos indica si el paciente consume o no cerveza de 7^o), para el que su soporte es especialmente elevado (muy cercano al 100 %). La intervención de dicho atributo genera un conjunto de dependencias con un alto factor de certeza (superior a 0.9), que, al fin y al cabo, podrían considerarse casi dependencias funcionales. Por desgracia, dichas dependencias pueden resultarnos de poca o ninguna utilidad, si exceptuamos el hecho de que nos indican que el atributo $CERV7$ no debe ser considerado en lo sucesivo. Por lo demás, no hallamos ninguna otra relación lo suficientemente interesante que involucrara a ninguno de los otros atributos.

C.2.1.4. Relaciones entre factores sociales e índice de masa corporal (BMI).

Estudiando las dependencias obtenidas para los tres grupos de pacientes, encontramos resultados similares en todos ellos. En dichos resultados, se revelan algunas relaciones relativamente altas con el atributo BMI en el consecuente, tales como éstas,

$[AESTUD] \rightarrow [BMI]$, *supp* 17,44 %, CF 0,53
$[EDUCAL] \rightarrow [BMI]$, *supp* 17,04 %, CF 0,51
$[RESLAB] \rightarrow [BMI]$, *supp* 19,77 %, CF 0,51

C.2.1.5. Relaciones entre factores sociales y presión sanguínea.

En este apartado hemos de destacar el hecho de que, considerando los tres grupos de pacientes, sólo hallamos dependencias por encima de los umbrales definidos en el grupo Normal de pacientes. De entre dichas dependencias,

aquéllas con mayores valores para el factor de certeza son las que se muestran a continuación,

$[AESTUD] \rightarrow [DIAST1]$, *supp* 15,48 %, CF 0,49
$[AESTUD] \rightarrow [DIAST2]$, *supp* 15,17 %, CF 0,48
$[EDUCAL] \rightarrow [DIAST1]$, *supp* 17,43 %, CF 0,47
$[EDUCAL] \rightarrow [DIAST2]$, *supp* 17,32 %, CF 0,47
$[RESLAB] \rightarrow [DIAST1]$, *supp* 18,87 %, CF 0,45
$[RESLAB] \rightarrow [DIAST2]$, *supp* 19,10 %, CF 0,46

C.2.2. Relaciones en las que intervienen actividades físicas

Otro experimento realizado sobre el que merece la pena detenernos es el de la búsqueda de relaciones entre las actividades físicas registradas en los pacientes y los siguientes grupos de atributos. A continuación, mostramos los resultados obtenidos al respecto.

C.2.2.1. Relaciones entre actividades físicas y tabaco.

En general, se puede decir que obtuvimos varias dependencias con un alto factor de certeza (hasta 0.76) y con el atributo $TSINFUM$ como consecuente, a partir de las cuales podemos plantear algunas interpretaciones como las ya expuestas en el subapartado C.2.1.

$[FISENT] \rightarrow [TSINFUM]$, *supp* 28,14 %, CF 0,69
$[FISTRT] \rightarrow [TSINFUM]$, *supp* 45,26 %, CF 0,62
$[TRANST] \rightarrow [TSINFUM]$, *supp* 30,33 %, CF 0,71
$[TTRANST] \rightarrow [TSINFUM]$, *supp* 42,03 %, CF 0,64
$[FISENT, FISTRT] \rightarrow [TSINFUM]$, *supp* 16,45 %, CF 0,73
$[FISENT, TRANST] \rightarrow [TSINFUM]$, *supp* 10,64 %, CF 0,76
$[FISENT, TTRANST] \rightarrow [TSINFUM]$, *supp* 15,87 %, CF 0,73
$[FISTRT, TRANST] \rightarrow [TSINFUM]$, *supp* 17,41 %, CF 0,75
$[FISTRT, TTRANST] \rightarrow [TSINFUM]$, *supp* 23,88 %, CF 0,72
$[TRANST, TTRANST] \rightarrow [TSINFUM]$, *supp* 17,44 %, CF 0,75

Pero, hasta donde llega nuestro conocimiento, encontramos más interesantes las dependencias que se muestran a continuación, obtenidas sobre los

pacientes del grupo Patológico,

$$[INTENSF] \rightarrow [FISTRT], \textit{supp}\ 14{,}82\,\%, CF\ 0{,}58$$
$$[TFUM] \rightarrow [FISTRT], \textit{supp}\ 21{,}45\,\%, CF\ 0{,}53$$

De acuerdo con los resultados, podemos decir que existe una relación relativamente alta entre el número de cigarrillos diarios (atributo $INTENSF$) y el tiempo que el paciente lleva fumando (atributo $TFUM$) con la actividad física tras la jornada laboral.

C.2.2.2. Relaciones entre actividades físicas y consumo de alcohol.

Analizando las dependencias obtenidas con nuestro método, podemos afirmar que no existen diferencias significativas entre grupos de pacientes en las que se relacionen las actividades físicas y el consumo del alcohol, exceptuando el hecho de que no aparecieron dependencias de interés con el atributo $TTRANST$ en el consecuente para aquellos pacientes dentro del grupo Patológico. De las restantes dependencias obtenidas, destacaremos aquéllas que nos revelan relaciones entre el consumo de alcohol (atributo $ALCOHOL$) y el tipo de actividad física tras el trabajo junto con el tiempo empleado en llegar al puesto de trabajo. Las mostramos a continuación,

$$[ALCOHOL] \rightarrow [FISTRT], \textit{supp}\ 24{,}41\,\%, CF\ 0{,}45$$
$$[CERVDI] \rightarrow [FISTRT], \textit{supp}\ 22{,}70\,\%, CF\ 0{,}47$$
$$[VINODI] \rightarrow [FISTRT], \textit{supp}\ 23{,}34\,\%, CF\ 0{,}46$$
$$[LICDI] \rightarrow [FISTRT], \textit{supp}\ 21{,}53\,\%, CF\ 0{,}47$$
$$[ALCOHOL] \rightarrow [TTRANST], \textit{supp}\ 22{,}82\,\%, CF\ 0{,}41$$
$$[CERVDI] \rightarrow [TTRANST], \textit{supp}\ 21{,}21\,\%, CF\ 0{,}43$$
$$[VINODI] \rightarrow [TTRANST], \textit{supp}\ 21{,}47\,\%, CF\ 0{,}41$$
$$[LICDI] \rightarrow [TTRANST], \textit{supp}\ 20{,}00\,\%, CF\ 0{,}43$$

C.2.2.3. Relación entre actividades físicas y BMI.

Conforme a los resultados obtenidos, no hallamos prácticamente ninguna diferencia entre las dependencias extraídas para cada uno de los tres grupos. Para los pacientes del grupo Patológico, eso sí, el valor del factor de certeza es

algo mayor, llegando hasta 0.56, aunque, en general, la interpretación de los resultados podría ser la misma. Algunos ejemplos de las dependencias obtenidas son los siguientes,

$[FISENT] \rightarrow [BMI]$, *supp* 16,22 %, CF 0,51
$[TRANST] \rightarrow [BMI]$, *supp* 24,93 %, CF 0,44
$[TTRANST] \rightarrow [BMI]$, *supp* 24,82 %, CF 0,47
$[FISENT, FISTRT] \rightarrow [BMI]$, *supp* 10,22 %, CF 0,56
$[FISTRT, TRANST] \rightarrow [BMI]$, *supp* 15,80 %, CF 0,50
$[FISTRT, TTRANST] \rightarrow [BMI]$, *supp* 16,10 %, CF 0,55

C.2.2.4. Relaciones entre actividad física y presión sanguínea.

En este caso, la principal diferencia que encontramos entre los tres grupos estriba en el número de dependencias obtenidas para cada uno de ellos. Este número es mayor para pacientes del grupo Normal, donde aparecen dependencias con los atributos $DIAST1$ o $DIAST2$ como consecuentes, y que no aparecen en pacientes de los dos grupos restantes. En cuanto a dependencias con los atributos $FISTRT$ o $TTRANST$ como consecuentes, las obtenidas presentan valores similares del factor de certeza para todos los grupos.

$[FISTRT, TRANST] \rightarrow [DIAST1]$, *supp* 11,99 %, CF 0,51
$[FISTRT, TRANST] \rightarrow [DIAST2]$, *supp* 12,04 %, CF 0,51
$[FISTRT, TTRANST] \rightarrow [DIAST1]$, *supp* 15,30 %, CF 0,47
$[FISTRT, TTRANST] \rightarrow [DIAST2]$, *supp* 15,48 %, CF 0,48
$[SYST1, DIAST1] \rightarrow [FISTRT]$, *supp* 15,02 %, CF 0,45
$[SYST1, SYST2] \rightarrow [FISTRT]$, *supp* 15,30 %, CF 0,45
$[SYST1, DIAST2] \rightarrow [FISTRT]$, *supp* 15,26 %, CF 0,46
$[DIAST1, SYST2] \rightarrow [FISTRT]$, *supp* 15,64 %, CF 0,45
$[SYST2, DIAST2] \rightarrow [FISTRT]$, *supp* 16,77 %, CF 0,45
$[TRANST, TTRANST] \rightarrow [DIAST1]$, *supp* 13,10 %, CF 0,52
$[TRANST, TTRANST] \rightarrow [DIAST2]$, *supp* 12,88 %, CF 0,51
$[SYST1, DIAST1] \rightarrow [TTRANST]$, *supp* 14,69 %, CF 0,44
$[SYST1, SYST2] \rightarrow [TTRANST]$, *supp* 14,97 %, CF 0,44
$[SYST1, DIAST2] \rightarrow [TTRANST]$, *supp* 14,72 %, CF 0,44

$[DIAST1, SYST2] \rightarrow [TTRANST]$, *supp* 15,29 %, CF 0,44
$[SYST2, DIAST2] \rightarrow [TTRANST]$, *supp* 16,19 %, CF 0,43

C.2.2.5. Relaciones entre actividades físicas y niveles de colesterol.

Para aquellos pacientes pertenecientes al grupo de riesgo, encontramos destacable el hecho de que aparezcan dependencias entre el nivel de colesterol (atributo $CHLST$) y el nivel de triglicéridos (atributo $TRIGL$), formando parte del antecedente, y los atributos $FISTRT$ y $TTRANST$, por lo que se refiere al consecuente.

$[CHLST] \rightarrow [FISTRT]$, *supp* 19,02 %, CF 0,47
$[TRIGL] \rightarrow [FISTRT]$, *supp* 14,62 %, CF 0,50
$[CHLST] \rightarrow [TTRANST]$, *supp* 17,63 %, CF 0,43
$[TRIGL] \rightarrow [TTRANST]$, *supp* 13,58 %, CF 0,46

En lo que se refiere al resto de grupos, no se hallaron dependencias con el atributo $CHLST$ como antecedente para el grupo Normal,

$[TRIGL] \rightarrow [FISTRT]$, *supp* 14,39 %, CF 0,46
$[TRIGL] \rightarrow [TTRANST]$, *supp* 13,84 %, CF 0,44,

como tampoco fue posible hallar dependencias de interés en las que apareciera el atributo $TTRANST$ como consecuente, en pacientes dentro del grupo Patológico.

$[CHLST] \rightarrow [FISTRT]$, *supp* 20,40 %, CF 0,51
$[TRIGL] \rightarrow [FISTRT]$, *supp* 16,05 %, CF 0,53

C.2.3. Relaciones en las que interviene el consumo de alcohol

Como ya comentamos con anterioridad, el atributo $CERV7$, perteneciente al grupo de atributos sobre consumo del alcohol, presentaba un soporte muy alto y, por dicha razón, decidimos no considerar aquellas dependencias en las que se viera involucrado, por considerarlas desinformativas o distractoras. Por este motivo expuesto, el atributo $CERV7$ no fue considerado en el análisis.

C.2.3.1. Relaciones entre consumo de alcohol y tabaco.

En lo que respecta a las diferencias entre grupos, obtuvimos resultados similares para los tres, aunque con un valor significativamente más alto para el factor de certeza en aquellos pacientes del grupo de Riesgo (con una diferencia de 0.5 para los otros dos grupos y de 0.7 para este grupo). Las dependencias resultantes nos revelan la existencia de una posible relación entre el consumo de diversas bebidas alcohólicas y el tiempo que lleva un ex-fumador sin probar el tabaco, como se muestra a continuación,

$[ALCOHOL] \rightarrow [TSINFUM]$, *supp* 33,17 %, CF 0,69
$[CERVDI] \rightarrow [TSINFUM]$, *supp* 29,87 %, CF 0,69
$[VINODI] \rightarrow [TSINFUM]$, *supp* 31,49 %, CF 0,70
$[LICDI] \rightarrow [TSINFUM]$, *supp* 29,20 %, CF 0,71

C.2.3.2. Relación entre consumo de alcohol y BMI.

Nuestro análisis revela que existe una fuerte relación entre consumo de alcohol (cualquiera de las bebidas consideradas) y el índice de masa corporal, BMI. El número de dependencias que se obtuvo fue bastante alto, con un valor de certeza en torno al 0.5 (por lo cual, si bien no es un valor muy alto, sí que merecería un estudio a un nivel más local), y especialmente en los pacientes pertenecientes al grupo Patológico, como podemos ver por nuestros resultados,

$[ALCOHOL] \rightarrow [BMI]$, *supp* 22,74 %, CF 0,50
$[CERV10] \rightarrow [BMI]$, *supp* 30,32 %, CF 0,41
$[VINO] \rightarrow [BMI]$, *supp* 29,69 %, CF 0,42
$[LICOR] \rightarrow [BMI]$, *supp* 29,85 %, CF 0,42
$[CERVDI] \rightarrow [BMI]$, *supp* 18,93 %, CF 0,52
$[VINODI] \rightarrow [BMI]$, *supp* 22,10 %, CF 0,50
$[LICDI] \rightarrow [BMI]$, *supp* 22,08 %, CF 0,49
$[ALCOHOL, CERV10] \rightarrow [BMI]$, *supp* 12,28 %, CF 0,55
$[ALCOHOL, CERV12] \rightarrow [BMI]$, *supp* 15,59 %, CF 0,55
$[ALCOHOL, VINO] \rightarrow [BMI]$, *supp* 12,57 %, CF 0,56
$[ALCOHOL, LICOR] \rightarrow [BMI]$, *supp* 12,37 %, CF 0,55
$[ALCOHOL, CERVDI] \rightarrow [BMI]$, *supp* 10,50 %, CF 0,59

$[ALCOHOL, VINODI] \rightarrow [BMI]$, *supp* 11,49 %, CF 0,58
$[ALCOHOL, LICDI] \rightarrow [BMI]$, *supp* 11,34 %, CF 0,56

C.2.3.3. Relación entre consumo de alcohol y presión sanguínea.

En lo que respecta a este apartado, solamente pudimos hallar dependencias interesantes para aquellos pacientes pertenecientes al grupo Normal. Dichas dependencias nos revelan unas relaciones relativamente alta entre el consumo de bebidas alcohólicas y la presión diastólica, como se muestra a continuación,

$[ALCOHOL] \rightarrow [DIAST1]$, *supp* 21,19 %, CF 0,43
$[CERVDI] \rightarrow [DIAST1]$, *supp* 22,51 %, CF 0,45
$[VINODI] \rightarrow [DIAST1]$, *supp* 22,74 %, CF 0,43
$[LICDI] \rightarrow [DIAST1]$, *supp* 21,26 %, CF 0,44
$[ALCOHOL] \rightarrow [DIAST2]$, *supp* 21,00 %, CF 0,43
$[CERVDI] \rightarrow [DIAST2]$, *supp* 22,21 %, CF 0,44
$[VINODI] \rightarrow [DIAST2]$, *supp* 22,34 %, CF 0,42
$[LICDI] \rightarrow [DIAST2]$, *supp* 21,10 %, CF 0,43

C.2.4. Relaciones entre pliegues dérmicos y BMI

Para finalizar, y hasta donde nosotros podemos afirmar, por medio de nuestra metodología no fue posible hallar una correlación entre los atributos referidos a los pliegues de piel y el índice de masa corporal. Las dependencias que obtuvimos tan sólo muestran una relación unidireccional entre los atributos $TRIC$ (pliegues sobre el músculo tríceps) y $SUBSC$ (pliegues bajo el músculo subescapular) y el BMI, tales como las siguientes,

$[TRIC] \rightarrow [BMI]$, *supp* 15,85 %, CF 0,54
$[SUBSC] \rightarrow [BMI]$, *supp* 17,28 %, CF 0,58

Los resultados obtenidos son similares en los tres grupos de pacientes.

C.3. Experimentos sobre datos de color de suelos

De forma similar al caso anterior, hemos desplazado aquí la descripción de los atributos involucrados en la base de datos sobre color de suelos sobre la que realizamos nuestra experimentación en el capítulo 4.

En primer lugar, encontramos una descripción de los atributos de los que constaba la relación, indicando el tipo de éstos. A continuación, mostramos las tablas con las relaciones de similitud que se definieron sobre los atributos de tipo categórico, junto con los conjuntos de etiquetas lingüísticas definidos sobre el dominio subyacente de los atributos numéricos. Todos estos valores fueron suministrados por expertos edafólogos en una fase previa al análisis de los datos.

Tabla C.17: Atributos considerados en la base de datos de color de suelos

Columna	**Descripción**	**Tipo**
cod_perf	Código de perfil	Numérico
cod_ecol	Código de mesoambiente	Categórico
cod_hori	Código de horizonte	Numérico
faoreduc	Clave FAO reducida	Categórica
tipo_hor	Tipo de horizonte	Categórica
orientac	Orientación	Categórica
fisiogra	Fisiografía	Categórica
pendient	Pendiente(%)	Numérica
vegetaci	Vegetación	Categórica
material	Material original	Categórica
pmedia	Precipitación media anual	Numérica
tmedia	Temperatura anual media	Numérica
altitud	Altitud	Numérica
profundi	Profundidad efectiva	Numérica
grado_de	Grado de erosión	Categórica
hue_hume	Hue húmedo	Categórica
value_hu	Value húmedo	Categórica
croma_hu	Chroma húmedo	Categórica
hue_seco	Hue seco	Categórica
value_se	Value seco	Categórica
croma_se	Chroma seco	Categórica
tipo_est	Tipo de estructura	Categórica
clase_es	Clase de estructura	Numérica
grado_es	Grado de estructura	Categórica
arena	% Arena	Numérica
arcilla	%Arcilla	Numérica
co	% Carbono Orgánico	Numérica
carbonat	% Carbonato Cálcico	Numérica
ph	pH	Numérica
agua	Agua Útil	Numérica
fe	% Hierro Total	Numérica
cec	CEC	Numérica

Tabla C.18: Atributos de la base de datos de color de suelos, agrupados de acuerdo a su semántica

Grupo semántico	**Atributo**	**Comentarios**
Estaciones ambientales	Mesoambiente	Son combinaciones multidimensionales de factores ambientales que definen espacios más o menos homogéneos de influencia en el desarrollo posterior del suelo. A los factores ambientales se les ha denominado factores formadores del suelo.
Factores formadores	Altitud	Los factores formadores no forman parte del individuo suelo y no son partes o componentes de su estructura. Son factores ambientales generales, susceptibles de ser medidos, que actúan como agentes causales de los procesos edafogenéticos que conducen al desarrollo del suelo.
	Precipitación media anual	
	Temperatura media anual	
	Material original	
Horizontes	Tipo de horizonte	Expresan las características de zonas homogéneas dentro del suelo y son resultado final de una serie de procesos edafogenéticos y de la actuación de los agentes (factores) formadores.
Componentes	% Arena	Los componentes y propiedades son características, morfológicas o analíticas, susceptibles de ser medidas o descritas en cada horizonte; pueden actuar como diagnósticos del mismo. En el aspecto de la génesis, las propiedades son consecuencia de los componentes.
	% Arcilla	
	% Carbono orgánico	
	% Hierro libre	
Propiedades	Value	
	Chroma	
	Hue	

Tabla C.19: Relaciones de similitud (Atributo FAOREDUC)

FAOREDUC	2	3	4	5	6	7	8	9	10	11	12	13
1	0.3	0.3	0.5	0.3	0.3	0.3	0.3	0.5	0.3	0.5	0.3	0.5
2		0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.5	0.3
3			0.3	0.5	0.3	0.3	0.5	0.3	0.3	0.3	0.3	0.3
4				0.3	0.3	0.3	0.3	0.5	0.3	0.5	0.3	0.5
5					0.3	0.3	0.5	0.3	0.3	0.3	0.3	0.3
6						0.3	0.3	0.3	0.3	0.3	0.3	0.3
7							0.3	0.3	0.3	0.3	0.3	0.3
8								0.3	0.3	0.3	0.3	0.3
9									0.3	0.5	0.3	0.5
10										0.3	0.3	0.3
11											0.3	0.5
12												0.3

Tabla C.20: Códigos para el atributo FAOREDUC

Valor	clave
Arenosol	1
Cambisol	2
Chernozems	3
Fluvisol	4
Kastanozems	5
Litosol	6
Luvisol	7
Phaeozems	8
Regosol	9
Rendzina	10
Solonchack	11
Xerosol	12
Yermosol	13

Tabla C.21: Relaciones de similitud (Atributo TIPO_HOR)

TIPO_HOR	Ah	Ap	Bk	Bt	Btk	Bw	Bwk	C	Ck
A	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3
Ah		0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3
Ap			0.3	0.3	0.3	0.3	0.3	0.3	0.3
Bk				0.3	0.3	0.3	0.3	0.3	0.3
Bt					0.3	0.3	0.3	0.3	0.3
Btk						0.3	0.3	0.3	0.3
Bw							0.3	0.3	0.3
Bwk								0.3	0.3
C									0.3

Tabla C.22: Relaciones de similitud (Atributo FISIOGRA)

FISIOGRA	Fondo ladera	Ladera	Cima	Meseta
Llano	0.5	0.2	0.2	0.2
F. lad.		0.2	0.2	0.2
Ladera			0.2	0.2
Cima				0.5

Tabla C.23: Etiquetas lingüísticas (Atributo PENDIENT)

Etiqueta	α	β	γ	δ
Flat	0	0	1	2
Gently sloping	2	3	4	5
Sloping	5	6	9	10
Strongly sloping	10	11	14	15
Moderately steep	15	16	29	30
Steep	30	31	59	60
Very steep	60	61	100	100

Tabla C.24: Relaciones de similitud (Atributo VEGETACI)

$VEGETACI$	2	3	4	5	6	7	8
1	0.5	0.5	0.2	0.2	0.2	0.2	0.2
2		0.5	0.2	0.2	0.2	0.2	0.2
3			0.2	0.2	0.2	0.2	0.2
4				0.5	0.5	0.5	0.5
5					0.5	0.5	0.5
6						0.5	0.5
7							0.5

Tabla C.25: Códigos para el atributo VEGETACI

Valor	clave
Bosque natural	1
Bosque de repoblación	2
Matorral alto	3
Herbácea	4
Matorral bajo degradado	5
Cultivo arbolado	6
Cultivo herbáceo	7
Regadío	8

Tabla C.26: Relaciones de similitud (Atributo MATERIAL)

$MATERIAL$	2	3	4	5	6	7	8	9
1	0.2	0.2	0.2	0.5	0.2	0.2	0.2	0.0
2		0.2	0.2	0.2	0.5	0.2	0.2	0.0
3			0.2	0.2	0.2	0.2	0.2	0.0
4				0.2	0.2	0.2	0.2	0.0
5					0.2	0.2	0.2	0.0
6						0.2	0.2	0.0
7							0.2	0.0
8								0.0

Tabla C.27: Códigos para el atributo MATERIAL

Valor	clave
Ácido coluvial	1
Ácido aluvial	2
Ácido sobre mat. compacto	3
Ácido sobre mat. no compacto	4
Calcáreo coluvial	5
Calcáreo aluvial	6
Calcáreo sobre mat. compacto	7
Calcáreo sobre mat. no compacto	8
Roca volcánica	9

Tabla C.28: Etiquetas lingüísticas (Atributo PMEDIA)

Etiqueta	α	β	γ	δ
Baja	183	183	315	490
Media	490	664	731	818
Alta	818	905	1287	1287

Tabla C.29: Etiquetas lingüísticas (Atributo TMEDIA)

Etiqueta	α	β	γ	δ
Baja	0	0	6.5	8.5
Media	8.5	10.5	12.5	14.7
Alta	14.7	16.9	21.0	21.0

Tabla C.30: Etiquetas lingüísticas (Atributo ALTITUD)

Etiqueta	α	β	γ	δ
Baja	65	65	380	860
Media	860	1341	1460	1700
Alta	1700	1940	3020	3020

Tabla C.31: Etiquetas lingüísticas (Atributo PROFUNDI)

Etiqueta	α	β	γ	δ
Baja	2	2	12	17
Media	17	22	28	37
Alta	37	45	66	66

Tabla C.32: Relaciones de similitud (Atributo GRADO)

GRADO	Moderate	Severe
Slight	0.5	0.5
Moderate		0.5

Tabla C.33: Relaciones de similitud (Atributo HUE_HUME)

HUE_HUME	1.25YR	2.5YR	3.75YR	6.75YR	7.5YR	8.75YR	10YR	1.25Y	2.5Y	5Y	10Y
10R	0.3	0.3	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
1.25YR		0.3	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
2.5YR			0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
3.75YR				0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
6.75YR					0.5	0.3	0.3	0.3	0.3	0.1	0.0
7.5YR						0.3	0.3	0.3	0.3	0.1	0.0
8.75YR							0.3	0.3	0.3	0.1	0.0
10YR								0.3	0.3	0.1	0.0
1.25Y									0.3	0.1	0.0
2.5Y										0.1	0.0
5Y											0.0

Tabla C.34: Relaciones de similitud (Atributo VALUE_HU)

VALUE_HU	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	8
2	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.2
2.5		0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.2
3			0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.2
3.5				0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.2
4					0.3	0.3	0.3	0.3	0.3	0.3	0.2
4.5						0.3	0.3	0.3	0.3	0.3	0.2
5							0.3	0.3	0.3	0.3	0.2
5.5								0.3	0.3	0.3	0.2
6									0.3	0.3	0.2
6.5										0.3	0.2
7											0.2

Tabla C.35: Relaciones de similitud (Atributo CROMA_HU)

CROMA_HU	0.5	1	1.5	2	2.5	3	3.5	4	5	6	7	8
0	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.2	0.2	0.2	0.2
0.5		0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.2	0.2	0.2	0.2
1			0.3	0.3	0.3	0.3	0.3	0.3	0.2	0.2	0.2	0.2
1.5				0.3	0.3	0.3	0.3	0.3	0.2	0.2	0.2	0.2
2					0.3	0.3	0.3	0.3	0.2	0.2	0.2	0.2
2.5						0.3	0.3	0.3	0.2	0.2	0.2	0.2
3							0.3	0.3	0.2	0.2	0.2	0.2
3.5								0.3	0.2	0.2	0.2	0.2
4									0.2	0.2	0.2	0.2
5										0.2	0.2	0.2
6											0.2	0.2
7												0.2

Tabla C.36: Relaciones de similitud (Atributo HUE_SECO)

HUE_SECO	2.5YR	3.75YR	5YR	6.75YR	7.5YR	8.75YR	10YR	1.25Y	2.5Y	5Y
10R	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
2.5YR		0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.1
3.75YR			0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.1
5YR				0.3	0.3	0.3	0.3	0.3	0.3	0.1
6.75YR					0.5	0.3	0.3	0.3	0.3	0.1
7.5YR						0.3	0.3	0.3	0.3	0.1
8.75YR							0.3	0.3	0.3	0.1
10YR								0.3	0.3	0.1
1.25Y									0.3	0.1
2.5Y										0.1

Tabla C.37: Relaciones de similitud (Atributo VALUE_SE)

$VALUE\_SE$	4	4.5	5	5.5	6	6.5	7	7.5	8	8.5
3	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
4		0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3
4.5			0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3
5				0.3	0.3	0.3	0.3	0.3	0.3	0.3
5.5					0.3	0.3	0.3	0.3	0.3	0.3
6						0.3	0.3	0.3	0.3	0.3
6.5							0.3	0.3	0.3	0.3
7								0.3	0.3	0.3
7.5									0.3	0.3
8										0.3

Tabla C.38: Relaciones de similitud (Atributo CROMA_SE)

$CROMA\_SE$	1	1.5	2	2.5	3	3.5	4	4.5	5	6	7	7.5	8
0	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3
1		0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.3	0.3	0.3	0.3
1.5			0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.3	0.3	0.3	0.3
2				0.5	0.5	0.5	0.5	0.5	0.5	0.3	0.3	0.3	0.3
2.5					0.5	0.5	0.5	0.5	0.5	0.3	0.3	0.3	0.3
3						0.5	0.5	0.5	0.5	0.3	0.3	0.3	0.3
3.5							0.5	0.5	0.5	0.3	0.3	0.3	0.3
4								0.5	0.5	0.3	0.3	0.3	0.3
4.5									0.5	0.3	0.3	0.3	0.3
5										0.3	0.3	0.3	0.3
6											0.3	0.3	0.3
7												0.5	0.5
7.5													0.5

Tabla C.39: Relaciones de similitud (Atributo TIPO_ES)

$TIPO\_ES$	2	3	4	5	6	7	8	9
1	0.4	0.4	0.4	0.4	0.2	0.2	0.2	0.2
2		0.4	0.4	0.4	0.2	0.2	0.2	0.2
3			0.4	0.4	0.2	0.2	0.2	0.2
4				0.4	0.2	0.2	0.2	0.2
5					0.2	0.2	0.2	0.2
6						0.2	0.2	0.2
7							0.2	0.2
8								0.2

Tabla C.40: Códigos para el atributo TIPO_ES

Valor	clave
Granular	1
Migajosa	2
Subangular blocky	3
Angular blocky	4
Prismatic	5
Platy	6
Rock structure	7
Massive	8
Single grain	9

Tabla C.41: Etiquetas lingüísticas (Atributo CLASE_ES)

Etiqueta	α	β	γ	δ
Very fine	0	0	0.75	1.0
Fine	1.0	1.25	1.75	2.0
Medium	2.0	2.25	4.75	5.0
Coarse	5.0	5.25	9.75	10.0
Very coarse	10.0	10.25	20.0	20.0

Tabla C.42: Relaciones de similitud (Atributo GRADO_ES)

GRADO_ES	Weak	Moderate	Strong
Very Weak	0.3	0.3	0.3
Weak		0.3	0.3
Moderate			0.3

Tabla C.43: Etiquetas lingüísticas (Atributo ARENA)

Etiqueta	α	β	γ	δ
Baja	0.4	0.4	21.2	30.6
Media	30.6	40.0	48.9	56.4
Alta	56.4	63.9	91	91

Tabla C.44: Etiquetas lingüísticas (Atributo ARCILLA)

Etiqueta	α	β	γ	δ
Baja	1.31	1.31	10.0	15.0
Media	15.0	20.0	26.0	33.1
Alta	33.1	40.1	69.5	69.5

Tabla C.45: Etiquetas lingüísticas (Atributo CO)

Etiqueta	α	β	γ	δ
Baja	0	0	0.37	0.57
Media	0.57	0.77	1.40	1.94
Alta	1.94	2.48	19.5	19.5

Tabla C.46: Etiquetas lingüísticas (Atributo CARBONAT)

Etiqueta	α	β	γ	δ
Baja	0.00	0.00	8.2	15.75
Media	15.75	23.3	31.0	46.4
Alta	46.4	61.8	85.60	85.60

Tabla C.47: Etiquetas lingüísticas (Atributo PH)

Etiqueta	α	β	γ	δ
Baja	0.37	0.37	5.60	6.35
Media	6.35	7.10	7.50	7.85
Alta	7.85	8.20	8.90	8.90

Tabla C.48: Etiquetas lingüísticas (Atributo arena)

Etiqueta	α	β	γ	δ
Baja	0.06	0.06	0.8	1.1
Media	1.1	1.40	1.70	2.0
Alta	2.0	2.30	8.6	8.6

Tabla C.49: Etiquetas lingüísticas (Atributo FE)

Etiqueta	α	β	γ	δ
Baja	0.0	0.0	0.9	1.25
Media	1.25	1.60	2.1	2.9
Alta	2.9	3.7	5.70	5.70

Tabla C.50: Etiquetas lingüísticas (Atributo CEC)

Etiqueta	α	β	γ	δ
Baja	0.26	0.26	6.54	9.01
Media	9.01	11.48	17.21	25.11
Alta	25.11	33.0	53.20	53.20

C.4. Descripción de las particiones sobre tipos de suelos

En este apartado complementamos el capítulo 5, y en él incluiremos las tablas descriptivas de los distintos tipos de suelos atendiendo a las diferentes clasificaciones de los mismos de acuerdo con criterios científicos o de usuario. Dichas tablas pueden utilizarse para interpretar y comprender mejor las correspondencias que se hallaron en el apartado de experimentación del capítulo correspondiente.

Tabla C.51: Tabla grupos (I)

Unidad	Descripción
1	Profundo, lastra no detectada, alta tractorabilidad, algo inclinado, pedregosidad baja, surcos no detectados, pérdida no detectada, arcilloso, pardo oscuro
2	Profundo, lastra no detectada, alta tractorabilidad, llano, pedregosidad baja, surcos no detectados, pérdida no detectada, arcilloso, pardo claro
3	Profundo, lastra no detectada, alta tractorabilidad, algo inclinado, pedregosidad baja, surcos presentes, pérdida no detectada, arcilloso, gris oscuro
4	Profundo, lastra no detectada, alta tractorabilidad, algo inclinado, pedregosidad baja, surcos no detectados, pérdida no detectada, arcilloso, gris oscuro
5	Profundo, lastra no detectada, baja tractorabilidad, algo inclinado, pedregosidad baja, surcos presentes, pérdida detectada, arcilloso, gris oscuro
6	Profundidad media, lastra media profundidad, media tractorabilidad, algo inclinado, pedregosidad media, surcos no detectados, pérdida no detectada, franco, pardo oscuro
7	Profundidad media, lastra media profundidad, media tractorabilidad, algo inclinado, pedregosidad media, surcos no detectados, pérdida no detectada, arcilloso, pardo claro
8	Profundidad media, lastra profunda, alta tractorabilidad, algo inclinado, pedregosidad baja, surcos no detectados, pérdida no detectada, franco, pardo oscuro
9	Somero, lastra somera, alta tractorabilidad, algo inclinado, pedregosidad alta, surcos no detectados, pérdida no detectada, arenoso, gris claro
10	Profundo, lastra no detectada, media tractorabilidad, algo inclinado, pedregosidad alta, surcos presentes, pérdida detectada, franco, pardo oscuro

Tabla C.52: Tabla grupos (II)

11	Profundo, lastra no detectada, baja tractorabilidad, inclinado, pedregosidad media, surcos no detectados, pérdida no detectada, franco, pardo claro
12	Profundidad media, lastra media profundidad, baja tractorabilidad, algo inclinado, pedregosidad alta, surcos presentes, pérdida detectada, arcilloso, rojo
13	Profundidad media, lastra no detectada, media tractorabilidad, inclinado, pedregosidad alta, surcos presentes, pérdida detectada, arenoso, pardo oscuro
14	Profundo, lastra no detectada, baja tractorabilidad, muy inclinado, pedregosidad media, surcos presentes, pérdida detectada, franco, gris claro
15	Profundidad media, lastra media profundidad, alta tractorabilidad, llano, pedregosidad baja, surcos no detectados, pérdida no detectada, arcilloso, pardo claro
16	Profundidad media, lastra media profundidad, baja tractorabilidad, inclinado, pedregosidad media, surcos presentes, pérdida detectada, arcilloso, gris oscuro
17	Profundidad media, lastra media profundidad, alta tractorabilidad, algo inclinado, pedregosidad baja, surcos presentes, pérdida detectada, arcilloso, gris oscuro
18	Profundidad media, lastra somera, baja tractorabilidad, muy inclinado, pedregosidad media, surcos presentes, pérdida detectada, arcilloso, gris claro
19	Profundo, lastra profunda, baja tractorabilidad, muy inclinado, pedregosidad alta, surcos presentes, pérdida detectada, arenoso, gris oscuro

Tabla C.53: Tabla codunida

Unidad	Descripción
1	Litosoles
2	Fluvisoles calcáreos
3	Fluvisoles eútricos
5	Regosoles calcáreos
6	Regosoles calcáreos-Regosoles eútricos
8	Regosoles calcáreos-Xerosoles cálcicos
9	Regosoles calcáreos-Cambisoles calcicos
10	Regosoles calcáreos-Litosoles-Cambisoles cálcicos
11	Regosoles calcáreos-Regosoles eútricos-Cambisoles eútricos
12	Xerosoles cálcicos-Xerosoles gípsicos-Cambisoles cálcicos
14	Cambisoles cálcicos
15	Cambisoles cálcicos-Regosoles calcáreos
16	Cambisoles cálcicos-Luvisoles crómicos
17	Cambisoles cálcicos-Regosoles calcáreos-Litosoles
19	Cambisoles eútricos-Regosoles eútricos-Luvisoles crómicos
20	Cambisoles eútricos-Luvisoles crómicos
21	Luvisoles crómicos

Tabla C.54: Tabla exp13

Unidad	Descripción
1	Suelos sobre roca carbonatada compacta, de poco espesor, de evolución variable y alta pendiente
2	Suelos sobre material aluvial calcáreo reciente
3	Suelos sobre material aluvial no calcáreo reciente
4	Suelos sobre margas, de espesor moderado-alto, pendientes moderadas a fuertes y baja evolución
5	Suelos sobre margocalizas, filitas, conglomerados y coluvios, espesor moderado, pendientes fuertes y baja evolución
6	Suelos sobre margas, espesor alto, pendientes moderadas, sin horizonte cálcico y evolución media
7	Suelos sobre margocalizas, filitas, conglomerados y coluvios, espesor moderado-alto, pendientes moderadas, horizonte cálcico y evolución media
8	Suelos sobre margas gípsicas de bajo o moderado grado de evolución
9	Suelos de evolución moderada sobre coluvios o conglomerados, pendientes bajas, espesor moderado y horizonte cálcico o petrocálcico
10	Suelos de evolución alta sobre coluvios o conglomerados, pendientes bajas, espesor moderado-alto y horizonte cálcico o petrocálcicos
11	Suelos muy evolucionados, espesor alto, sin horizonte petrocálcico
12	Suelos sobre micaesquistos y cuarcitas de escasa evolución, bajo-medio espesor y altas pendientes
13	Suelos sobre micaesquistos y cuarcitas de moderada-alta evolución, espesor medio y pendientes moderadas-bajas

Tabla C.55: Tabla grupos4

Unidad	Descripción
1	Profundo, lastra media profundidad, alta tractorabilidad, algo inclinado, pedregosidad baja, surcos no detectados, pérdida no detectada, arcilloso, gris claro
2	Profundidad media, lastra no detectada, baja tractorabilidad, inclinado, pedregosidad media, surcos presentes, pérdida detectada, franco, gris oscuro
3	Profundo, lastra profunda, media tractorabilidad, algo inclinado, pedregosidad media, surcos no detectados, pérdida no detectada, franco, pardo claro
4	Profundidad media, lastra profunda, alta tractorabilidad, algo inclinado, pedregosidad media, surcos presentes, pérdida detectada, arcilloso, gris claro

Tabla C.56: Tabla grunida

Unidad	Descripción
1	Litosoles
2	Fluvisoles
3	Regosoles
4	Xerosoles
5	Cambisoles
6	Luvisoles

Tabla C.57: Tabla exp5

Unidad	Descripción
1	Suelos sobre roca carbonatada compacta
2	Suelos sobre materiales aluviales recientes
3	Suelos sobre margas, margocalizas, filitas, conglomerados y coluvios, materiales carbonatados o recarbonatados, pendientes moderadas a fuertes y baja a moderada evolución.
4	Suelos sobre conglomerados y coluvios en pendientes de moderadas a bajas, de moderada o muy alta evolución
5	Suelos sobre micaesquistos y cuarcitas

Bibliografía

[Agrawal et al., 1993] Agrawal R., Imielinski T., Swami A. *Mining Association Rules between Sets of Items in Large Databases*. Proc. of the 1993 ACM SIGMOD Conf., Washington DC, USA.

[Agrawal y Srikant, 1995] Agrawal R., Srikant R. *Fast Algorithms for Mining Association Rules*. Procs. of the 11th ICDE, Marzo 1995.

[Appriou et al., 2001] Appriou A., Ayoun A., Benferhat S., Besnard P., Cholvy L., Cooke R., Cuppens F., Dubois D., Fargier H., Grabisch M., Kruse R., Lang J., Moral S., Prade H., Saffiotti A., Smets P., and Sossai C. *Fusion: General concepts and characteristics. Int. Journal of Intelligent Systems*, 16(10):1107–34, 2001.

[Aranda et al., 2003] Aranda V., Calero J., Delgado G., Sánchez D., Serrano J.M., Vila M.A. *Using Data Mining Techniques to Analyze Correspondences Between User and Scientific Knowledge in an Agricultural Environment*. En M. Piattini, J. Filipe and J. Braz (eds.) *Enterprise Information Systems IV*, pp. 75–89, Kluwer Academic Publishers, 2003.

[Aranda et al., 2002] Aranda V., Calero J., Delgado G., Sánchez D., Serrano J., and Vila M.A. *Flexible land classification for olive cultivation using user knowledge.* In *Proceedings of 1st. Int. ICSC Conf. On Neuro-Fuzzy Technologies (NF'2002)*, La HaBana (Cuba), 16-19 Enero 2002.

[Backer, 1975] Backer E.A. *Nonstatistical Type of Uncertainty in Fuzzy Events*, Colloq. Math. Soc. János Bolyai, 1975.

[Benzécri, 1963] Benzécri J.P. *Cours de Linguistique Mathématique*. Rennes: Université de Rennes, 1963.

[Berzal et al., 2001a] Berzal F., Blanco I., Sánchez D., and Vila M. *A new framework to assess association rules*. In Hoffmann, F., editor, *Advances in Intelligent Data Analysis. Fourth International Symposium, IDA'01. Lecture Notes in Computer Science 2189*, pages 95–104. Springer-Verlag, 2001.

[Berzal et al., 2003] Berzal F., Blanco I., Sánchez D., Serrano J.M., and Vila M. *A definition for fuzzy approximate dependencies*. Sometido para su publicación en Fuzzy Sets and Systems.

[Berzal et al., 2001b] Berzal F., Cubero J.C., Marín N., Serrano J.M. *TBAR: An efficient method for association rule mining in relational databases*. Data & Knowledge Engineering, 37 pp 47–64, 2001.

[Bezdek y Harris, 1978] Bezdek J.C., Harris J.D. *Fuzzy Partitions and Relations: An Axiomatic Basis for Clustering*. Fuzzy Sets and Systems 1, 111–127, 1978.

[Bezdek y Pal, 1992] Bezdek J.C., Pal S.K., Eds. *Fuzzy Models for Pattern Recognition*, IEEE Press, New York, 1992.

[Bigham y Ciolkosz, 1993] Bigham J.M., Ciolkosz E.J., eds. *Soil color*. Soil Sci. Soc. Am. Spec. Publ. No. 31, 159 pp., 1993.

[Blanco et al., 2000] Blanco I., Martín-Bautista M.J., Sánchez D., Vila, M.A. *On the support of dependencies in relational databases: Strong approximate dependencies*. Data Mining and Knowledge Discovery. Sometido.

[Blanco et al., 2002] Blanco I., Sánchez D., Serrano J.M, Vila M.A., Galindo J. *FuzzyQueries 2+, una herramienta para la integración de consultas flexibles, cálculo de agregaciones y resúmenes, y extracción de conocimiento*. Actas del XI Congreso Español sobre Tecnologías y Lógica Fuzzy, ESTYLF'2002, pp. 337-342. León (España), 17 al 20 de Septiembre, 2002.

[Blanco et al., 2003] Blanco I., Sánchez S., Serrano J.M., Vila M.A. *A new proposal of aggregation function: the linguistic summary*. Sometido para

el International Fuzzy System Association World Congress, IFSA 2003. Del 29 Junio al 2 de Julio, 2003. Estambul (Turquía).

[Blanco, 2001] Blanco I. (2001). *Deducción en Bases de Datos Relacionales Difusas*. Tesis Doctoral. Departamento de Ciencias de la Computación e Inteligencia Artificial. Universidad de Granada.

[Bosc et al., 1994] Bosc P., Pivert O., Farquhar K. *Integrating Fuzzy Queries into an Existing Database Management System: An Example*. International Journal of Intelligent Systems 9, pp. 475–492, 1994.

[Bosc et al., 1996] Bosc P., Dubois D., Prade H. *More results on functional dependencies and quotient operators in fuzzy databases*. Technical Report IRIT/96-10-R, Institute de Recherche en Informatique de Toulouse, Marzo 1996.

[Bosc et al., 1997] Bosc P., Lietard L., Pivert O. *Functional Dependencies Revisited Under Graduality and Imprecision*. Annual Meeting of NAFIPS, pp. 57–62, 1997.

[Boulicaut et al, 1998] Boulicaut J.F., Klementtinen and Mannila H. *Querying inductive databases: A case study on the MINE RULE operator*. In Proc. II European Symp. on Principles of Data Mining and Knowledge Discovery (PKDD'98), Nantes (F), 23-26 September 1998.

[Bra y Paredaens, 1983] Bra P.D. y Paredaens J. *Horizontal decompositions for handling exceptions to functional dependencies*. Advances in Database Theory, 2:123–144, 1983.

[Brin et al., 1997] Brin S., Motwani R., Ullman J.D., and Tsur S. *Dynamic itemset counting and implication rules for market basket data*. SIGMOD Record vol. 26, no. 2, pp. 255–264, 1997.

[Buckles y Petry, 1982] Buckles B.P., and Petry F.E. *A Fuzzy Representation of Data for Relational Databases*. Fuzzy Sets and Systems, 7. 213–226, 1982.

[Buckles y Petry, 1984] Buckles B.P., and Petry F.E. *Extending the Fuzzy Database with Fuzzy Numbers*. Information Sciences, 34. 145–155, 1984.

[Calero et al., 2003] Calero J., Delgado G., Sánchez-Marañón M., Sánchez D., Serrano J.M., Vila M.A. *Helping user to discover association rules. A case in soil color as aggregation of other soil properties*. Procs. of ICEIS'2003. Angers (Francia), 23–26 Abril 2003.

[Calero et al., 2002] Calero J.A., Serrano J.M, Aranda V., Sánchez D., Sánchez-Marañón M., Vila M.A., Delgado G. *Evaluación del olivar granadino con información del usuario empleando técnicas de análisis multivariante y data mining*. Proceedings of the 1st International IFOAM Scientific Conference on Ecological Olive Growing: Production and Culture. Puente Génave (España), 24–25 Mayo 2002.

[Chan y Au, 1997] Chan K.C.C. y Au W.-H. *Mining Fuzzy Association Rules*, in Proc. of the 6th Int'l Conf. on Information and Knowledge Management, pp. 209-215, Las Vegas, Nevada, 1997.

[Chen et al, 1992] Chen G., Kerre E.E., and Vandenbulcke J. *Fuzzy Functional Dependency and its Axiomatic System in a Fuzzy Relational Data Model*. In International Conference on Information Processing and Management of Uncertainty in Knowledge Based Systems, IPMU'92, pp. 313–317, 1992.

[Chen et al, 1991] Chen G., Vandenbulcke J., and Kerre E.E. *A Step Towards the Theory of Fuzzy Relational Database Design*. In Proceedings on Computer Management 91, pp. 44–47, 1991.

[Chien et al., 2001] Chien B.C., Lin Z.L., Hong T.P. *An Efficient Clustering Algorithm for Mining Fuzzy Quantitative Association Rules*. In Procs. of the Ninth International Fuzzy Systems Association World Congress, pp. 1306-1311, 2001.

[Cholvy y Moral, 2001] Cholvy L. and Moral S. *Merging databases: Problems and examples. Int. Journal of Intelligent Systems*, 16(10):1193–1221, 2001.

[Codd, 1970] Codd E.F. *A Relational Model of Data for Large Shared Data Banks*. Commun. ACM, 13(6): 377–387, 1970.

[Cohen, 1995] Cohen P.R., *Empirical Methods for Artificial Intelligence*. The MIT Press, 1995.

[Cox, 1994] Cox T.F. *Multidimensional scaling*. Col. Monographs on statistics and applied probability 59. Ed. Mark L. Davison. Chapman and Hall, 1994.

[Cubero et al., 1994a] Cubero J.C., Vila M.A. *A New Definition of Fuzzy Functional Dependency in Fuzzy Relational Databases*. Int. Journal on Intelligent Systems 9(5), pp. 441-448, 1994.

[Cubero et al., 1994b] Cubero J.C., Pons O., Vila M.A. *Weak and Strong Resemblances in Fuzzy Functional Dependencies*. Proc. IEEE Int. Conf. on Fuzzy Systems, Orlando/FL, USA, pp. 162-166, 1994.

[Cubero et al., 1995] Cubero J.C., Medina J.M., Pons O., Vila M.A. *The generalized selection: An alternative way for the quotient operations in fuzzy relational databases. Fuzzy Logic and Soft Computing*, B. Bouchon-Meunier, R. Yager, and L.A. Zadeh, Eds. World Scientific Press, 1995.

[Cubero et al., 1998a] Cubero J.C., Cuenca F., Blanco I., Vila M.A. *Incomplete Functional Dependencies versus Knowledge Discovery in Databases*. Proc. of EUFIT'98, Aachen, Germany, pp. 731-74, 1998.

[Cubero et al., 1998b] Cubero J.C., Medina J.M., Pons O., Vila M.A. *Fuzzy loss less decompositions in databases*. Fuzzy Sets and Systems, 97(2):145–167, 1998.

[Date, 1990] Date C.J. *An Introduction to Database Systems Vol I. (5.ed.)*. Addison Wesley, 1990.

[DeLuca y Termini, 1972] De Luca A. and Termini S. *A definition of non-probabilistics entropy in the setting of fuzzy set theory*. Information and Control 20 pp.301-312, 1972.

[Delgado et al, 1994] Delgado M., Verdegay J.L., Vila M.A. *Decision Making Models*. International Journal of Intelligent Systems, 9(4):365-378, 1994.

[Delgado et al, 1999a] Delgado M., Sánchez D., Vila M.A. *Fuzzy Quantified Dependencies in Relational Databases.* EUFIT'99, Aachen (Germany), 1999.

[Delgado et al, 1999b] Delgado M., Sánchez D., Vila M.A. *Mining Approximate Dependencies Using Association Rules Measures.* Procs of the IDA'99.

[Delgado et al., 2000a] Delgado M., Martín-Bautista M.J., Sánchez D., Vila M.A. *Mining strong approximate dependencies from relational databases.* In Proceedings of IPMU'2000.

[Delgado et al., 2000b] Delgado M., Sánchez D., Vila M.A. *Fuzzy cardinality based evaluation of quantified sentences.* International Journal of Approximate Reasoning, vol. 23, pp. 23-66, 2000.

[Delgado et al., 2000c] Delgado M., Sánchez D., Serrano J.M., Vila M.A. *A Survey of methods to evaluate quantified sentences.* Mathware and Soft Computing, Vol. VII, 2-3, 2000.

[Delgado et al., 2002] Delgado M., Martín-Bautista M.J., Sánchez D., Vila M.A. *A Probabilistic Definition of a Nonconvex Fuzzy Cardinality.* Fuzzy Sets and Systems 126 (2), pp. 41-54, 2002.

[Delgado et al., 2003a] Delgado M., Marín N., Sánchez D., Vila M.A. *Fuzzy Association Rules: General Model and Applications.* IEEE Transactions on Fuzzy Systems 11 (2), pp. 214-225, 2003.

[Díaz-Hermida et al., 2003] Díaz-Hermida F., Cariñena P., Bugarín A., Barro S. *Definition and classification of semi-fuzzy quantifiers for the evaluation of fuzzy quantified sentences.* Aceptado en International Journal of Approximate Reasoning, 2003.

[Dubes y Jain, 1988] Dubes R., Jain A. *Algorithms that Cluster Data*, Prentice-Hall, Englewood Cliffs, N.J., 1988.

[Dubois y Prade, 1979] Dubois D., and Prade H. *Fuzzy Sets and Systems: Theory and Applications.* Academic Press, N.Y., 1979.

[Dubois y Prade, 1985] Dubois D., and Prade H. *Fuzzy cardinality and the modeling of imprecise quantification*. Fuzzy Sets and Systems 16 pp. 190-230, 1985.

[Dubois y Prade, 1988] Dubois D., and Prade H. *Possibility Theory. An Approach to Computerized Processing of Uncertainty*. Plenum Press, N.Y., 1988.

[Dunn, 1974] Dunn J. *A graph theoretic analysis of pattern classification via Tamura's fuzzy relation*. IEEE Trans. SMC-4, 3, 310–313, 1974.

[Escofier, 1992] Escofier B. *Análisis factoriales simples y múltiples : objetivos, métodos e interpretación*. Bilbao, Universidad del País Vasco, 1992.

[FAO, 1968] FAO 1968. *Definitions of soil units for the soil maps of the world.* Technical report, FAO. World Soil Resources Reports 33.

[FAO, 1998] FAO 1998. *The world reference base for soil resources*. World Soil Resources Rep. 84. ISSS/AISS/IBG/ISRIC/FAO, Rome.

[Fayyad et al, 1995] Fayyad U.M., and Uthurusamy R. (eds.) *Procs. of the 1st Int. Conference on Knowledge Discovery and Data Mining*. Montreal, 1995.

[Frawley et al, 1991] Frawley W.J., Piatetsky-Shapiro G., and Matheus C.J. *Knowledge discovery databases: An Overview*. In Piatetsky-Shapiro and Frawley (eds) *Knowledge Discovery in Databases*, AAAI/MIT, 1–27, 1991.

[Fukami et al, 1979] Fukami S., Umano M., Muzimoto M., Tanaka H. *Fuzzy Database Retrieval and Manipulation Language*, IEICE Technical Reports, vol. 78, 233, pp. 65–72, AL-78–85 (Automata and Language), 1979.

[Galindo, 1999] Galindo J. *Tratamiento de la Imprecisión en Bases de Datos Relacionales: Extensión del Modelo y Adaptación de los SGBD Actuales*. Tesis Doctoral. Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada, 1999.

[Galindo et al., 1997] Galindo J., Medina J.M., Aranda M.C. *Una solución al problema de la División Relacional Difusa*. VII Congreso Español sobre Tecnologías y Lógica Fuzzy, ESTYLF'97, pp. 51–56, Tarragona (España), Septiembre 1997.

[Galindo et al, 1998] Galindo J., Medina J.M., Pons O., Cubero J.C. *A Server for Fuzzy SQL Queries*, en *Flexible Query Answering Systems*, eds. T. Andreasen, H. Christiansen and H.L. Larsen, Lecture Notes in Artificial Intelligence (LNAI) 1495, pp. 164-174. Ed. Springer, 1998. International Conference on Flexible Query Answering Systems, FQAS'98, Roskilde (Denmark), May 1998.

[Glöckner, 2003a] Glöckner I. *Evaluation of Quantified Propositions in Generalized Models of Fuzzy Quantification*. Submitted to the International Journal on Approximate Reasoning, Elsevier Science, 15th January 2003.

[Glöckner, 2003] Glöckner I. *Fuzzy Quantifiers, Multiple Variable Binding and Branching Quantification*. Procs. of IFSA 2003, Mini-Track on Cardinality, Quantification and Aggregation on Fuzzy Sets and Fuzzy Bags, Estambul (Turquía) Julio 2003.

[Hair et al, 1999] Hair J.F., Anderson R.E., Tatham R.L., Black W.C. *Análisis Multivariante*. 5^a ed. Prentice Hall Iberia, Madrid, 1999.

[Han et al, 2000] Han J., Pei J., Yin Y. *Mining frequent patterns without candidate generation*. In Proc. 2000 ACM SIGMOD Int. Conf. On Management of Data, Dallas, TX, USA, pp. 1–12, 2000.

[Hidber, 1999] Hidber C. *Online association rule mining*. In Proc. 1999 ACM SIGMOD Int. Conf. On Management of Data, pp. 145–156, 1999.

[Hipp et al., 2000] Hipp J., Güntzer U., Nakhaeizadeh G. *Algorithms for association rule mining - a general survey and comparison*. SIGKDD Explorations, vol. 2, no. 1, pp. 58–64, 2000.

[Hong et al., 1999] Hong T.P., Kuo C.S., Chi S.C. *A Fuzzy Data Mining Algorithm for Quantitative Values*. Procs. of the Third International Confer-

ence on Knowledge-Based Intelligent Information Engineering Systems, pp. 480–483, Adelaide, Australia, 1999.

[Houtsma y Swami, 1993] Houtsma M., Swami A. *Set-oriented mining of association rules*. Research Report RJ 9567, IBM Almaden Research Center, San Jose, California, Octubre 1993.

[Huhtala et al., 1998] Huhtala Y., Karkkainen J., Porkka P., Toivonen H. *Efficient Discovery of Functional and Approximate Dependencies using Partitions*. Proc. of the 14th Int. Conference on Data Engineering, pp. 392–401, 1998.

[Hussain et al., 1999] Hussain, F., Liu, H., Tan, C.L., and Dash, M. (1999). Discretization: An Enabling Technique. Technical Report, The National University of Singapore, June 1999.

[Imielinski y Mannila, 1996] Imielinski T., Mannila H. *A database perspective on knowledge discovery*. Communications of the ACM, 39:11, 1996.

[Kandel y Yelowitz, 1974] Kandel A., Yelowitz L. *Fuzzy Chains*. IEEE Trans. SMC-4, 5, 472–475, 1974.

[Kimball y Merz, 2000] Kimball y Merz. *The data webhouse toolkit*, John Wiley, 2000.

[Kiss, 1990] Kiss A. *On Fuzzy Relational Databases*. In Mathematical Sciences. Past and Present, vol. 3, pp. 1183–1193, 1990.

[Kiss, 1991] Kiss A. *γ-Decomposition of Fuzzy Relational Databases*. In Annales Univ. Budapest, Sect. Comp., vol. 12, pp. 133–142, 1991.

[Kivinen y Mannila, 1995] Kivinen J., Mannila H. *Approximate Dependency Inference from Relations*. Theoretical Computer Science 149(1), pp. 129–149, 1995.

[Klir et al., 1995] Klir G.J., Yuan B. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Pearson Education POD; 1st edition (May 11, 1995)

[Klir et al., 1997] Klir G.J., St. Clair U., Yuan B. *Fuzzy Set Theory: Foundations and Applications*. Pearson Education POD; 1st edition (April 17, 1997)

[Korth y Silberschatz, 1993] Korth H.F., Silberschatz A. *Fundamentos de Bases de datos (2.ed)*. Mc Graw Hill, 1993.

[Kraft y Buell, 1993] Kraft D.H. y Buell D.A. *Fuzzy sets and generalized boolean retrieval systems*, in Readings in Fuzzy Sets for Intelligent Systems, Dubois D. & Prade H., Eds., pp. 648–659. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[Li y Clifton, 2000] Li W. and Clifton C. *SEMINT: A tool for identifying attribute correspondences in heterogeneous databases using neural networks. Data Knowledge Engineering*, 33(1):44–84, 2000.

[Liu y Kerre, 1998] Liu Y. y Kerre E. *An overview of fuzzy quantifiers part I (interpretations) and II (reasoning and applications)*. Fuzzy Sets and Systems 95, pp. 1–21, 135–146, 1998.

[López de Mantaras y Valverde, 1988] López de Mantaras R., Valverde L. *New Results in Fuzzy Clustering based on the Concept of Indistinguishability Relation*, IEEE Trans. Pattern Anal. Machine Intell. 10, 754-757, 1988.

[LUCDEME, 1981] LUCDEME. *Mapa de suelos de Cehegin. Mapa 1:100000 y memoria.* MAPA-ICONA-University of Murcia, 1981.

[LUCDEME, 1987a] LUCDEME. *Mapa de suelos de Tabernas. Mapa 1:100000 y memoria.* MAPA-ICONA-CSIC, 1987.

[LUCDEME, 1987b] LUCDEME. *Mapa de suelos de Vera. Mapa 1:100000 y memoria.* MAPA-ICONA-University of Granada, 1987.

[Mannila et al., 1994] Mannila H., Toivonen H., Inkeri Verkamo A. *Efficient Algorithms for Discovering Association Rules*. KDD-94: AAAI Workshop on Knowledge Discovery in Databases, Seattle, Washington, Julio 1994.

[Marín et al, 2003] Marín N., Sánchez D., Serrano J.M., Vila M.A. *The linguistic cardinal: A proposal to solve the "select countçase in a fuzzy query environment.* Sometido para su publicación en International Journal on Intelligence Systems.

[Martín-Bautista, 2000] Martín-Bautista M.J. *Modelos de Computación Flexible para la Recuperación de Información*, Tesis Doctoral. Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada, Septiembre 2000.

[Medina et al, 1995] Medina J.M., Vila M.A., Pons O., Cubero J.C. *Towards the implementation of a generalized fuzzy relational database model.* Fuzzy Sets and Systems v. 75 pp. 273–289, 1995.

[Medina et al, 1994] Medina J.M., Pons O., Vila M.A. *GEFRED: A Generalized Model for Fuzzy Relational Databases.* Information Sciences v. 77(6) pp. 87-109, 1994.

[Medina, 1994] Medina J.M. *Bases de datos relacionales difusas. Modelo teórico y aspectos de su implementación.* Tesis Doctoral. Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada, 1994.

[Meo et al, 1996] Meo R., Psaila G., Ceri S. *A new SQL-like operator for mining association rules.* In Proc. VLDB'96, pages 122-133, September 1996. Morgan Kaufmann.

[Monge y Elkan, 1996] Monge A. and Elkan C. *The field matching problem: Algorithms and applications.* In *Proc. KDD'96*, pages 267–270, 1996.

[Munsell, 1954] Munsell. *Soil Color Charts.* Munsell Color Company Inc. Baltimore, Maryland, 1954.

[Oyonarte, 1990] Oyonarte C. *Estudio Edáfico de la Sierra de Gádor (Almería). Evaluación para usos forestales.* Tesis doctoral. Universidad de Granada, 1990.

[Park et al., 1995] Park J.S., Chen M.S., Yu P.S. *An Effective Hash-Based Algorithm for Mining Association Rules*. Procs. 1995 ACM-SIGMOD Int. Conf. Management of Data, 1995.

[Pawlak, 1984] Pawlak Z. *Rough classification*, Int. Journal of Man-Machine Studies, 20:469–483, 1984.

[Pedrycz, 1998a] Pedrycz W. *Fuzzy Set Technology in Knowledge Discovery*. Fuzzy Sets and Systems 98, pp.279–290, 1998.

[Pedrycz, 1998b] Pedrycz W. *An introduction to fuzzy sets : analysis and design*. Cambridge, Mass.: MIT Press, 1998.

[Pérez-Pujalte y Prieto, 1980] Pérez-Pujalte A. y Prieto P. *Mapa de suelos 1:200000 de la provincia de granada y memoria explicativa*. Technical report, CSIC, 1980.

[Prade y Testemale, 1984] Prade H., Testemale C. *Generalizing Database Relational Algebra for the Treatment of Incomplete/Uncertain Information and Vague Queries*. Information Sciences, 34. 115–143, 1984.

[Prade y Testemale, 1987] Prade H., Testemale C. *Fuzzy Relational Databases: Representational issues and Reduction Using Similarity Measures*. J. Am. Soc. Information Sciences, 38(2), pp. 118–126, 1987.

[Pfahringer y Kramer, 1995] Pfahringer, B. and Kramer, S. *Compression-based evaluation of partial determinations*. In *Proc. First Int'l Conf. Knowledge Discovery and Data Mining (KDD'95)*, pages 234–239, 1995.

[Piatetsky-Shapiro et al, 1993] Piatetsky-Shapiro G., Matheus C.J. *Probabilistic Data Dependencies*. Procs. of the 1993 Workshop in Machine Learning, Aberdeen, Scotland, 1993.

[Potoczny, 1984] Potoczny H.B. *On Similarity Relations in Fuzzy Relational Databases*. Fuzzy Sets and Systems, 12:231-235, 1984.

[Rasmussen, 1997] Rasmussen, D. *Summary SQL, A General Purpose Fuzzy Query Language*. Technical Report, Roskilde University, 1997.

[Rundensteiner et al, 1989] Rundensteiner E.A., Hawkes L.W., and Bandler W. *On Nearness Measures in Fuzzy Relational Data Models*. International Journal of Approximate Reasoning, 3. 267–298, 1989.

[Sánchez, 1999] Sánchez D. *Adquisición de relaciones entre atributos en bases de datos relacionales*. Tesis Doctoral. Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada, 1999.

[Sánchez-Marañón, 1992] Sánchez-Marañón M. *Los suelos del Macizo de Sierra Nevada. Evaluación y capacidad de uso*. Tesis doctoral. Universidad de Granada, 1992.

[Sánchez-Marañón et al., 1997] Sánchez-Marañón M., Delgado G., Melgosa M., Hita E., Delgado R. *CIELAB color parameters and their relationship to soil characteristic in Mediterranean Red Soils*. Soil Sci. 11, 833-842, 1997.

[Schwertmann, 1993] Schwertmann U. *Relations between iron oxides, soil color and soil formation*. Soil Sci. Soc. Am. Spec. Publ. Nº. 31, 51-69, 1993.

[Schulze et al., 1993] Schulze D.G., Nagel J.L., Van Scoyoc G.E., Henderson T.L., Baumgardner M.F., and Scott D.E. *Significance of organic matter in determining soil colors*. Soil Sci. Soc. Am. Spec. Publ. Nº. 31, 71-90, 1993.

[Serrano et al., 2001] Serrano J.M., Vila M.A., Aranda V., Delgado G. *Using Fuzzy Relational Databases to represent agricultural and environmental information. An example within the scope of olive cultivation in Granada*. Mathware & Soft Computing VIII, n.3, pp. 275-289, 2001.

[Shenoi y Melton, 1989] Shenoi S., Melton A. *Proximity relations in the fuzzy relational database model*. Fuzzy Sets and Systems, 31, 285–296, 1989.

[Shenoi et al., 1992] Shenoi S., Melton A., Fan L.T. *Functional Dependencies and Normal Forms in the Fuzzy Relational Database Model*. Information Sciences, 60:1–28, 1992.

[Shortliffe y Buchanan, 1975] Shortliffe, E. and Buchanan, B. *A model of inexact reasoning in medicine*. Mathematical Biosciences, 23:351-379, 1975.

[Silverstein et al., 1998] Silverstein C., Brin S., y Motwani R. *Beyond market baskets: Generalizing association rules to dependence rules*. Data Mining and Knowledge Discovery, 2:39–68, 1998.

[Soil Survey Staff, 1975] Soil Survey Staff. *Soil Taxonomy*. U.S. Dept. Agri. Handbook No. 436, 754 pp, 1975.

[Srikant y Agrawal, 1995] Srikant R., Agrawal R. *Mining Generalized Association Rules*. Procs. of the 21st Int'l Conference on Very Large Databases. Zurich, Suiza, Septiembre 1995.

[Srikant y Agrawal, 1996] Srikant R., Agrawal R. *Mining Quantitative Association Rules in Large Relational Tables*. Procs. of the ACM SIGMOD Conference on Management of Data, 1996.

[Sys et al., 1991] Sys I.C., Van Ranst E. & Debaveye I.J. *Land Evaluation*. Agricultural Publications, n. 7. University of Ghent, 1991.

[Tahani, 1977] Tahani V. *A Conceptual Framework for Fuzzy Query Processing-A Step toward Very Intelligent Database Systems*. Information Process. Management, 13, pp. 289–303, 1977.

[Tamura et al., 1971] Tamura S., Higuchi S., Tanaka K. *Pattern classification based on fuzzy relations*. IEEE Trans. SMC-1, 61–66, 1971.

[Torrent et al., 1980] Torrent J., Schwertmann U., and Schulze D.G. *Iron oxide mineralogy of two river terraces sequences in Spain*. Geoderma 23, 191-208, 1980.

[Ullman, 1989] Ullman J.D. *Principles of Database and Knowledge-Base Systems Computer Science Vol I y II*. Bpress New York USA, 1989.

[Umano, 1982] Umano M. *Freedom-O: A Fuzzy Database System*. Fuzzy Information and Decision Processes. Gupta-Sanchez edit. North-Holand Pub. Comp., 1982.

[Umano y Fukami, 1994] Umano M., amd Fukami S. *Fuzzy Relational Algebra for Possibility-Distribution-Fuzzy Relational Model of Fuzzy Data.* Journal of Intelligence Information Systems, 3, pp. 7–27, 1994.

[Vila et al, 1999] Vila M.A., Delgado M., Gómez-Skarmeta A.F. *Pattern Recognition with Evidential Knowledge.* International Journal of Intelligent Systems, v. 14(2), 145–164, 1999.

[Wang y Tsai, 1999] Wang S.-L., Tsai J.-S. *Discovery of Approximate Dependencies from Proximity-based Fuzzy Databases.* Proceedings of the 3rd International Conference on Knowledge-based Intelligent Information Engineering Systems, August 1999, Adelaide, Australia, 234-237.

[Wang et al., 1999] Wang S.-L., Tsai J.-S., Chien B.-C. *Mining Approximate Dependencies Using Partitions on Similarity-relation-based Fuzzy Databases.* Proceedings of the IEEE SMC'99, October 1999, Tokyo, Japan, V-871–V-875.

[Wang et al., 2000] Wang S.-L., Tsai J.-S., Hong T.-P. *Mining Functional Dependencies from Fuzzy Relational Databases.* Proceedings of the ACM SAC 2000, Fuzzy Application and Soft Computing Track, March 2000, Italy, 490-493.

[Wright, 1998] Wright, P. *Knowledge Discovery In Databases: Tools and Techniques.* ACM Crossroads - Issue 5.2 - Networks and Distributed Systems, 1998.

[Yager et al, 1987] Yager R.R., Ovchinnikov S., et al. *Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh.* Wiley Intersc., 1987.

[Yang y Singhal, 1999] Yang Y., Singhal M. *Fuzzy Functional Dependencies and Fuzzy Association Rules.* In *Data Warehousing and Knowledge Discovery, First International Conference, DaWaK '99, Florence, Italy, August 30 - September 1, 1999, Proceedings.* Mohania M. y Tjoa A.M. (Eds.), LNCS 1676, pp. 229–240, Springer-Verlag Berlin Heidelberg, 1999.

[Zadeh, 1965] Zadeh L.A. *Fuzzy Sets.* Information Control, 8:338-353, 1965.

[Zadeh, 1971] Zadeh L.A. *Similarity Relations and Fuzzy Orderings.* Information Sciences, 3:177-200, 1971.

[Zadeh, 1975a] Zadeh L.A. *The Concept of a Linguistic Variable and its Applications to Approximate Reasoning, I.* Information Sciences, 8:199-249, 1975.

[Zadeh, 1975b] Zadeh L.A. *The Concept of a Linguistic Variable and its Applications to Approximate Reasoning, II.* Information Sciences, 8:301-357, 1975.

[Zadeh, 1976] Zadeh L.A. *The Concept of a Linguistic Variable and its Applications to Approximate Reasoning, III.* Information Sciences, 9:43-80, 1976.

[Zadeh, 1978] Zadeh L.A. *Fuzzy Sets as a Basis for a Theory of Possibility.* Fuzzy Sets and Systems, 1, pp. 3–28, 1978.

[Zadeh, 1979a] Zadeh L.A. *Fuzzy sets and information granularity*, In: Gupta M. M. et al. (eds) Advances in Fuzzy Set Theory and Applications. North Holland, 3–18, 1979.

[Zadeh, 1979b] Zadeh L.A. *A computational approach to fuzzy quantifiers in natural languages.* Comp. and Math. with Appl. 9: 149–184, 1979.

[Zadeh, 1983] Zadeh L.A. *A computational approach to fuzzy quantifiers in natural languages.* Computing and Mathematics with Applications, vol. 9, no. 1, pp. 149-184, 1983.

[Zadeh, 1996] Zadeh L.A. *Fuzzy sets, fuzzy logic, and fuzzy systems : selected papers.* Singapore [etc.]: World Scientific, 1996.

[Zhang et al., 1997] Zhang Z., Lu Y., Zhang B. *An Effective Partitioning-Combining Algorithm for Discovering Quantitative Association Rules.* National Natural Science Foundation of China, 1997.

[Zemankova y Kandel, 1984] Zemankova-Leech M., Kandel A., *Fuzzy Relational Databases – A Key to Expert Systems.* Köln, Germany, TÜV Rheinland, 1984.

[Ziarko, 1991] Ziarko, W. *The Discovery, Analysis and Representation of Data Dependencies in Databases*. In: G. Piatetsky-Shapiro and W. Frawley (eds.): *Knowledge Discovery in Databases*. AAAI/MIT Press, pp. 195–209, 1991.

[Zimmermann, 1991] Zimmermann H.J. *Fuzzy Set Theory and its Applications,* 2^{nd} *Edition*. Ed. Kluwer Academic Publishers, 1991.