UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE CIENCIAS MATEMÁTICAS

Departamento de Estadística e Investigación Operativa

ANÁLISIS DE SEGMENTACIÓN EN EL ANÁLISIS DE DATOS SIMBÓLICOS

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

María del Carmen Bravo Llatas

Bajo la dirección del doctor

José Miguel García – Santesmases Martín - Tesorero

Madrid, 2001

ISBN: 84-669-1791-8

UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE CIENCIAS MATEMÁTICAS
Departamento de Estadística
e Investigación Operativa I

ANÁLISIS DE SEGMENTACIÓN EN EL ANÁLISIS DE DATOS SIMBÓLICOS

TESIS DOCTORAL

María del Carmen Bravo Llatas

Madrid, 2001

UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE CIENCIAS MATEMÁTICAS
Departamento de Estadística
e Investigación Operativa I

ANÁLISIS DE SEGMENTACIÓN EN EL ANÁLISIS DE DATOS SIMBÓLICOS

María del Carmen Bravo Llatas

Memoria para optar al grado de Doctora en Ciencias Matemáticas, realizada bajo la dirección del Profesor Dr. D. José Miguel García-Santesmases Martín-Tesorero

JOSÉ MIGUEL GARCÍA-SANTESMASES MARTÍN-TESORERO,
PROFESOR TITULAR DEL DEPARTAMENTO DE ESTADÍSTICA E INVESTIGACIÓN OPERATIVA I DE LA UNIVERSIDAD COMPLUTENSE DE MADRID

CERTIFICA:

Que la presente Memoria titulada:

ANÁLISIS DE SEGMENTACIÓN EN EL ANÁLISIS DE DATOS SIMBÓLICOS

ha sido realizada bajo mi dirección por Doña María del Carmen Bravo Llatas, Licenciada en Ciencias Matemáticas, y constituye su Tesis para optar al grado de Doctora en Ciencias Matemáticas.

Y para que conste, en cumplimiento de la legislación vigente y a los efectos oportunos, firmo la presente en Madrid a 7 de Septiembre de dos mil uno.

A mis seres queridos

Índice General

Prólogo 1

I *Fundamentos* 11

1 Conceptos del Análisis de Datos Simbólicos 13

1.1 Introducción . . . 13

1.2 Análisis de Datos . . . 16

1.2.1 Variables monoevaluadas . . . 16

1.2.2 Matriz de datos . . . 17

1.3 Análisis de Datos Simbólicos . . . 17

1.3.1 Matriz de datos simbólicos . . . 19

1.3.2 Variables multievaluadas . . . 20

1.3.3 Variables modales probabilistas . . . 24

1.3.4 Variables modales posibilistas . . . 30

1.3.5 Conjunto de descripciones simbólicas . . . 38

1.4 Objetos simbólicos . . . 40

1.4.1 Relaciones de dominio . . . 41

1.4.2 Eventos . . . 46

1.4.3 Aserciones . . . 54

1.4.4 Otros tipos de datos y objetos simbólicos . . . 62

1.4.5 Generalización . . . 65

1.5 Operaciones sobre conjuntos de aserciones . . . 67

1.5.1 Unión, intersección y complementariedad 68
1.5.2 Conjunción 72
1.6 Conclusión 76

2 Segmentación **79**
2.1 Árboles de Segmentación 79
2.1.1 Introducción 80
2.1.2 Datos de partida 81
2.1.3 Objetivo y Método 82
2.1.4 Esquema del algoritmo 85
2.1.5 Nodos del árbol 86
2.1.6 Criterios 87
2.1.7 Antecedentes 88
2.2 Árboles de Segmentación con incertidumbre 101
2.2.1 Introducción 101
2.2.2 Método con incertidumbre 102
2.2.3 Antecedentes 108
2.3 Conclusión 121

II *Segmentación y Análisis de Datos Simbólicos* **123**

3 Segmentación para Datos Estratificados **125**
3.1 Introducción 125
3.1.1 Datos de partida 128
3.1.2 Estratos y Objetivos 130
3.2 Método y representación 133
3.2.1 Árbol 135
3.2.2 Nodos del árbol 139
3.2.3 Estratos 142

3.2.4 Criterios 143
3.3 Algoritmo 149
3.3.1 Inicialización 150
3.3.2 Admisibilidad 151
3.3.3 Maximización 152
3.3.4 Nodos decisionales 153
3.3.5 Parada 155
3.4 Conclusión 158

4 Método Datos Monoevaluados, Modales Probabilistas y Extensiones 159

4.1 Criterios para datos monoevaluados 160
4.1.1 Elementos posibles de partición 160
4.1.2 Función de admisibilidad 162
4.1.3 Medidas de contenido de información 163
4.1.4 Descripción de la estimación de la variable clase 167
4.1.5 Condición de nodo decisional 169
4.1.6 Condición de parada 171
4.2 Criterios para datos modales probabilistas 171
4.2.1 Elementos posibles de partición 171
4.2.2 Función de admisibilidad 173
4.2.3 Medidas de contenido de información 173
4.2.4 Descripción de la estimación de la variable clase 178
4.2.5 Condición de nodo decisional 179
4.2.6 Condición de parada 180
4.3 Caracterización del árbol 180
4.3.1 Datos monoevaluados y probabilistas 180
4.3.2 Datos monoevaluados 184
4.3.3 Datos modales probabilistas 190

4.4 Descripción simbólica de los estratos 193
4.4.1 Datos monoevaluados 194
4.4.2 Datos modales probabilistas 195
4.4.3 Interpretación de los estratos 195
4.4.4 Ventajas del método 197
4.5 Predicción 203
4.5.1 Nivel de relación con el árbol 203
4.5.2 Reglas de predicción para datos monoevaluados 206
4.5.3 Reglas de predicción para datos modales probabilistas . . . 207
4.6 Calidad del árbol 210
4.6.1 Antecedentes 210
4.6.2 Aproximación para el árbol de datos estratificados 213
4.7 Extensiones del método 222
4.7.1 Peso en los individuos 222
4.7.2 Probabilidades 'a priori' de las clases 226
4.7.3 Peso en los individuos y probabilidades 'a priori' de las clases 228
4.8 Extension del método a otros datos simbólicos 229
4.9 Aplicaciones 241
4.9.1 Datos SES, 1995 241
4.9.2 Normas de interpretación 254
4.9.3 Apreciación de los municipios 255
4.9.4 Datos relacionados con la actividad laboral 262
4.9.5 Datos probabilistas SES, 1995 267
4.9.6 Conclusiones 271
4.10 Conclusión 275

5 Implementación del Método **277**

5.1 Especificaciones 277
5.2 Entrada 280

5.2.1 SDT v2.22b 280
5.2.2 Ficheros de datos SODAS 281
5.2.3 SDTEDITOR v2.22 282
5.3 Requisitos y criterios adoptados en la implementación 283
5.4 Salida 286
5.4.1 Fichero de resultados 286
5.4.2 Fichero de diagnósticos 288
5.4.3 Fichero gráfico y visualización gráfica del árbol 289
5.5 Adaptaciones posibles 292
5.5.1 Implementaciones futuras 292

Conclusiones **295**

Apéndices **301**

Apéndice A. Diseño del programa SDT **305**

Apéndice B. Diseño del programa SDTEDITOR **335**

Bibliografía **348**

Índice de Tablas

4.1 Datos de entrada de 4.4.4 198

4.2 Extensión del método para datos posibilistas o difusos 235

4.3 Extensiones del método. Estimaciones de la variable clase Z, según tipos de predictores y clases. 236

4.4 Tabla de contribuciones relativas y absolutas. Datos SES. 251

4.5 Datos SES probabilistas. Tabla de contribuciones relativas y absolutas 270

Índice de Figuras

4.1 Árbol para datos estratificados 199
4.2 Árbol de decisión tradicional 200
4.3 Árbol de Segmentación para los datos SES, 5 niveles 245
4.4 Árbol de Segmentación para los datos SES, 3 niveles 248
4.5 Árbol de Segmentación para datos municipales, 5 niveles 257
4.6 Árbol de Segmentación para datos municipales, 3 niveles 260
4.7 Árbol de Segmentación para datos relacionados con la actividad, 4 niveles 265
4.8 Árbol de Segmentación para los datos probabilistas SES, 3 niveles 269

5.1 Interfaces externas de SDT y SDTEDITOR en SODAS 279
5.2 SDTEDITOR. Árbol con nodos explorables. 283
5.3 SDTEDITOR. Información general del árbol. 290
5.4 SDTEDITOR. Enmarcación de un estrato e infomación de nodos 291
5.5 SDTEDITOR. Árbol de cinco niveles en una ventana 293

Agradecimientos

Quisiera mostrar mi agradecimiento a todas aquellas personas que han colaborado o me han ayudado durante estos años de elaboración de esta Memoría, sin cuyo apoyo esta larga tarea es posible que no hubiera visto su fin.

Especialmente, mi gran agradecimiento al Profesor José Miguel García-Santesmases por aceptar el compromiso de dirigirme en este trabajo y brindarme la oportunidad de trabajar con él. Por sus ideas, nuestras reuniones y su paciencia.

Mi sincero agradecimiento al Profesor Edwin Diday, de la Universidad de Paris IX-Dauphine, por introducirme en el mundo simbólico allá por 1992 y por las agradables discusiones acerca de sus teorías, que comenzaron por el *concepto de champiñon* y los *champiñones que estamos viendo*. También le estoy muy agradecida por invitarme a participar en el proyecto ESPRIT-IV 20821 SODAS.

Al Profesor Hans-Hernann Bock, del Institut für Statistik, de Aachen por sus revisiones y sugerencias de una parte de este trabajo. A Emmanuel Périnel, de INRIA (Francia) por las discusiones *segmento-simbólicas* y en especial por las referencias bibliográficas que me proporcionó. Al Profesor Antonio Ciampi, de la Universidad McGill de Montreal, y a Yves Lechevallier, de INRIA por las conversaciones que mantuvimos acerca de la Segmentación y los objetos simbólicos.

A los compañeros del proyecto SODAS, por su atención a mi trabajo y por las discusiones *teórico-simbólicas* que mantuvimos entre todos, de especial interés en el proyecto las ideas de Catherine Bouillet y la Profesora Mireille Gettler-Summa; por las sugerencias para la realización del *software* y su documentación,

en especial a Marc Csernel, George Hébrail, la Profesora Monique Noirhomme y Michel Muenier; y, por su colaboración en la comprensión de sus bases de datos y por la utilización del *software* del método presentado en esta Memoria, en particular a Isabelle Piroth, Patricia Calvo, Carlos Marcelo y Anjeles Iztueta.

A Alberto Fernandez al que se debe la codificación del método. Por las duras y largas jornadas de trabajo en colaboración y en solitario y por su profesionalidad.

A los miembros del proyecto Clorec de INRIA y del laboratorio Lise-Ceremade de la Universidad de Paris-IX Dauphine, por su gentileza en las breves estancias que allí pasé. A la Profesora Paula Brito por sus invitaciones y amistad. A Emmanuel Périnel, Marie Chavent, Fréderic Vautrain, Myriam Touati. A Mireille Gettler-Summa por sus agradables invitaciones campestres.

Y, finalmente, aunque no por ello menos importante, quisiera agradecer la colaboración que el Centro de Proceso de Datos me prestó al permitir mi participación en el proyecto ESPRIT-IV 20821 facilitándome la asistencia a todas las reuniones necesarias con los componentes del Consorcio. Sin este apoyo institucional, mi participación en el proyecto no hubiera sido posible. A mis compañeros de Apoyo a Investigación, que prescindieron de mí durante estas reuniones.

A Luis, Miguel Angel, José Luis y Abelardo, por su ayuda con el $latex_{2\varepsilon}$.

A todos mis amigos y a mi familia, que durante todos estos años me oyeron hablar de este trabajo con los altibajos correspondientes. A Neluca, Isabel, Pedro, Marta, Ramón y Josefina por su apoyo incondicional. A Luis, el primer Doctor de la familia. A Belén y Maena con las que compartí muchas tardes realizando nuestros Trabajos de Investigación. A Belén, Nacho y Javier, por su apoyo constante. A David, Consuelo y Victoria por su ayuda en momentos clave. A los de siempre, Lurdes, Manolo, Carmen, Marc y Gonzalo. A Beatriz, Gonzalo, Margarita, Miguel Angel, Ignacio, Severino, Belén y José Luis a los que conocí mientras realizaba este trabajo y que finalmente ven concluido.

Y a todos los demás que amablemente me han apoyado en algún momento.

Prólogo

Prólogo

Esta Memoria se encuadra dentro del marco del *Análisis de Datos Simbólicos* y de las técnicas de *Segmentación*.

El Análisis de Datos Simbólicos permite el análisis de conocimientos. La *extracción de conocimientos* del Análisis de Datos clásico es la obtención de resultados por sí mismos explicativos que representan conceptos. En esta Memoria se presenta la formalización de los conceptos mediante los objetos simbólicos que constituyen la entrada y la salida de las técnicas de Análisis de Datos Simbólicos. Además, estos objetos simbólicos permiten consultas a una base de datos permitiendo la propagación de los conceptos.

El Análisis de Datos Simbólicos permite la extensión de la Estadística a la Estadística de las intenciones o conceptos y más concretamente la extensión de problemas, métodos y algoritmos de Análisis de Datos a datos simbólicos. Según Diday el Análisis de Datos Simbólicos crea un puente entre la Estadística y el Aprendizaje Automático.

Ya Aristóteles en su Organón (Aristotle, IV a.C.) distingue entre un individuo y la descripción del mismo. Si bien los conceptos de *intención* y *extensión* se deben a Arnauld y Nicole (Arnauld y Nicole, 1662) es Diday (Diday (1987,1988)) quien formaliza estos términos del Análisis de Datos Simbólicos. La intención de un concepto constituye su descripción, mientras que la extensión es el conjunto de individuos cuya descripción es acorde a la del concepto. La intención, que se representa por un objeto simbólico, se describe por los datos simbólicos y por un

mecanismo de reconocimiento de los individuos de la extensión.

Diday introduce los objetos simbólicos y presenta una formalización que permite tratar conocimientos más ricos que los datos habituales, y establece una relación con el modelo clásico de Análisis de Datos (Diday, 1991). Un objeto simbólico representa una intención, un concepto y se define, en términos generales, como una conjunción de valores, o conjuntos de valores que pueden ser ponderados, de variables. Constituye una descripción en intención de una clase de individuos que constituyen la extensión.

Según Diday (Diday, 2000), el Análisis de Datos Simbólicos nace influido por tres campos:

- El Análisis Exploratorio de Datos,
- La Inteligencia Artificial, donde gran esfuerzo se realiza por el desarrollo de lenguajes de representación de conocimiento,
- Y, la Taxonomía Numérica usada en las Ciencias Biológicas.

La formalización de los objetos simbólicos ya ha evolucionado desde sus inicios (Diday (1987, 1988, 1991, 1993a, 1993b)). Esta Memoria presenta la adoptada en el libro editado por Bock y Diday (Bock y Diday, 2000a), que según sus editores es la primera monografía sistemática y completa del Análisis de Datos Simbólicos.

Las ventajas del Análisis de Datos Simbólicos son:

1. Forma de representación del conocimiento en lenguaje fácilmente comprensible al usuario.
2. Análisis Estadístico de Datos que representan intenciones o conceptos. Así como el Análisis de Datos extrae conocimientos, el Análisis de Datos Simbólicos extrae nuevos conocimientos a partir de conocimientos previos.
3. Extensión de las técnicas del Análisis de Datos a los datos simbólicos.

4. Se representan y analizan datos de mayor complejidad que los datos tradicionales, ya que contienen variación interna, como los intervalos, conjuntos de valores, distribuciones de probabilidad, etc... Y, además, son estructurados como las taxonomías y las dependencias jerárquicas y lógicas entre variables. Es decir, los datos simbólicos contienen además metadatos.

5. Los resultados de los análisis se interpretan fácilmente en el lenguage del usuario.

6. Los objetos de entrada y salida de las técnicas de Análisis de Datos Simbólicos son representados por un único formalismo, comprensible al usuario.

7. Los objetos simbólicos pueden venir dados por el conocimiento de un experto.

8. Las intenciones se pueden extraer de bases de datos reagrupando o agregando datos individuales (Stéphan et al., 2000).

9. Cada intención viene acompañada de una extensión que es el conjunto de individuos de una base de datos que *se adecuan* a la intención. Una misma intención puede aplicarse (es una consulta) a diversas bases de datos o a una misma base de datos en distintos momentos de tiempo, facilitando la propagación de conceptos.

10. Desde un punto de vista formalista, se presenta una representación unificada de aproximaciones a la *incertidumbre:* datos expresados como distribuciones de probabilidad, de posibilidad, conjuntos difusos, creencias (Diday, 1995a).

11. Se preserva la confidencialidad de los datos individuales, al analizar agrupaciones de los mismos.

12. Se da una solución a la selección de información y almacenamiento de datos de un DataWarehouse, aportándose a las Oficinas de Estadística un medio de extracción de conocimientos de sus grandes bases de datos.

13. La relación con el Análisis de Datos tradicional queda preservada, toda vez que el Análisis de Datos Simbólicos aplicado a los datos habituales es el Análisis de Datos tradicional. Además, a partir de datos simbólicos se puede realizar Analisis de Datos tradicional mediante la obtención de matrices de similaridades entre objetos simbólicos (Gowda y Diday, 1992, De Carvalho, 1994, De Carvalho y Diday, 1998, Bock, 2000b, Bacelar-Nicolau, 2000, Esposito et al., 2000b).

Los métodos de Segmentación y Discriminación del Análisis de Datos, se conocen en Aprendizaje Automático como métodos de aprendizaje de conceptos, enmarcados en la *Inferencia Inductiva*. Son métodos que permiten determinar una representación simbólica, generalmente discriminante, de una clasificación dada. Las técnicas de Segmentación generan reglas de clasificación a partir de observaciones.

Michalski (Michalski, 1983) define la *Inferencia Inductiva* como el proceso de ir de un conocimiento derivado de la observación de algunos objetos a un conocimiento más estructurado en forma de *complejos* que son soportados con diversa intensidad por los datos observados. Una de las técnicas de inferencia inductiva es el *Aprendizaje Automático* a partir de ejemplos, en el cual el conocimiento observado se compone de unos objetos de clases conocidas y el conocimiento estructurado derivado del proceso se expresa por un conjunto de reglas que permiten la clasificación de éstos y de nuevos ejemplos de clases desconocidas. Quinlan define el Aprendizaje como la adquisición de conocimiento estructurado en forma de conceptos, redes de discriminación o reglas de producción (Quinlan, 1986a, Wu, 1993).

La aportación que incorpora esta Memoria es el Análisis de Segmentación para datos simbólicos estratificados. Extiende los métodos de Segmentación a la presencia de estratos en la población de una parte y a la presencia de estratos e incertidumbre en los predictores de otra y se presenta una formalización genera-

lizada del método en términos de objetos simbólicos. Frecuentemente, en grandes volumenes de datos, como es el caso de las Oficinas de Estadística, la población no sólo se encuentra dividida en clases conocidas sino que se encuentra estratificada en subpoblaciones y se hace necesario explicar o discriminar las clases de los individuos y predecir la clase de nuevos individuos de estratos conocidos; explicar y agrupar los estratos por reglas de predicción comunes; y, finalmente, describir agregadamente un grupo de individuos, el estrato, mediante objetos simbólicos que representan las propiedades o descripciones de un elemento genérico del estrato que describen. El método proporciona una descripción por objetos simbólicos de los estratos, como información agregada de los mismos, representando una generalización o intención de los mismos, expresada en términos de reglas de predicción de las clases.

Esta Memoria se ocupa de predictores cualitativos o categóricos. Los datos simbólicos considerados son los modales probabilistas representados por distribuciones de probabilidad, siendo caso particular de éstos, los datos categóricos. Se incorpora también el tratamiento de información estructurada de los datos, en particular el tratamiento de reglas de no aplicabilidad[1].

Se presenta una formalización generalizada del método propuesto en términos de objetos simbólicos de tal forma que el algoritmo general puede ser extendido a otros tipos de datos simbólicos y a otras semánticas de incertidumbre, aportándose propuestas concretas.

La parte I de esta Memoria se centra en los fundamentos: los conceptos básicos del Análisis de Datos Simbólicos en el capítulo 1 y la Segmentación, en el capítulo 2.

El capítulo 1 presenta los conceptos básicos del Análisis de Datos Simbólicos e introduce los datos y los objetos simbólicos que representan una formalización única para los datos complejos y la incertidumbre. Los datos simbólicos repre-

[1] Una regla de no aplicabilidad establece una relación entre variables que identifica las variables que son no aplicables debido a los valores de otra variable.

sentan los datos de entrada del método propuesto en esta Memoria. En concreto, los datos simbólicos modales probabilistas. Los objetos simbólicos son necesarios para la representación del árbol y la descripción de los estratos en los capítulos 3 y 4. La aportación novedosa de este capítulo es la formalización de los datos, variables y objetos simbólicos posibilistas y difusos, en el nuevo marco de representación introducido en Bock y Diday, 2000a, si bien los objetos simbólicos posibilistas habían sido introducidos en Diday (1991, 1995a). Éstos son introducidos por varios motivos: para destacar el marco común de representación del conocimiento mediante objetos simbólicos; por ser la entrada principal de las referencias de Segmentación con incertidumbre que se presentan en el capítulo 2; y debido a que la formalización del método propuesto se extiende a este tipo de datos, como se presenta en el capítulo 4.

El capítulo 2 introduce la Segmentación y el tratamiento de la incertidumbre en la Segmentación y presenta una breve revisión bibliográfica de las mismas. Este capítulo ya apunta el modo de representación del árbol por objetos simbólicos y la presentación de los criterios en presencia de incertidumbre en terminología de datos y objetos simbólicos.

La parte II de esta Memoria, se centra en la Segmentación y el Análisis de Datos Simbólicos.

El capítulo 3 presenta un nuevo método de Segmentación para datos estratificados. Se representa el árbol mediante un conjunto de objetos simbólicos, que permite la entrada y el Análisis de Datos Simbólicos. Como resultado de la aplicación del método, se clasifican los estratos por reglas de predicción comunes y se obtiene una descripción de los mismos por conjuntos de objetos simbólicos. Se crea además el marco que permite una formalización general del método que lo hace extensible a otros datos simbólicos y otros tipos de expresión de la incertidumbre.

El capítulo 4 particulariza los criterios del método de Segmentación para datos estratificados establecidos en el capítulo 3, cuando los datos de entrada son

monoevaluados y modales probabilistas. Caracteriza el árbol obtenido, aporta criterios para la interpretación de los nodos del árbol y los estratos, destacando sus ventajas respecto al método de Segmentación tradicional. Aporta normas de predicción para nuevos individuos, y medidas de calidad del árbol, proponiéndo algunas mejoras al método presentado en el capítulo 3, incluidos criterios de parada adicionales. Propone extensiones al método que incluyen el peso en los individuos y las probabilidades *a priori* de las clases. Propone desde un punto de vista general, extensiones del método a otros datos simbólicos y expresiones de incertidumbre, proporcionando tipos de objetos simbólicos y criterios concretos en estas extensiones. Y, finalmente se presentan algunas aplicaciones que incorporan ayudas a la interpretación de los resultados.

El capítulo 5 presenta la implementación del método propuesto en dos programas de *software*, uno de construcción de árboles y otro de visualización de los mismos. Presenta las especificaciones generales de los programas, así como las características de la entrada y salida y los parámetros de entrada de los programas.

Finalmente se exponen las conclusiones de esta Memoria.

Acompañan a esta Memoria dos apéndices con los diseños de los programas de *software* desarrollados.

De este trabajo se han derivado algunas publicaciones (Bravo y García-Santesmases, (2000a, 2000b), Bravo, 2000a), y comunicaciones en congresos internacionales: ASMDA97 (Bravo y García-Santesmases, 1997), NTTS98 (Bravo y García-Santesmases, 1998) e IFCS00 (Bravo, 2000b).

El trabajo desarrollado en este Memoria se ha visto motivado por mi interés personal en el Análisis de Datos (Bravo (1991, 1994), Bravo y Marina, 1996) y la experiencia profesional derivada de la colaboración en proyectos de investigación y tesis doctorales en el Análisis de los Datos.

Mi participación en el proyecto europeo ESPRIT IV 20821 SODAS[2] del IV programa marco de la UE (coordinador científico: Profesor Edwin Diday) me ha brindado la oportunidad de implementar en un *software* los resultados de esta investigación e incorporarlos al paquete SODAS 1.04 de obtención, visualización y Análisis de Datos Simbólicos (Morineau, 2000). Gracias a esta participación, se han podido contrastar los resultados de esta investigación con los participantes del proyecto y sobre todo ha sido posible ofrecer estos resultados a los demás miembros de la comunidad científica, mediante el programa de *software* desarrollado. El proyecto ha sido cofinanciado por EUROSTAT demostrando el interés y la necesidad de la Estadística Oficial de la UE en consolidar datos, obtener y analizar conocimientos de las grandes bases de datos que se almacenan en la actualidad; y, viendo en el Análisis de Datos Simbólicos un medio prometedor para ello.

[2]Los participantes del proyecto ESPRIT IV 20821 SODAS son: THOMSON-CSF detexis (FR) coordinador administrativo, las universidades de Paris IX-Dauphine (FR), UCM(E), FUNDP (BE), FUNDP-MA (BE), DMS (IT), RWTH (GE), DIB(IT), UOA(GR) y LEAD(P); los centros de investigación INRIA (FR) y CRP-CU (LU), las empresas EDF (FR) y CISIA (FR) y los Institutos Oficiales de Estadística INE (P), EUSTAT (E) y CSO (UK).

Parte I

Fundamentos

Capítulo 1

Conceptos del Análisis de Datos Simbólicos

1.1 Introducción

En el Análisis de Datos, los datos analizados proceden de observaciones únicas de determinadas *variables* sobre individuos únicos. La gran cantidad de datos que se recogen en la actualidad en empresas y Oficinas de Estadística, el inmenso tamaño de las bases de datos, la necesidad de sistemas de información Estadística y el uso cada vez más extendido de Internet y multimedia hacen necesario el procesamiento y el análisis de estructuras de datos más complejas que los datos clásicos: los *datos simbólicos*. El origen de los datos simbólicos es diverso. Pueden provenir de la agregación de individuos considerando clases o grupos de los mismos y la descripción de las propiedades de estas clases por nuevos tipos de variables y datos, las *variables* y *datos simbólicos* (Stéphan et al., 2000, Gettler-Summa, 1999). Estas agregaciones de individuos pueden establecerse "a priori" o provenir del resultado de otros análisis, como el Análisis de Conglomerados, por ejemplo (Gettler-Summa et al., 1994, Goupil et al., 2000). Los datos simbólicos representan de este modo las propiedades o descripciones de un elemento genérico

de la clase que describen.

Desde otro punto de vista, los datos simbólicos pueden establecerse, así mismo, por el conocimiento del experto sin necesidad de datos individuales y, así mismo, venir dados con incertidumbre (véase Prade, 1985). También, los datos simbólicos permiten representar metadatos tales como: asociaciones entre las categorías de una variable, formando taxonomías; dependencias jerárquicas entre variables estableciendo aquellas que son no aplicables según los valores de otra variable; y dependencias lógicas entre variables.

Como se introduce en el prólogo, los *objetos simbólicos* representan *conceptos*, entendido un concepto como la *intención* y la *extensión* del mismo. La intención de un concepto representa las propiedades que lo definen y que lo hacen distinto de los demás conceptos. La extensión de un concepto se compone de los individuos que se definen por el concepto o que cumplen las propiedades que definen el concepto. Un objeto simbólico constituye una descripción en intención de una clase de individuos que constituyen la extensión. Los objetos simbólicos se describen por variables y datos simbólicos y proporcionan un mecanismo de vuelta a bases de datos o conjuntos de individuos en el sentido de conocer aquéllos que se *adecuan* o *relacionan* con las descripciones simbólicas representadas por los objetos (las *intenciones*), según determinadas *relaciones* que también forman parte de las intenciones. Estos individuos constituyen la *extensión* de los objetos simbólicos.

El Análisis de Datos Simbólicos es una extensión de las técnicas de Análisis de Datos aplicadas a matrices de datos simbólicos siendo el Análisis de Datos un caso particular del Análisis de Datos Simbólicos (Diday, 1991). El Análisis de Datos Simbólicos puede verse como un Análisis de Datos de las *intenciones* o descripciones de elementos genéricos de clase, de una clase de individuos. Así mismo, como se introducía en el prólogo, pueden obtenerse matrices de similaridad o de disimilaridad entre objetos simbólicos (aplicadas a las filas de una matriz de datos simbólicos) y aplicar a éstas matrices, técnicas de Análisis de Datos tradi-

cional (Gowda y Diday, 1992, De Carvalho, 1994, De Carvalho y Diday, 1998, Bock, 2000b, Bacelar-Nicolau, 2000, Esposito et al., 2000b).

Como se apuntaba en el prólogo, las definiciones y notación de variables simbólicas y objetos simbólicos han estado en constante evolución desde sus inicios (Diday (1987, 1988, 1991, 1993a, 1993b)). Las definiciones contenidas en este capítulo tratan de seguir fielmente los conceptos, notación y definiciones contenidas en la primera monografía de Análisis de Datos Simbólicos (Bock y Diday, 2000a). En este libro, se realiza por primera vez una distinción explícita entre variables y datos simbólicos de una parte, y objetos simbólicos de otra (Diday, 2000, Bock, 2000a, Bock y Diday, 2000b). Se exponen en este capítulo las particularizaciones necesarias para la comprensión de los capítulos posteriores de esta Memoria. Esta Memoria se centra en el caso de las variables cualitativas o categóricas y por tanto, se presentan aquí las variables simbólicas relacionadas.

En 1.2 se presenta la matriz del Análisis de Datos. En 1.3 se presentan la matriz de datos en el Análisis de Datos Simbólicos y los distintos tipos de variables y datos simbólicos: en 1.3.2, las variables simbólicas multievaluadas; en 1.3.3, las variables simbólicas modales probabilistas, que representan los datos de la parte II, y en 1.3.4, las variables simbólicas modales posibilistas y difusas. Esta última formalización es una contribución novedosa, si bien fueron introducidos los objetos simbólicos posibilistas en Diday (1991, 1995a). En 1.4 se introducen los objetos simbólicos, presentándose en 1.4.1 las relaciones de dominio necesarias en la definición de los mismos; y en 1.4.2 y 1.4.3 los tipos más habituales de objetos simbólicos que son los eventos y las aserciones. En 1.4.4, se introducen otros tipos de datos y objetos simbólicos y en 1.4.5, antecedentes a la generalización de datos y objetos simbólicos a partir de grupos de individuos. Por último, en 1.5 se introducen algunas operaciones entre aserciones, necesarias en el capítulo 4.

Las variables y datos multievaluados y modales probabilistas definen los datos de entrada y son necesarios para la representación del árbol mediante objetos sim-

bólicos en la parte II de esta Memoria. Se introducen las variables, datos y objetos simbólicos posibilistas y difusos por varios motivos: de una parte para destacar el marco común de representación del conocimiento con incertidumbre y variabilidad que presentan los objetos simbólicos y de otra por ser las distribuciones de posibilidad y los conjuntos difusos la forma más extendida de representación de la incertidumbre en los árboles de Segmentación que se presentan en el capítulo 2. Además, en el capítulo 4 se presenta esta forma de incertidumbre en la extensión del método desarrollado en esta Memoria a otros datos simbólicos. La generalización por datos y objetos simbólicos se introduce como antecedente a la generalización por objetos simbólicos de los estratos que se obtiene en los capítulos 3 y 4.

1.2 Análisis de Datos

1.2.1 Variables monoevaluadas

Sea $\Omega = \{\omega_1, ..., \omega_n\}$ un *conjunto de individuos* y sea $\mathcal{Y}$ un conjunto o dominio de posibles valores observados. A continuación se definen las variables monoevaluadas del Análisis de Datos.

Definición 1.1 ***Variable monoevaluada*** *definida en* Ω*. Se dice que* Y *es una variable monoevaluada con dominio* $\mathcal{Y}$ *si es una aplicación* $Y : \Omega \longrightarrow \mathcal{Y}$ *tal que dado* $\omega \in \Omega$ *le asocia un único valor* $Y(\omega) \in \mathcal{Y}$*, que es la* ***descripción del individuo*** ω *en* $\mathcal{Y}$ ***dada por la variable monoevaluada*** Y*. Se dice que* Y *es una* ***variable categórica monoevaluada*** *si el dominio* $\mathcal{Y}$ *es un conjunto finito cuyos valores no permiten establecer una relación de orden entre ellos.*

Sin pérdida de generalidad, se asume en lo sucesivo que la imagen de una variable Y coincide con $\mathcal{Y}$.

Se puede extender la definición anterior al caso multivariante. Sean $Y_1, ...Y_p$, p variables categóricas monoevaluadas definidas en Ω con dominios $\mathcal{Y}_1, ..., \mathcal{Y}_p$,

respectivamente. Sea $\mathcal{Y} := \mathcal{Y}_1 \times ... \times \mathcal{Y}_p$ el producto cartesiano de los dominios $\mathcal{Y}_j$. El **vector de variables categóricas monoevaluadas** $Y = (Y_1, ...Y_p)$ definido en Ω, es la aplicación:

$$\begin{aligned} Y: \ & \Omega \longrightarrow \mathcal{Y} \\ & \omega \longmapsto Y(\omega) = (Y_1(\omega), ...Y_p(\omega)) \end{aligned} \tag{1.1}$$

que a un individuo $\omega \in \Omega$ asocia el vector $Y(\omega) = (Y_1(\omega), ...Y_p(\omega))$. El vector $Y(\omega) \in \mathcal{Y}$ es la **descripción** del individuo ω en $\mathcal{Y}$ definida por las variables Y_j y el conjunto $\mathcal{Y}$ es el **conjunto de las descripciones de los elementos de** Ω.

1.2.2 Matriz de datos

La matriz de datos en el Análisis de Datos es la matriz $[X]$ cuyas filas $(X^{(i)}), i = 1, .., n$ representan observaciones del vector Y de variables monoevaluadas en n unidades que son los elementos de Ω.

El Análisis de Datos categóricos trata del estudio de las relaciones del conjunto Ω de individuos con las variables descriptivas $Y_1, ..., Y_p$ de los individuos en dicho conjunto, donde cada variable Y_j toma para cada individuo $\omega \in \Omega$ una única categoría de un dominio finito $\mathcal{Y}_j$. En el Análisis de Datos tradicional se estudia el par $(\Omega, \mathcal{Y}_1 \times ... \times \mathcal{Y}_p)$, con Ω conjunto de individuos e $\mathcal{Y}_1 \times ... \times \mathcal{Y}_p$, el conjunto de descripciones de elementos de Ω (Diday, 2000, Bock y Diday, 2000b).

1.3 Análisis de Datos Simbólicos

El principal objetivo de Análisis de Datos Simbólicos es extender las técnicas de Análisis de Datos a estructuras de datos más complejas. Si bien los datos simbólicos pueden describir individuos, un caso muy habitual es utilizarlos para describir información agregada de clases de individuos. También, en un nivel superior, pueden ser utilizados para describir información agregada de clases de

clases de individuos y así, sucesivamente...

Algunas estructuras de datos más complejas a los datos clásicos, en relación con datos categóricos son: las variables y datos multievaluados y los modales probabilistas, posibilistas y difusos. Es decir:

1. Un conjunto de valores puede describir un individuo, objeto o clase de individuos. Por ejemplo, para la variable $Colorpetalo$, un dato representado como $\{amarillo, naranja\}$ puede corresponder a:

   - una familia de rosas que tienen como color de sus pétalos o bien $amarillo$ o bien $naranja$.
   - una rosa cuyos pétalos son de color $amarillo$ y $naranja$.

   En esta representación, la variable $Colorpetalo$ es una variable simbólica multievaluada y el dato simbólico $\{amarillo, naranja\}$ es un dato multievaluado.

2. Los datos referentes a un individuo, objeto o clase de individuos pueden venir dados por una distribución de probabilidad: Por ejemplo, para la variable $Empleo$, la distribución $(sí(0.8), no(0.2))$ puede representar:

   - una subpoblación o clase de individuos de los cuales el 20% es desempleado. Este dato representa la *variación* en la variable $Empleo$ de una subpoblación.
   - un individuo que en la globalidad de su vida en activo ha estado el 20% del tiempo desempleado. Este dato representa una *incertidumbre.*

   En esta representación, la variable simbólica $Empleo$ es una variable modal probabilista y la distribución $(sí(0.8), no(0.2))$ es un dato modal probabilista.

3. Una distribución de posibilidad sobre un conjunto de valores puede describir individuos, objetos o clases de individuos.

4. Un conjunto de grados de pertenencia a varios conjuntos difusos pueden describir individuos, objetos o clases de individuos.

En 1.3.1 se presenta la matriz del Análisis de Datos Simbólicos y en 1.3.2, 1.3.3 y 1.3.4 se definen las variables simbólicas categóricas: las variables multievaluadas que asocian conjuntos de categorías y las variables modales que asocian conjuntos de categorías con unos modos respectivos.

1.3.1 Matriz de datos simbólicos

Sea $E = \{e_1, ..., e_n\}$ un conjunto de objetos. Como casos particulares más frecuentes se tiene que E es un subconjunto de Ω o un subconjunto de las clases de Ω, es decir, $E \subseteq \Omega$ o $E \subseteq \mathcal{P}(\Omega)$. En el segundo caso, los datos simbólicos correspondientes describen clases de individuos de Ω. Y sea $\mathcal{Y}$ un conjunto finito de elementos sin relación de orden. La descripción de un elemento $e \in E$ por una variable con dominio $\mathcal{Y}$ puede darse por:

- Un elemento del conjunto $\mathcal{Y}$. Este es el caso de una variable monoevaluada tratada en el Análisis de Datos clásico (véase definición 1.1).

- Un subconjunto de elementos del conjunto $\mathcal{Y}$. Este es el caso de una variable simbólica multievaluada definida en 1.3.2.

- Un subconjunto de elementos del conjunto $\mathcal{Y}$ donde cada uno de ellos es ponderado por un peso o modo. Este es el caso de una variable simbólica modal. Según el tipo de peso, se pueden distinguir varios tipos de variables modales:

  - Variables modales probabilistas, frecuentistas, ...(en 1.3.3).

– Variables modales posibilistas y difusas (en 1.3.4).

– Variables modales de creencia (en 1.4.4)

Las variables categóricas monoevaluadas son un caso particular de las variables simbólicas.

La matriz de datos en el Análisis de Datos Simbólicos es la matriz $[X]$ cuyas filas $(X^{(i)}), i = 1, .., n$ representan n unidades u objetos del conjunto E descritos por un vector de variables simbólicas $Y = (Y_1, ..., Y_p)$. Es decir, las celdas de una fila se corresponden con los datos simbólicos descritos por el vector Y aplicado a un elemento de E. Las matrices (1.10) y (1.26) son ejemplos de matrices de datos simbólicos.

El Análisis de Datos Simbólicos estudia las relaciones del conjunto E con las variables simbólicas $Y_1, ..., Y_p$ sobre los elementos de E. En el Análisis de Datos Simbólicos se estudia el par $(E, \mathcal{D})$, con E un conjunto de elementos y $\mathcal{D}$ un conjunto de descripciones simbólicas de elementos de E (Diday, 2000, Bock y Diday, 2000b). Una síntesis de conjuntos de descripciones simbólicas se muestra en 1.3.5.

A continuación, se definen los distintos tipos de variables y datos simbólicos, así como los distintos tipos de conjuntos de descripciones simbólicas.

1.3.2 Variables multievaluadas

Definición 1.2 ***Variable categórica multievaluada*** *definida en E. Se dice que Y es una variable categórica multievaluada si es una aplicación:*

$$\begin{aligned} Y: \quad & E \longrightarrow \mathcal{P}(\mathcal{Y}) \\ & e \longmapsto Y(e) \end{aligned} \tag{1.2}$$

$Y(e)$ es la ***descripción (multievaluada)*** *de un elemento $e \in E$ en $\mathcal{P}(\mathcal{Y})$ dada por la variable multievaluada Y y $\mathcal{P}(\mathcal{Y})$ es el* ***conjunto de descripciones***

(multievaluadas) *de los elementos de* E.

Se puede extender la definición anterior al caso multivariante. Sean $Y_1, ..., Y_p$, p variables categóricas multievaluadas definidas en E, con dominios respectivos $\mathcal{Y}_j$, sea $\mathcal{P}(\mathcal{Y}) := \mathcal{P}(\mathcal{Y}_1) \times ... \times \mathcal{P}(\mathcal{Y}_p)$ el producto cartesiano[1] de las partes de dichos dominios. El **vector de variables categóricas multievaluadas** Y definido en E es:

$$\begin{aligned} Y: \quad & E \longrightarrow \mathcal{P}(\mathcal{Y}) \\ & e \longmapsto Y(e) = (Y_1(e), ..., Y_p(e)) \end{aligned} \tag{1.3}$$

$Y(e)$ es la **descripción (multievaluada) de un elemento** $e \in E$ en $\mathcal{P}(\mathcal{Y})$ dada por el vector de variables multievaluadas Y y $\mathcal{P}(\mathcal{Y})$ es el **conjunto de descripciones (multievaluadas)** de los elementos de E.

En el caso en que $E \subseteq \mathcal{P}(\Omega)$, la variable Y (en (1.2)) o el vector Y (en (1.3)) se llama **descriptor (multievaluado) de clases de individuos** de Ω y $\mathcal{P}(\mathcal{Y})$ **conjunto de las descripciones (multievaluadas) de clases** de Ω, o de los elementos de $\mathcal{P}(\Omega)$.

Descripción de clase de individuos a partir de descripciones de individuos

Sea el conjunto $E = \{S_1, ..., S_m\} \subseteq \mathcal{P}(\Omega)$ un subconjunto de $\mathcal{P}(\Omega)$. Se presenta la forma más habitual de descripción de una clase por generalización de las descripciones de los individuos que la componen (Stéphan et al., 2000).

Caso univariante. Sea $\tilde{Y}$ una variable categórica monoevaluada definida en Ω con dominio $\mathcal{Y}$:

$$\begin{aligned} \tilde{Y}: \quad & \Omega \longrightarrow \mathcal{Y} \\ & \omega \longmapsto \widetilde{Y}(\omega) \end{aligned} \tag{1.4}$$

[1] Se elige esta notación por ser la utilizada en Bock y Diday, 2000a.

(Véase definición 1.1). A partir de la variable monoevaluada $\tilde{Y}$ de descripción de individuos se define la variable multievaluada Y en E de descripción de clase de individuos como:

$$\begin{array}{rl} Y: & E \longrightarrow \mathcal{P}(\mathcal{Y}) \\ & S_i \longmapsto Y(S_i) = \{\tilde{Y}(\omega)/\omega \in S_i\} \end{array} \tag{1.5}$$

$Y(S_i)$ es la **descripción de la clase** S_i en $\mathcal{P}(\mathcal{Y})$ dada por la **variable multievaluada** Y definida en $\mathcal{P}(\Omega)$ **inducida por la variable monoevaluada** $\tilde{Y}$ definida en Ω.

Caso multivariante. Sea $\tilde{Y} = (\tilde{Y}_1, ..., \tilde{Y}_p)$ un vector de variables categóricas monoevaluadas definidas en Ω con dominios respectivos $\mathcal{Y}_j$:

$$\begin{array}{rl} \tilde{Y}: & \Omega \longrightarrow \mathcal{Y}(= \mathcal{Y}_1 \times ... \times \mathcal{Y}_p) \\ & \omega \longmapsto \tilde{Y}(\omega) = (\tilde{Y}_1(\omega), ..., \tilde{Y}_p(\omega)) \end{array} \tag{1.6}$$

(Véase 1.2.1). Sea $\mathcal{P}(\mathcal{Y}) := \mathcal{P}(\mathcal{Y}_1) \times ... \times \mathcal{P}(\mathcal{Y}_p)$. A partir del vector de variables monoevaluadas $\tilde{Y}$ de descripción de individuos se define el vector de variables multievaluadas $Y = (Y_1, ..., Y_p)$ en E de descripción de clase de individuos como:

$$\begin{array}{rll} Y: & E \longrightarrow \mathcal{P}(\mathcal{Y}) & \\ & S_i \longmapsto Y(S_i) & = (Y_1(S_i), ..., Y_p(S_i)) \\ & & = (\{\tilde{Y}_1(\omega)/\omega \in S_i\}, ..., \{\tilde{Y}_p(\omega)/\omega \in S_i\}) \end{array} \tag{1.7}$$

$Y(S_i)$ es la **descripción de la clase** S_i en $\mathcal{P}(\mathcal{Y})$ dada por el **vector de variables multievaluados** Y definido en $\mathcal{P}(\Omega)$ **inducido por el vector de variables monoevaluadas** $\tilde{Y}$ definido en Ω**.** La descripción de la clase S_i de individuos se obtiene a partir de las categorías de $\mathcal{Y}$ que son observadas por el vector $\tilde{Y}$ en los individuos $\omega \in S_i$.

Ejemplo 1.1 ***Variables y datos multievaluados.*** *Sea el conjunto de in-*

dividuos $\Omega = \{\omega_1, ..., \omega_7\}$ *descrito por las variables categóricas monoevaluadas* $\widetilde{Y}_1 = \widetilde{sexo}$ *e* $\widetilde{Y}_2 = \widetilde{profesión}$ *con dominios respectivos* $\mathcal{Y}_1 = \{varón, mujer\}$ *e* $\mathcal{Y}_2 = \{informática, secretaria, administrativa\}$. *La matriz de los individuos se representa por:*

$$\begin{pmatrix} id & \widetilde{sexo} & \widetilde{profesión} \\ \omega_1 & mujer & informática \\ \omega_2 & mujer & secretaria \\ \omega_3 & mujer & secretaria \\ \omega_4 & varón & informática \\ \omega_5 & varón & administrativa \\ \omega_6 & varón & informática \\ \omega_7 & varón & administrativa \end{pmatrix} \tag{1.8}$$

$$Sea \qquad E \subset \mathcal{P}(\Omega), \qquad E = \{S_1, S_2\} = \{\{\omega_1, \omega_2, \omega_3\}, \{\omega_4, \omega_5, \omega_6, \omega_7\}\} \tag{1.9}$$

A partir de los descriptores de individuos $\widetilde{sexo}$ *y* $\widetilde{profesión}$ *se pueden definir los descriptores de clase de individuos sexo y profesión según (1.5). Las variables multievaluadas sexo y profesión se definen como:*

$$\begin{aligned} sexo: \quad & E \longrightarrow \mathcal{P}(\{varón, mujer\}) \\ & S_i \longmapsto sexo(S_i) = \{\widetilde{sexo}(\omega)/\omega \in S_i\} \end{aligned}$$

$$\begin{aligned} y \qquad profesión: \quad & E \longrightarrow \mathcal{P}(\{informática, secretaria, administrativa\}) \\ & S_i \longmapsto profesión(S_i) = \{\widetilde{profesión}(\omega)/\omega \in S_i\} \end{aligned}$$

Así, por ejemplo para la clase de individuos S_1, *se tiene que* $sexo(S_1) = \{mujer\}$ *y* $profesión(S_1) = \{informática, secretaria\}$ *y el vector de descripciones mul-*

tievaluadas (véase (1.7)) para la clase S_1 *es:*

$$\begin{aligned}(sexo, profesión)(S_1) &= (sexo(S_1), profesión(S_1)) \\ &= (\{mujer\}, \{informática, secretaria\})\end{aligned}$$

En este caso, la matriz de datos simbólicos que representa E *es:*

$$\begin{pmatrix} ID & sexo & profesión \\ S_1 & \{mujer\} & \{informática, secretaria\} \\ S_2 & \{varón\} & \{informática, administrativa\} \end{pmatrix} \tag{1.10}$$

Las variables categóricas monoevaluadas son un caso particular de las variables categóricas multievaluadas que describen los elementos de E por categorías únicas.

1.3.3 Variables modales probabilistas

Una variable modal es aquella que describe un elemento del conjunto E no sólo por un subconjunto de elementos del conjunto $\mathcal{Y}$ sino también por unos modos o pesos de cada uno de ellos. Las variables modales probabilistas asocian a cada elemento de E una distribución de probabilidad o de frecuencias que puede ser:

- estimada de la observación de una variable monoevaluada sobre un individuo, en diversos instantes de tiempo.

- derivada de la observación de una variable monoevaluada en una clase de individuos, estimadas las probabilidades como frecuencias relativas.

- no derivada de la observación directa, sino que es una distribución de probabilidad subjetiva derivada de un conocimiento 'a priori' o que tiene en cuenta la imprecisión o incertidumbre en la recogida de datos. Las variables

modales posibilistas y los conjuntos difusos, introducidos en la siguiente sección, también tiene en cuenta la imprecisión e incertidumbre.

También las variables modales pueden representar para cada una de las categorías una frecuencia, en lugar de una probabilidad o una posibilidad.

Sea $\mathcal{Y} = \{z_1, ..., z_x\}$, y sea $\mathcal{M}(\mathcal{Y}) = \{q/q$ es una distribución de probabilidad definida en $\mathcal{Y}\}$, el **conjunto de descripciones modales probabilistas** de elementos de E. Una descripción $q \in \mathcal{M}(\mathcal{Y})$ se define como:

$$\begin{aligned} q: \quad & \mathcal{Y} \longrightarrow [0,1] \\ & z_i \longmapsto q(z_i) \end{aligned} \tag{1.11}$$

con $\sum_{i=1,...,x} q(z_i) = 1$.

Se identifica el dato simbólico o descripción simbólica q (en (1.11)) con

$$q \equiv (z_1 q(z_1), ..., z_x q(z_x)) \tag{1.12}$$

También se identifica con la expresión correspondiente a (1.12) en la que desaparecen los términos que no se encuentran en el soporte de q.

Definición 1.3 ***Variable modal probabilista*** *definida en E. Se dice que Y es una variable modal probabilista si es una aplicación*

$$\begin{aligned} Y: \quad & E \longrightarrow \mathcal{M}(\mathcal{Y}) \\ & e \longmapsto Y(e) = q_e \end{aligned} \tag{1.13}$$

tal que dado $e \in E$ le asocia $Y(e) = q_e$, donde q_e es una distribución de probabilidad en el conjunto $\mathcal{Y}$ de posibles valores de observación completado por una $\sigma-$álgebra.

$Y(e)$ es la **descripción modal probabilista** (en $\mathcal{M}(\mathcal{Y})$) del elemento $e \in E$ dada por la variable modal probabilista Y. En el caso de que $E \subseteq \mathcal{P}(\Omega)$, la

variable Y es un **descriptor modal probabilista de clases de individuos** de Ω y $\mathcal{M}(\mathcal{Y})$ el **conjunto de las descripciones modales probabilistas de clases** de Ω, o de los elementos de $\mathcal{P}(\Omega)$.

La definición 1.3 de variable modal probabilista se puede extender a una variable modal cuyos modos asociados a las categorías de $\mathcal{Y}$ son frecuencias o pesos (véase Bock, 2000a).

Se puede extender la definición 1.3 al caso multivariante. Sean $Y_1, ..., Y_p$, p variables modales probabilistas definidas en E, con dominios respectivos $\mathcal{Y}_j$ y conjuntos de descripciones respectivos $\mathcal{M}(\mathcal{Y}_j)$. Sea $\mathcal{M}(\mathcal{Y}\ ) := \mathcal{M}(\mathcal{Y}_1) \times ... \times \mathcal{M}(\mathcal{Y}_p)$ el producto cartesiano[2] de dichos conjuntos de descripciones. El **vector de variables modales probabilistas** $Y = (Y_1, ..., Y_p)$ se define como:

$$\begin{array}{rl} Y: & E \longrightarrow \mathcal{M}(\mathcal{Y}) \\ & e \longmapsto Y(e) = (Y_1(e), ..., Y_p(e)) = (q_{e,1}, ..., q_{e,p}) \end{array} \tag{1.14}$$

donde $q_{e,j}$ es una distribución de probabilidad definida como:

$$\begin{array}{rl} q_{e,j}: & \mathcal{Y}_j \longrightarrow [0,1] \\ & y \longmapsto q_{e,j}(y) \end{array} \tag{1.15}$$

para $j \in \{1, ..., p\}$.

El conjunto $\mathcal{M}(\mathcal{Y})$ es el **conjunto de descripciones modales probabilistas** de los elementos de E. Dado $e \in E$, $Y(e) \in \mathcal{M}(\mathcal{Y})$ es la **descripción modal probabilista** de e dada por el vector de variables modales Y.

En el caso de que $E \subseteq \mathcal{P}(\Omega)$, el vector Y en (1.14) se llama **descriptor modal probabilista de clases de individuos** de Ω y $\mathcal{M}(\mathcal{Y})$ **conjunto de las descripciones modales probabilistas de clases** de Ω, o de los elementos de $\mathcal{P}(\Omega)$.

[2]Se elige esta notación por ser la utilizada en Bock y Diday, 2000a.

Descripción de clase de individuos a partir de descripciones de individuos

Sea $E = \{S_1, ..., S_m\} \subseteq \mathcal{P}(\Omega)$ un subconjunto de $\mathcal{P}(\Omega)$. Se presenta la forma más habitual de descripción de la clase por generalización de las descripciones de los individuos de dicha clase (Stéphan et al., 2000).

Caso univariante. Sea $\tilde{Y}$ una variable categórica monoevaluada definida en Ω con dominio $\mathcal{Y}$ definida por:

$$\begin{array}{rl} \tilde{Y}: & \Omega \longrightarrow \mathcal{Y} \\ & \omega \longmapsto \widetilde{Y}(\omega) \end{array} \tag{1.16}$$

(Véase definición 1.1). Sea $\mathcal{M}(\mathcal{Y}) := \{q/q$ es una distribución de probabilidad definida en $\mathcal{Y}\}$. A partir de la variable monoevaluada $\tilde{Y}$ de descripción de individuos se define la variable modal Y de descripción de clase de individuos como:

$$\begin{array}{rl} Y: & E \longrightarrow \mathcal{M}(\mathcal{Y}) \\ & S_i \longmapsto Y(S_i) = q_{S_i} \end{array} \tag{1.17}$$

donde la distribución de probabilidad q_{S_i} se define como:

$$\begin{array}{rl} q_{S_i}: & \mathcal{Y} \longrightarrow [0,1] \\ & y \longmapsto q_{S_i}(y) = \frac{Card(\{\omega \in S_i / \tilde{Y}(\omega) = y\})}{Card(S_i)} \end{array} \tag{1.18}$$

La distribución de probabilidad de una clase de individuos $S_i \in \mathcal{P}(\Omega)$ en $\mathcal{M}(\mathcal{Y})$ se obtiene a partir de las frecuencias relativas de las categorías de $\mathcal{Y}$ que son observadas por la variable $\tilde{Y}$ en los individuos $\omega \in S_i$, siguiendo la tendencia frecuentista de la probabilidad.

La distribución $Y(S_i) = q_{S_i}$ es la **descripción de la clase de individuos** $S_i \in \mathcal{P}(\Omega)$ definida por una **variable modal probabilista** Y definida en $\mathcal{P}(\Omega)$ **inducida por una variable monoevaluada** $\tilde{Y}$ definida en Ω.

Caso multivariante. Sea $\tilde{Y} = (\tilde{Y}_1, ..., \tilde{Y}_p)$ un vector de variables categóricas monoevaluadas definidas en Ω con dominios respectivos $\mathcal{Y}_j$:

$$\begin{array}{rl} \tilde{Y}: & \Omega \longrightarrow \mathcal{Y}(= \mathcal{Y}_1 \times ... \times \mathcal{Y}_p) \\ & \omega \longmapsto \tilde{Y}(\omega) = (\tilde{Y}_1(\omega), ..., \tilde{Y}_p(\omega)) \end{array} \tag{1.19}$$

(Véase 1.2.1). Sea $\mathcal{M}(\mathcal{Y}\ ) := \mathcal{M}(\mathcal{Y}_1) \times ... \times \mathcal{M}(\mathcal{Y}_p)$. A partir del vector de variables monoevaluadas $\tilde{Y}$ de descripción de individuos se define el vector de variables modales probabilistas $Y = (Y_1, ..., Y_p)$ en E de descripción de clase de individuos como:

$$\begin{array}{rl} Y: & E \longrightarrow \mathcal{M}(\mathcal{Y}) \\ & S_i \longmapsto Y(S_i) = (Y_1(S_i), ..., Y_p(S_i)) = (q_{S_i 1}, ..., q_{S_i p}) \end{array} \tag{1.20}$$

donde la distribución de probabilidad $q_{S_i j}$, para $j \in \{1, ..., p\}$ se define como:

$$\begin{array}{rl} q_{S_i j}: & \mathcal{Y}_j \longrightarrow [0, 1] \\ & y_j \longmapsto q_{S_i j}(y_j) = \frac{Card(\{\omega \in S_i / \tilde{Y}_j(\omega) = y_j\})}{Card(S_i)} \end{array} \tag{1.21}$$

$Y(S_i)$ es la **descripción de la clase de individuos** $S_i \in \mathcal{P}(\Omega)$ el **vector de variables modales probabilistas** definido $\mathcal{P}(\Omega)$, **inducido por** el **vector** $\tilde{Y}$ **de variables monoevaluadas** definido en Ω.

Ejemplo 1.2 ***Variables y datos probabilistas.*** *Sea el conjunto de individuos* $\Omega = \{\omega_1, ..., \omega_7\}$ *del ejemplo 1.1. La descripción de las variables y la matriz de los individuos pueden verse en el ejemplo 1.1 en (1.8).*

Sea $E = \{S_1, S_2\} = \{\{\omega_1, \omega_2, \omega_3\}, \{\omega_4, \omega_5, \omega_6, \omega_7\}\} \subset \mathcal{P}(\Omega)$. *A partir de los descriptores de individuos* $\widetilde{sexo}$ *y* $\widetilde{profesión}$ *se definen los descriptores de clase de*

individuos sexo y profesión como variables modales probabilistas según (1.17):

$$\begin{array}{rl} sexo: & E \longrightarrow \mathcal{M}(\{varón, mujer\}) \\ & S_i \longmapsto q_{sexo,i} \end{array} \tag{1.22}$$

con $q_{sexo,i}$, definido según (1.18) por:

$$\begin{array}{rll} q_{sexo,i}: & \{varón, mujer\} & \longrightarrow [0,1] \\ & varón & \longmapsto \frac{Card(\{\omega \in S_i / \widetilde{sexo}(\omega) = varón\})}{Card(S_i)} \\ & mujer & \longmapsto \frac{Card(\{\omega \in S_i / \widetilde{sexo}(\omega) = mujer\})}{Card(S_i)} \end{array} \tag{1.23}$$

La variable modal probabilista profesión se define:

$$\begin{array}{rl} profesión: & E \longrightarrow \mathcal{M}(\{informática, secretaria, administrativa\}) \\ & S_i \longmapsto q_{profesión,i} \end{array} \tag{1.24}$$

con $q_{profesión,i}$, definido según (1.18) por:

$$\begin{array}{rll} q_{profesión,i}: & \{informática, secretaria, & \longrightarrow [0,1] \\ & administrativa\} & \\ & informática & \longmapsto \frac{Card(\{\omega \in S_i / \widetilde{profesión}(\omega) = informática\})}{Card(S_i)} \\ & secretaria & \longmapsto \frac{Card(\{\omega \in S_i / \widetilde{profesión}(\omega) = secretaria\})}{Card(S_i)} \\ & administrativa & \longmapsto \frac{Card(\{\omega \in S_i / \widetilde{profesión}(\omega) = administrativa\})}{Card(S_i)} \end{array} \tag{1.25}$$

Así, por ejemplo para la clase de individuos S_1, se tiene que $sexo(S_1) = (mujer1)$ (según (1.23)) y $profesión(S_1) = (informática\frac{1}{3}, secretaria\frac{2}{3})$ (según (1.25)). El vector de descripciones modales (véase (1.20)) para la clase S_1 es:

$$\begin{array}{rcl} (sexo, profesión)(S_1) & = & (sexo(S_1), profesión(S_1)) \\ & = & ((mujer1), (informática\frac{1}{3}, secretaria\frac{2}{3})) \end{array}$$

En este caso, la matriz de datos simbólicos que representa E es:

$$\begin{pmatrix} ID & sexo & profesión \\ S_1 & (mujer1) & (informática\frac{1}{3}, secretaria\frac{2}{3}) \\ S_2 & (varón1) & (informática\frac{1}{2}, administrativa\frac{1}{2}) \end{pmatrix} \tag{1.26}$$

Las variables categóricas monoevaluadas son un caso particular de las variables categóricas modales probabilistas que describen los elementos de E por distribuciones de probabilidad degeneradas.

1.3.4 Variables modales posibilistas

Las distribuciones de posibilidad y los conjuntos difusos son otras formas de representación de la incertidumbre y pueden encuadrarse en el marco de las variables y datos simbólicos. Según Diday (Diday (1991,1995a)) las *variables modales posibilistas* asocian a cada elemento de E una distribución de posibilidad sobre el conjunto de categorías. En este caso, el peso asociado a cada categoría para un elemento de E representa el grado que tiene dicha categoría de ser real o relevante para ese elemento de E.

Desde otro punto de vista, las categorías de una variable se pueden definir cómo conjuntos difusos y los elementos de E tienen unos grados de pertenencia a estos conjuntos difusos. Se extienden las variables modales posibilistas de Diday (1991, 1995) a variables expresadas por conjuntos difusos.

En esta sección se presentan las variables modales posibilistas y las variables modales posibilistas definidas por conjuntos difusos. Para extensión de los conceptos de posibilidad y conjuntos difusos, véase Zadeh, 1978 y Dubois y Prade, 1984.

Distribuciones de posibilidad

Definición 1.4 *Una **distribución de posibilidad** q definida en $\mathcal{Y}$ es una función:*

$$\begin{array}{rccc} q: & \mathcal{Y} & \longrightarrow & [0,1] \\ & y & \longmapsto & q(y) \end{array} \tag{1.27}$$

Se dice **normalizada** si $\exists y \in \mathcal{Y} | q(y) = 1$. Se dice que este elemento, es un elemento totalmente posible.

El **núcleo** de una distribución de posibilidad q es:

$$C(q) = \{y \in \mathcal{Y} | q(y) = 1\} \tag{1.28}$$

El **soporte** de una distribución de posibilidad q es:

$$S(q) = \{y \in \mathcal{Y} | q(y) > 0\} \tag{1.29}$$

Definición 1.5 *P es una **medida de posibilidad** definida sobre $\mathcal{P}(\mathcal{Y})$ si es una función*

$$\begin{array}{rccc} P: & \mathcal{P}(\mathcal{Y}) & \longrightarrow & [0,1] \\ & A & \longmapsto & P(A) \end{array} \tag{1.30}$$

tal que:

- $P(\mathcal{Y}) = 1$
- $P(\emptyset) = 0$
- $\forall A, B \in \mathcal{P}(\mathcal{Y})$, *se tiene que* $P(A \cup B) = \max\{P(A), P(B)\}$

Una medida de posibilidad cumple las siguientes propiedades:

- $\max\{P(A), P(A^c)\} = 1$. Dos sucesos contrarios pueden ser totalmente posibles simultáneamente, pero al menos uno lo es.

- Si $A \subseteq B$ entonces $P(A) \leq P(B)$

- $P(A) + P(A^c) \geq 1$

- $P(A \cap B) \leq \min\{P(A), P(B)\}$

Definición 1.6 *N es una **medida de necesidad** definida sobre $\mathcal{P}(\mathcal{Y})$ si es una función*

$$\begin{array}{rccl} N: & \mathcal{P}(\mathcal{Y}) & \longrightarrow & [0,1] \\ & A & \longmapsto & N(A) \end{array} \tag{1.31}$$

tal que:

- $N(\mathcal{Y}) = 1$

- $N(\emptyset) = 0$

- $\forall A, B \in \mathcal{P}(\mathcal{Y})$ *se tiene que* $N(A \cap B) = \min\{N(A), N(B)\}$

Una medida de necesidad N cumple las siguientes propiedades:

- $\min\{N(A), N(A^c)\} = 0$. Dos sucesos contrarios no puede ser necesarios simultáneamente[3].

- $N(A) + N(A^c) \leq 1$

- $N(A \cup B) \geqslant \max\{N(A), N(B)\}$

[3] $N(A) = 0$ significa una ausencia total de certeza respecto a A pero no que A sea necesariamente falsa.

Una medida de posibilidad P lleva asociada una medida de necesidad N definida como:

$$\begin{array}{rlll} N: & \mathcal{P}(\mathcal{Y}) & \longrightarrow [0,1] \\ & A & \longmapsto 1 - P(A^c) \end{array} \tag{1.32}$$

La necesidad de un suceso se corresponde con la imposibilidad de su complementario. Se cumplen las siguientes propiedades entre una medida de posibilidad P y una medida de necesidad N asociadas:

- $P(A) = 1 - N(A^c)$
- $N(A) \leq P(A)$
- $N(A) > 0 \Longrightarrow P(A) = 1$
- $P(A) < 1 \Longrightarrow N(A) = 0$, es decir, un suceso debe ser completamente posible para ser algo necesario.

A partir de una distribución de posibilidad normalizada q se pueden definir las medidas de posibilidad P y de necesidad N siguientes:

$$\begin{array}{rlll} P: & \mathcal{P}(\mathcal{Y}) & \longrightarrow [0,1] \\ & A & \longmapsto P(A) = \sup_{y \in A} q(z) \end{array} \tag{1.33}$$

$$\begin{array}{rlll} N: & \mathcal{P}(\mathcal{Y}) & \longrightarrow [0,1] \\ & A & \longmapsto N(A) = \inf_{y \in A^c} \{1 - q(z)\} \end{array} \tag{1.34}$$

Variables modales posibilistas

Sea $\mathcal{Y} = \{z_1, ..., z_x\}$, y sea $\mathcal{M}(\mathcal{Y}) = \{q | q$ es una distribución de posibilidad definida en $\mathcal{Y}\}$ el **conjunto de descripciones modales posibilistas** de elementos

de E. Una descripción $q \in \mathcal{M}(\mathcal{Y})$ se define como:

$$\begin{array}{rl} q: & \mathcal{Y} \longrightarrow [0,1] \\ & z_i \longmapsto q(z_i) \end{array} \tag{1.35}$$

Se identifica el dato simbólico o descripción simbólica q (en (1.35)) con

$$q \equiv (z_1 q(z_1), ..., z_x q(z_x)) \tag{1.36}$$

También se identifica con la expresión correspondiente a (1.36) en la que desaparecen los términos que no se encuentran en el soporte de q.

Definición 1.7 ***Variable modal posibilista*** *definida en E. Se dice que Y es una variable modal posibilista si es una aplicación*

$$\begin{array}{rlll} Y: & E & \longrightarrow & \mathcal{M}(\mathcal{Y}) \\ & e & \longmapsto & Y(e) = q_e \end{array} \tag{1.37}$$

con q_e una distribución de posibilidad definida en el conjunto $\mathcal{Y}$.

$\mathcal{M}(\mathcal{Y})$ es el **conjunto de descripciones modales posibilistas** de los elementos de E. Sea $e \in E$, $Y(e)$ es la **descripción modal posibilista** de e en $\mathcal{M}(\mathcal{Y})$. En el caso de que $E \subseteq \mathcal{P}(\Omega)$, la variable Y se llama **descriptor modal posibilista de clases de individuos** de Ω y $\mathcal{M}(\mathcal{Y})$ **conjunto de las descripciones modales posibilistas de clases** de Ω, o de los elementos de $\mathcal{P}(\Omega)$.

Se extiende en esta Memoria la definición de variable modal posibilista dada por Diday (1991, 1995) en la que imponía la condición de que las descripciones simbólicas q (en (1.35)) debían ser distribuciones de posibilidad normalizadas. A continuación, se extienden también las variables modales posibilistas a variables cuyas categorías vienen definidas por conjuntos difusos (véase definición 1.8).

Conjuntos difusos

Definición 1.8 *A es un **conjunto difuso** (Zadeh, 1965) de conjunto de referencia E si se caracteriza por la función de pertenencia:*

$$\begin{array}{rccl} q_A: & E & \longrightarrow & [0,1] \\ & e & \longmapsto & q_A(e) \end{array} \tag{1.38}$$

El valor $q_A(e)$ representa el grado de pertenencia de e al conjunto A. Un conjunto difuso es un conjunto con frontera imprecisa, es decir, los elementos del conjunto de referencia E pertenecen a él con un grado de pertenencia que toma valores en $[0,1]$. La función de pertenencia es una extensión de la función característica clásica de un subconjunto.

Zadeh (Zadeh, 1965) sugiere la utilización de los operadores min, max y el opuesto para la definición de las funciones de pertenencia de la intersección, unión y complementario de conjuntos difusos, que a su vez son conjuntos difusos. Posteriormente, se introducen las $T-$normas, $T-$conormas y operadores de negación, respectivamente, para estas operaciones entre conjuntos difusos que tienen como caso particular los definidos por Zadeh (Yager, 1980, Weber, 1983, Trillas et al., 1995). Recopilación de este punto puede verse en Amo, 1999.

Definición 1.9 *Una* **T$-$*norma o norma triangular*** *es una función* $T:[0,1]\times[0,1]\longrightarrow[0,1]$ *que verifica:*

$$\begin{array}{ll} T(1,x)=x & \forall x\in[0,1] \\ T(x,y)=T(y,x) & \forall x,y\in[0,1] \\ T(x,T(y,z))=T(T(x,y),z) & \forall x,y,z\in[0,1] \\ T(x,y)\leq T(u,v) & si\ 0\leq x\leq u\leq 1, 0\leq y\leq v\leq 1 \end{array}$$

Se deduce fácilmente que $T(0,x)=0$. Una $T-$norma puede verse como un operador de intersección o conjuntivo. Se tiene que el mínimo es la máxima

$T-$norma.

Definición 1.10 *Una* **T$-$*conorma*** *es una función* $S : [0,1] \times [0,1] \longrightarrow [0,1]$ *que verifica:*

$$\begin{array}{ll} S(0,x) = x & \forall x \in [0,1] \\ S(x,y) = S(y,x) & \forall x,y \in [0,1] \\ S(x,S(y,z)) = S(S(x,y),z) & \forall x,y,z \in [0,1] \\ S(x,y) \leq S(u,v) & si\ 0 \leq x \leq u \leq 1, 0 \leq y \leq v \leq 1 \end{array}$$

Una $T-$conorma se puede ver como un operador de unión o disyuntivo. Se tiene que el máximo es la mínima $T-$conorma. En Bandemer y Näther (1992), se presentan varias $T-$normas y $T-$conormas.

Definición 1.11 *Una* ***negación*** *es una función no creciente* $n : [0,1] \longrightarrow [0,1]$ *que verifica:*

$$n(0) = 1, n(1) = 0$$

Zadeh (Zadeh, 1978) ya nombra las *variables difusas* y actualiza la interpretación de la teoría de los conjuntos difusos introduciendo la teoría de la posibilidad donde un conjunto difuso es visto como una distribución de posibilidad, es decir, conjuntos de valores más o menos posibles de una variable auxiliar[4].

Las descripciones modales posibilistas en (1.36) pueden venir dadas por los grados de pertenencia de los elementos E a las categorías difusas de una variable

[4]Sea

$$Y : E \longrightarrow U \xrightarrow{\phi_F} [0,1]$$

una variable de rango continuo U definida sobre el conjunto E, y sea F un conjunto difuso de U con función de pertenencia ϕ_F. Este conjunto difuso se llama *restricción difusa*. Según Zadeh (Zadeh, 1978), este conjunto o restricción difusa puede verse como una distribución de posibilidad, es decir, valores más o menos posibles de una variable. En este planteamiento:

$$\text{Poss}(Y = u | Y \text{ es } F) = \phi_F(u)$$

Y. Las categorías de una variable modal posibilista Y se pueden definir como conjuntos difusos y los elementos de E tienen unos grados de pertenencia a estos conjuntos difusos. Es decir, la variable Y puede ser una tupla de conjuntos difusos $(z_1, ..., z_x)$ de conjuntos de referencia E definida como:

$$\begin{array}{rccl} Y: & E & \longrightarrow & [0,1] \times ...^x ... \times [0,1] \\ & e & \longmapsto & Y(e) = (q_{z_1}(e), ..., q_{z_x}(e)) \end{array} \tag{1.39}$$

con $(q_{z_1}(e), ..., q_{z_x}(e))$ la tupla de funciones de pertenencia aplicadas a un elemento e. El valor $q_{z_i}(e)$ representa el grado de pertenencia del elemento e al conjunto difuso z_i.

La tupla $Y(e) = (q_{z_1}(e), ..., q_{z_x}(e))$ es la **descripción modal posibilista** por conjuntos difusos de e en $\mathcal{M}(\mathcal{Y})$. Se realiza la identificación del dato simbólico:

$$(q_{z_1}(e), ..., q_{z_x}(e)) \equiv (z_1 q_{z_1}(e), ..., z_x q_{z_x}(e)) \tag{1.40}$$

también identificado con la expresión de la derecha en la que desaparecen los términos con $q_{z_i}(e) = 0$.

Nota 1.1 *La distinción de variables modales posibilistas y definidas por conjuntos difusos se debe principalmente a que existe una distinción semántica entre el concepto de distribución de posibilidad y el de conjunto difuso (Prade, 1985). Los valores dados por una distribución de posibilidad son valores que indican la mayor*

Se tiene que:

$$\begin{array}{rcl} \forall u & \in & U, \phi_{F \cup F^c}(u) = \max\{\phi_F(u), \phi_{F^c}(u)\} \geqslant \frac{1}{2} \\ \forall u & \in & U, \phi_{F \cap F^c}(z) = \min\{\phi_F(u), \phi_{F^c}(u)\} \leq \frac{1}{2} \end{array}$$

Peng y otros (Peng et al., 1991) también mencionan la intención y extensión de los conceptos y lo relacionan con las restricciones difusas. Dicen que los conjuntos difusos describen el grado de pertenencia de los objetos a los *conceptos*, es decir, la *extensión*. Y que los conceptos expresados por los conjuntos difusos pueden ser referidos a un universo U dado por Y. A esta variable la llaman *factor*. Los factores describen la *intención* de los conceptos.

o menor posibilidad de ser obtenidos por la variable si bien son excluyentes, mientras que los elementos pueden pertenecer a varios conjuntos difusos simultáneamente con diversos grados de pertenencia.

Un ejemplo de variable modal posibilista definida por conjuntos difusos puede ser la variable $Tiempo$ aplicada al conjunto de los días del año. La variable consta de las categorías $Nublado$ y $Soleado$ como conjuntos difusos. La observación de cada día del año se puede ver como un grado de pertenencia a $Nublado$ y un grado de pertenencia a $Soleado$.

Del mismo modo que las categorías del conjunto $\mathcal{Y}$ se pueden referir a conjuntos difusos de espacio de referencia E, un subconjunto de categorías A de $\mathcal{Y}$, puede asociarse a un conjunto difuso de espacio de referencia E. La función de pertenencia de este conjunto puede definirse a partir de una $T-$conorma o ser dotado de una función de pertenencia propia. En el primer caso, el grado de pertenencia de un individuo es la aplicación de la $T-$conorma[5] a los grados de pertenencia de los conjuntos difusos relativos a las categorías que componen A.

1.3.5 Conjunto de descripciones simbólicas

Como recapitulación a las variables y datos simbólicos, sea una variable definida en E como una aplicación:

$$Y : E \longrightarrow \mathcal{D} \tag{1.41}$$

con $\mathcal{D}$ un **conjunto de descripciones** de elementos de E **asociado al conjunto o dominio** $\mathcal{Y}$. Este conjunto se denota por $\mathcal{D}(\mathcal{Y})$ y puede ser:

- $\mathcal{D} = \mathcal{Y}$, en el caso de que Y sea una variable monoevaluada

[5]En esta y sucesivas referencias a $T-$normas y $T-$conormas, se extiende la definición de las mismas a más de dos valores. Por las propiedades conmutativa y transitiva de ambas, esta extensión es directa.

- $\mathcal{D} = \mathcal{P}(\mathcal{Y})$, en el caso de que Y sea una variable simbólica multievaluada

- $\mathcal{D} = \mathcal{M}(\mathcal{Y})$, en el caso de que Y sea una variable simbólica modal con:

  - $\mathcal{M}(\mathcal{Y}) = \mathcal{M}^{\Pr ob}(\mathcal{Y}) = \{q : (\mathcal{Y}, \mathcal{P}(\mathcal{Y})) \longrightarrow [0,1]/q$ es una distribución de probabilidad$\}$

  - o bien, $\mathcal{M}(\mathcal{Y}) = \mathcal{M}^{Pos}(\mathcal{Y}) = \{q : \mathcal{Y} \longrightarrow [0,1]/q$ es una distribución de posibilidad o son grados de pertenencia a categorías difusas$\}$

Se dice que una **descripción** $d \in \mathcal{D}$ está **asociada al conjunto o dominio** $\mathcal{Y}$. En el caso de que $E \subseteq \mathcal{P}(\Omega)$, el conjunto $\mathcal{D}$ se llama **conjunto de descripciones de clases de** Ω.

La generalización a un conjunto de descripciones simbólicas dada por un vector de variables simbólicas es directa. Un vector de variables simbólicas asocia a un elemento de E un vector de descripciones de los conjuntos de descripciones asociados, es decir, un elemento del conjunto de descripciones

$$\mathcal{D} = \mathcal{D}_1 \times ... \times \mathcal{D}_p \tag{1.42}$$

con $\mathcal{D}_j$ el conjunto de descripciones asociado al conjunto o dominio $\mathcal{Y}_j$. Este conjunto se denota por $\mathcal{D}(\mathcal{Y})$.

Por simplicidad en la notación se consideran los conjuntos de descripciones asociados a un vector de conjuntos o dominios $\mathcal{Y} = \mathcal{Y}_1 \times ..., \mathcal{Y}_p$ como los conjuntos $\mathcal{Y}$, $\mathcal{P}(\mathcal{Y})$, $\mathcal{M}^{\Pr ob}(\mathcal{Y})$ y $\mathcal{M}^{Pos}(\mathcal{Y})$ cuando los conjuntos de descripciones asociados a los conjuntos o dominios $\mathcal{Y}_j$ son todos del mismo tipo: monoevaluado, mul-

tievaluado, probabilista o posibilista, respectivamente. Es decir,

$$\mathcal{Y} \ : \ = \mathcal{Y}_1 \times ... \times \mathcal{Y}_p \tag{1.43a}$$

$$\mathcal{P}(\mathcal{Y}) \ : \ = \mathcal{P}(\mathcal{Y}_1) \times ... \times \mathcal{P}(\mathcal{Y}_p) \tag{1.43b}$$

$$\mathcal{M}^{\Pr ob}(\mathcal{Y}) \ : \ = \mathcal{M}^{\Pr ob}(\mathcal{Y}_1) \times ... \times \mathcal{M}^{\Pr ob}(\mathcal{Y}_p) \tag{1.43c}$$

$$\mathcal{M}^{Pos}(\mathcal{Y}) \ : \ = \mathcal{M}^{Pos}(\mathcal{Y}_1) \times ... \times \mathcal{M}^{Pos}(\mathcal{Y}_p) \tag{1.43d}$$

1.4 Objetos simbólicos

En 1.3 se han introducido los datos simbólicos, es decir, estructuras de datos de mayor complejidad que los datos clásicos. En esta sección, se introducen los objetos simbólicos. Un objeto simbólico se describe por unas variables monoevaluadas o simbólicas referidas a un conjunto E, por unos datos simbólicos y por unas *relaciones de dominio* que permiten la vuelta al conjunto E para obtener aquellos elementos de E cuyas descripciones dadas por dichas variables se *relacionan* con la descripción dada por los datos simbólicos. Un objeto simbólico por una parte representa la *intención* de un concepto y por otra proporciona una herramienta para la obtención de la *extensión* de esa intención o concepto en un conjunto de individuos o en una base de datos.

En 1.4.1 se definen las relaciones de dominio que permiten *relacionar* o comparar pares de descripciones. Estas relaciones son necesarias para relacionar la descripción de un elemento de E dada por una variable con un dato simbólico. En 1.4.2 y 1.4.3 se definen los objetos simbólicos más habituales, llamados *eventos* y *aserciones*. En 1.4.4 se presentan otros tipos de objetos simbólicos. En 1.4.5, antecedentes a la generalización por objetos simbólicos.

Sea el conjunto $E = \{e_1, ..., e_n\}$ de elementos descritos por p variables monoevaluadas o simbólicas $Y_1, ..., Y_p$ definidas en E con dominios finitos $\mathcal{Y}_1, ..., \mathcal{Y}_p$. Se puede considerar que todas las variables son simbólicas, dado que una variable

monoevaluada no es más que un caso particular de una variable simbólica sin más que considerar $E = \Omega$ y las variables Y_j, monoevaluadas. Por otra parte, el conjunto de referencia E suele coincidir con el conjunto Ω, y es por esto que los elementos de E son *individuos* en esta sección.

1.4.1 Relaciones de dominio

Sean $(\mathcal{D}_1, ..., \mathcal{D}_p)$ y $(\mathcal{D}'_1, ..., \mathcal{D}'_p)$, dos colecciones de p conjuntos de descripciones asociados a los dominios $\mathcal{Y}_1, ..., \mathcal{Y}_p$. En lo sucesivo, estos conjuntos $\mathcal{D}_j$, $\mathcal{D}'_j$ se denominan conjuntos de descripciones de clase, ya que es habitual que los datos simbólicos describan clases de individuos de Ω. Sin embargo, puedan ser conjuntos de descripciones de elementos de E, simplemente.

Definición 1.12 ***Relación de dominio****. Sean $\mathcal{D}$ y $\mathcal{D}'$ dos conjuntos de descripciones de clase asociados a un mismo dominio, $\mathcal{D} \times \mathcal{D}'$ su producto cartesiano, una relación de dominio $\mathcal{R}$ definida en $\mathcal{D} \times \mathcal{D}'$ es una aplicación:*

$$\begin{aligned} \mathcal{R}: \ & \mathcal{D} \times \mathcal{D}' \longrightarrow \mathcal{L} \\ & (d, d') \longmapsto \mathcal{R}(d, d') := [d\mathcal{R}d'] \end{aligned} \tag{1.43}$$

que a cada par de descripciones $(d, d') \in \mathcal{D} \times \mathcal{D}'$ le asocia un valor, denotado por $[d\mathcal{R}d']$ que mide el grado de adecuación o conexión de ambas descripciones.

$\mathcal{L}$ es el **conjunto de comparación de descripciones**. El valor $[d\mathcal{R}d']$ es el **nivel de relación entre** las **descripciones** d y d', o nivel de relación de la descripción d con la descripción d'. En general, $\mathcal{L} = \{0, 1\}$ o $\mathcal{L} = [0, 1]$. En 1.4.4, se presenta un ejemplo de un conjunto $\mathcal{L}$ no contenido en el intervalo $[0, 1]$, como una extensión de conjunto de comparación de descripciones a conjuntos ordenados cualesquiera.

El nivel de relación entre dos descripciones $[d\mathcal{R}d']$ no es más que el resultado de la comparación entre ellas según una relación $\mathcal{R}$. En el caso de que tenga sentido definirlos, se establece que el nivel de relación de una descripción con el

conjunto vacío es nulo, y con el conjunto $\mathcal{Y}$ es la unidad:

$$[d\mathcal{R}\emptyset] = [\emptyset\mathcal{R}d] = 0 \quad (1.45)$$

$$[d\mathcal{R}\mathcal{Y}] = [\mathcal{Y}\mathcal{R}d] = 1 \quad (1.46)$$

Aunque en principio las relaciones de dominio no se encuentran así definidas axiomáticamente, las relaciones propuestas en la literatura verifican estas dos propiedades.

Definición 1.13 ***Relación de dominio booleana.*** *$\mathcal{R}$ es una relación de dominio booleana si el conjunto de comparación de descripciones es $\mathcal{L}$ =$\{0, 1\}$. Dado un par de descripciones $(d, d') \in \mathcal{D} \times \mathcal{D}'$,*

- *cuando $[d\mathcal{R}d'] = 1$, entonces la relación entre d y d' es verdad o d y d' se relacionan; y,*
- *cuando $[d\mathcal{R}d'] = 0$, entonces la relación entre d y d' es falsa o d y d' no se relacionan.*

En lo sucesivo, se identifica en una relación booleana el valor 1 con *verdad* o v y el valor 0 con $falso$ o f.

En el caso de que el conjunto de comparación de descripciones sea $\mathcal{L}$ =$[0, 1]$, se dice que $\mathcal{R}$ es una **relación difusa**. Cuando el valor de $[d\mathcal{R}d']$ se acerque más a 1, entonces la relación es más fuerte para el par (d, d'); y cuando se acerque más a 0, entonces la relación es más débil para el par (d, d'). Para mayor documentación en relaciones difusas puede consultarse Bandemer y Näther (1992).

Si bien Bock y Diday (Bock y Diday, 2000b) presentan genéricamente la denominación de *relación difusa* para las relaciones no booleanas, aquí se distingue la denominación de **relación probabilista** cuando alguno de los conjuntos de descripciones en la relación sea un conjunto de descripciones modales probabilistas y cuando se aplique el cálculo de probabilidades o funciones entre distribuciones de probabilidad (Ejemplo: similaridades, divergencias, etc...).

Las relaciones de dominio probabilistas o difusas se denotan por $\sim$.

Un tipo particular de relaciones de dominio que se establecen entre descripciones simbólicas son las de tipo *matching* (Esposito et al., 2000a) que representan la comparación de dos descripciones dada una de ellas como patrón de referencia, siendo por lo general no simétricas. Una relación de este tipo puede establecerse entre una descripción genérica de clase tomada como patrón de referencia y la descripción de un individuo para establecer si el individuo puede ser considerado como un elemento de la clase descrita.

Definición 1.14 ***Producto de relaciones de dominio** o **relación producto**. Sea una colección de relaciones de dominio $(\mathcal{R}_1, ..., \mathcal{R}_p)$, cada $\mathcal{R}_j$ definida en el producto cartesiano $\mathcal{D}_j \times \mathcal{D}'_j$. Sean $\mathcal{D} := \mathcal{D}_1 \times ... \times \mathcal{D}_p$ y $\mathcal{D}' := \mathcal{D}'_1 \times ... \times \mathcal{D}'_p$ los correspondientes productos cartesianos de los conjuntos de descripciones. La relación producto $\mathcal{R} = \mathcal{R}_1 \times ... \times \mathcal{R}_p$ definida en $\mathcal{D} \times \mathcal{D}'$ es la aplicación:*

$$\begin{array}{rlll} \mathcal{R}: & \mathcal{D} \times \mathcal{D}' & \longrightarrow \mathcal{L}' & \\ & (d, d') & \longmapsto \mathcal{R}(d, d') := [d\mathcal{R}d'] & := g(\{[d_j \mathcal{R}_j d'_j], j = 1, ...p\}) \\ & & & = \wedge_{j=1,..,p}[d_j \mathcal{R}_j d'_j] \end{array} \tag{1.47}$$

que a cada par de descripciones $(d, d') \in \mathcal{D} \times \mathcal{D}'$, $d = (d_1, ..., d_p)$, $d' = (d'_1, ..., d'_p)$, le asocia un valor, denotado por $[d\mathcal{R}d']$. El conjunto $\mathcal{L}'$ es el conjunto comparación de descripciones. La aplicación $g(.)$ es una aplicación simétrica.

Notación 1.1 *La aplicación $g(.)$ se denota por $\wedge$ si bien este operador no es siempre el operador conjuntivo lógico estándar.*

La aplicación g, llamada **aplicación de combinación de niveles de relación** (o de adecuación) está definida como:

$$\begin{array}{rlll} g: & \mathcal{L} \times ..^{(p)}.. \times \mathcal{L} & \longrightarrow \mathcal{L}' \\ & (l_1, ..., l_p) & \longmapsto g(l_1, ..., l_p) \end{array} \tag{1.48}$$

Por lo general, $\mathcal{L}' \subseteq [0,1]$. En 1.4.4, hay un ejemplo de dos conjuntos $\mathcal{L}$ y $\mathcal{L}'$ no contenidos en el intervalo $[0,1]$, como una extensión de estos conjuntos ordenados cualesquiera.

Se considera que la función $g(.)$ verifica (salvo que se diga lo contrario):

$$g(1,v) = v, \forall v \in [0,1] \tag{1.49}$$

$$g(0,v) = 0, \forall v \in [0,1] \tag{1.50}$$

y de modo similar si la función $g(.)$ se aplica a mayor número de argumentos con alguno de ellos el valor unidad o el valor nulo:

$$g(1,l_2,...,l_p) = g(l_2,...,l_p), \forall l_2,...,l_p \in [0,1] \tag{1.51}$$

$$g(0,l_2,...,l_p) = 0, \forall l_2,...,l_p \in [0,1] \tag{1.52}$$

El valor $[d\mathcal{R}d']$ es **nivel de relación entre** las **descripciones** d y d', o nivel de relación de la descripción d con la descripción d'. El nivel de relación entre dos descripciones $[d\mathcal{R}d']$ por una relación producto $\mathcal{R}$ mide el grado de adecuación entre ellas como la aplicación de una función $g(.)$ a los niveles de relación $[d_j\mathcal{R}d'_j]$.

Definición 1.15 *Un **producto de relaciones de dominio** es **booleano** si el conjunto de comparación de descripciones $\mathcal{L}'$ en (1.47) es el conjunto $\{0,1\}$.*

Se aplica la distinción entre relaciones producto difusas y probabilistas como en el caso de las relaciones de dominio.

Proposición 1.1 *Sea una colección de relaciones de dominio $(\mathcal{R}_1,...,\mathcal{R}_p)$, cada $\mathcal{R}_j$ una relación de dominio booleana definida en el producto cartesiano $\mathcal{D}_j \times \mathcal{D}'_j$.*

La relación producto $\mathcal{R} = \mathcal{R}_1 \times ... \times \mathcal{R}_p$*:*

$$\begin{array}{rll} \mathcal{R}: & \mathcal{D} \times \mathcal{D}' & \longrightarrow \mathcal{L}(=[0,1]) \\ & (d, d') & \longmapsto [d\mathcal{R}d'] := g(\{[d_j\mathcal{R}_j d'_j], j = 1, ...p\}) = \wedge_{j=1...p}[d_j\mathcal{R}_j d'_j] \end{array} \tag{1.53}$$

con la función g *el operador conjuntivo lógico estándar, es una relación producto booleana.*

Demostración. Si $\mathcal{R}_j$ es una relación de dominio booleana definida en $\mathcal{D}_j \times \mathcal{D}'_j$, se tiene que:

$$\begin{array}{rll} \mathcal{R}_j: & \mathcal{D}_j \times \mathcal{D}'_j & \longrightarrow \{0,1\} \\ & (d_j, d'_j) & \longmapsto [d_j\mathcal{R}d'_j] \end{array} \tag{1.54}$$

Para cualquier par de descripciones $(d, d') \in \mathcal{D} \times \mathcal{D}'$, $d = (d_1, ...d_p)$ y $d' = (d'_1, ...d'_p)$, se tiene por (1.54):

$$[d_j\mathcal{R}_j d'_j] \in \{0,1\}, \forall j = 1, ..., p$$

Por tanto,

$$[d\mathcal{R}d'] := g(\{[d_j\mathcal{R}_j d'_j], j = 1, ...p\}) = \wedge_{j=1...qp}[d_j\mathcal{R}_j d'_j] \in \{0,1\} \tag{1.55}$$

Es más

$$[d\mathcal{R}d'] = 1 \Longleftrightarrow [d_j\mathcal{R}_j d'_j], \forall j = 1, ..., p \tag{1.56}$$

■

Algunos ejemplos de relaciones $\mathcal{R}$ entre conjuntos de descripciones y de funciones g de combinación de niveles de relación pueden verse en la página 48, una vez definidos *eventos* en 1.4.2 y en la página 57, una vez definidas las *aserciones*

en 1.4.3.

1.4.2 Eventos

Los objetos simbólicos que toman en consideración una única variable son los eventos.

Definición 1.16 ***Objeto simbólico de tipo evento***. *Un objeto simbólico de tipo evento definido en E es una tupla $(a, \mathcal{R}, d)$ donde:*

- *a es una función, denotada por $a = [Y\mathcal{R}d]$, con Y una variable monoevaluada o simbólica con dominio $\mathcal{Y}$ definida por $Y : E \longrightarrow \mathcal{D}$, y $\mathcal{D}$ un conjunto de descripciones de elementos de E, asociado al conjunto $\mathcal{Y}$. La función a es :*

$$\begin{array}{rccl} a: & E & \longrightarrow & \mathcal{L} \\ & e & \longmapsto & a(e) = [Y(e)\mathcal{R}d] \end{array} \tag{1.57}$$

  que asocia a cada elemento de E el nivel de relación de su descripción en $\mathcal{D}$ (dada por Y) con la descripción d.

- *$\mathcal{R}$ es una relación de dominio definida en $\mathcal{D} \times \{d\}$ (véase definición 1.12).*

- *d es una descripción de un conjunto de descripciones asociado al conjunto $\mathcal{Y}$ (véase 1.3.5).*

Se llaman indistintamente la tupla $(a, \mathcal{R}, d)$ y $a = [Y\mathcal{R}d]$, evento.

$a(e) = [Y(e)\mathcal{R}d]$ es el ***nivel de relación del individuo*** **e** ***con un evento*** **a** *o nivel de relación de la descripción de e en $\mathcal{D}$ (dada por Y) con la descripción d.*

Un objeto simbólico es en realidad una tupla $(a, \mathcal{R}, d)$, con $a = [Y\mathcal{R}d]$, donde d es una descripción, $\mathcal{R}$ una relación entre descripciones y a es una función

definida de E en $\mathcal{L}$ que mide el nivel de relación de la descripción de un elemento $e \in E$ dada por Y con la descripción d según la relación $\mathcal{R}$. Es una función que permite obtener la extensión del objeto simbólico en el conjunto de individuos E. Los individuos que pertenecen a esta extensión son aquellos cuya descripción se relaciona con d (en el caso booleano) o tiene un nivel de relación *alto* con d.

Definición 1.17 ***Evento booleano****. Un objeto simbólico de tipo evento es booleano si el conjunto de comparación de descripciones es $\mathcal{L} = \{0, 1\}$ (en (1.57)). En este caso, dado $e \in E$,*

- *si $a(e) = [Y(e)\mathcal{R}d] = 1$, entonces la descripción de e en $\mathcal{D}$ (dada por Y) se relaciona con la descripción d, o e se relaciona con el evento a;*

- *y, si $a(e) = [Y(e)\mathcal{R}d] = 0$, entonces la descripción de e en $\mathcal{D}$ (dada por Y) no se relaciona con la descripción d, o e no se relaciona con el evento a.*

Definición 1.18 ***Extensión de un evento*** *en E. Sea $(a, \mathcal{R}, d)$ con $a = [Y\mathcal{R}d]$, un evento booleano definido en E. Se llama* ***extensión del evento booleano*** *a* ***en*** *E y se denota por $Ext_E(a)$, al subconjunto de elementos de E cuya descripción en $\mathcal{D}$ (dada por Y) se relaciona con el evento a:*

$$Ext_E(a) = \{e \in E / a(e) = [Y(e)\mathcal{R}d] = 1\} \tag{1.58}$$

Sea $(a, \mathcal{R}, d)$ con $a = [Y\mathcal{R}d]$, un evento definido en E y $\delta \in [0, 1]$. Se llama ***extensión de nivel*** *δ* ***del evento*** *a* ***en*** *E y se denota por $Ext_{E,\delta}(a)$, al subconjunto de elementos de E cuya descripción en $\mathcal{D}$ (dada por Y) tiene un nivel de relación con el evento a igual a superior a δ:*

$$Ext_{E,\delta}(a) = \{e \in E / a(e) = [Y(e)\mathcal{R}d] \geq \delta\} \tag{1.59}$$

Ejemplos de eventos según distintos tipos de variable Y, relación $\mathcal{R}$ y descripción d.

Sea $(a, \mathcal{R}, d)$, con $a = [Y\mathcal{R}d]$ un evento (véase definición 1.16). Sea Y una variable monoevaluada (véase definición 1.1) o simbólica (véanse definiciones 1.2 y 1.3) con dominio finito $\mathcal{Y} = \{z_1, ...z_x\}$ y d una descripción asociada al conjunto $\mathcal{Y}$ (véase 1.3.5). A continuación se presentan algunos tipos de relación $\mathcal{R}$ que se pueden establecer dependiendo del tipo de variable Y y de la descripción d.

Sea Y una variable monoevaluada definida por:

$$\begin{array}{rl} Y: & \Omega \longrightarrow \mathcal{Y} \\ & \omega \longmapsto z_\omega \end{array} \tag{1.60}$$

se pueden definir los siguientes tipos de eventos:

- Si $d = z \in \mathcal{Y}$, el evento es $(a, \mathcal{R}, z)$ con $a = [Y\mathcal{R}z]$. Ejemplos de relaciones booleanas $\mathcal{R}$ son la relación igualdad "=" y la relación desigualdad "$\neq$". Es decir, con la relación "=" por ejemplo, se tiene:

$$a(\omega) = [Y(\omega) = z] = [z_\omega = z] = \begin{cases} 1 \Longleftrightarrow z_\omega = z \\ 0 \Longleftrightarrow z_\omega \neq z \end{cases} \tag{1.61}$$

- Si $d = D \in \mathcal{P}(\mathcal{Y})$, el evento es $(a, \mathcal{R}, D)$ con $a = [Y\mathcal{R}D]$. Ejemplos de relaciones booleanas $\mathcal{R}$ son la pertenencia "$\in$", la no pertenencia "$\notin$", etc... Con la relación "$\in$" por ejemplo, se tiene:

$$a(\omega) = [Y(\omega) \in D] = [z_\omega \in D] = \begin{cases} 1 \Longleftrightarrow z_\omega \in D \\ 0 \Longleftrightarrow z_\omega \notin D \end{cases} \tag{1.62}$$

- Si $d = q \equiv (z_1 q(z_1), ..., z_x q(z_x)) \in \mathcal{M}^{\Pr ob}(\mathcal{Y})$, el evento es $(a, \mathcal{R}, q)$ con $a = [Y\mathcal{R}q]$. Ejemplos de relaciones probabilistas $\mathcal{R}$ son relaciones que se establecen entre una categoría de $\mathcal{Y}$ y una distribución de probabilidad en

el conjunto $\mathcal{Y}$. Esta relación puede ser:

$$a(\omega) = [Y(\omega) \sim q] = [z_\omega \sim (z_1 q(z_1), ..., z_x q(z_x))] = q(z_\omega) \tag{1.63}$$

que representa la probabilidad de la categoría z_ω, según la distribución de probabilidad q en $\mathcal{Y}$.

- Si $d = q \equiv (z_1 q(z_1), ..., z_x q(z_x)) \in \mathcal{M}^{Pos}(\mathcal{Y})$, el evento es $(a, \mathcal{R}, q)$ con $a = [Y\mathcal{R}q]$. Ejemplos de relaciones difusas $\mathcal{R}$ son relaciones que se establecen entre una categoría de $\mathcal{Y}$ y una distribución de posibilidad en el conjunto $\mathcal{Y}$. Esta relación puede ser:

$$a(\omega) = [Y(\omega) \sim q] = [z_\omega \sim (z_1 q(z_1), ..., z_x q(z_x))] = q(z_\omega) \tag{1.64}$$

que representa la posibilidad de la categoría z_ω, según la distribución de posibilidad q en $\mathcal{Y}$.

Sea $\underline{Y \text{ una variable multievaluada}}$ definida por:

$$\begin{aligned} Y: \quad & E \longrightarrow \mathcal{P}(\mathcal{Y}) \\ & e \longmapsto D_e \end{aligned} \tag{1.65}$$

Se pueden definir los siguientes tipos de eventos:

- Si $d = z \in \mathcal{Y}$, el evento es $(a, \mathcal{R}, z)$ con $a = [Y\mathcal{R}z]$. Ejemplo de relación booleana $\mathcal{R}$ es la relación de pertenencia contraria "$\ni$".

$$a(e) = [Y(e) \ni z] = [D_e \ni z] = \begin{cases} 1 \Longleftrightarrow z \in D_e \\ 0 \Longleftrightarrow z \notin D_e \end{cases} \tag{1.66}$$

- Si $d = D \in \mathcal{P}(\mathcal{Y})$, el evento es $(a, \mathcal{R}, D)$ con $a = [Y\mathcal{R}D]$. Ejemplos de relaciones booleanas $\mathcal{R}$ son el contenido "$\subseteq$" y el continente "$\supseteq$". Otra

relación que puede establecerse es si la intersección entre dos descripciones es el $\varnothing$ o no, etc...

Por ejemplo, con la relación contenido "$\subseteq$" se tiene:

$$a(e) = [Y(e) \subseteq D] = [D_e \subseteq D] = \begin{cases} 1 \Longleftrightarrow D_e \subseteq D \\ 0 \Longleftrightarrow D_e \nsubseteq D \end{cases} \tag{1.67}$$

Un ejemplo de relación difusa $\mathcal{R}$ puede ser:

$$a(e) = [Y(e) \sim D] = [D_e \sim D] = \frac{Card(D_e \cap D)}{Card(D_e \cup D)} \tag{1.68}$$

- Si $d = q \equiv (z_1 q(z_1), ..., z_x q(z_x)) \in \mathcal{M}^{\Pr ob}(\mathcal{Y})$, el evento es $(a, \mathcal{R}, q)$ con $a = [Y \mathcal{R} q]$. Ejemplos de relaciones $\mathcal{R}$ son relaciones que se establecen entre un subconjunto de categorías de $\mathcal{Y}$ y una distribución de probabilidad sobre el conjunto $\mathcal{Y}$. Esta relación probabilista puede ser:

$$a(e) = [Y(e) \sim q] = [D_e \sim (z_1 q(z_1), ..., z_x q(z_x))] = \textstyle\sum_{z_i \in D_e} q(z_i) \tag{1.69}$$

que representa la probabilidad del subconjunto $D_e \subseteq \mathcal{Y}$, dada la ley de probabilidad en $\mathcal{Y}$ expresada por $(z_1 q(z_1), ..., z_x q(z_x))$.

- Si $d = q \equiv (z_1 q(z_1), ..., z_x q(z_x)) \in \mathcal{M}^{Pos}(\mathcal{Y})$, el evento es $(a, \mathcal{R}, q)$ con $a = [Y \mathcal{R} q]$. Ejemplos de relaciones difusas $\mathcal{R}$ son relaciones que se establecen entre un subconjunto de categorías de $\mathcal{Y}$ y una distribución de posibilidad sobre el conjunto $\mathcal{Y}$. Esta relación difusa puede ser:

$$a(e) = [Y(e) \sim q] = [D_e \sim (z_1 q(z_1), ..., z_x q(z_x))] = \max_{z_i \in D_e} q(z_i) \tag{1.70}$$

que representa la posibilidad del subconjunto $D_e \subseteq \mathcal{Y}$, dada la distribución de posibilidad en $\mathcal{Y}$ expresada por $(z_1 q(z_1), ..., z_x q(z_x))$ (véase definición 1.5).

Sea Y una variable modal probabilista definida por:

$$\begin{array}{rlll} Y: & E & \longrightarrow \mathcal{M}^{\Pr ob}(\mathcal{Y}) \\ & e & \longmapsto q_e \equiv (z_1 q_e(z_1), ..., z_x q_e(z_x)) \end{array} \tag{1.71}$$

Se pueden definir los siguientes tipos de eventos relacionados con esta variable:

- Si $d = z \in \mathcal{Y}$, el evento es $(a, \mathcal{R}, z)$ con $a = [Y\mathcal{R}z]$. Ejemplo de relación probabilista es como la (1.63):

$$a(e) = [Y(e) \sim z] = [(z_1 q_e(z_1), ..., z_x q_e(z_x)) \sim z] = q_e(z) \tag{1.72}$$

  que representa la probabilidad de la categoría z para el individuo descrito por la distribución de probabilidad q_e.

- Si $d = D \in \mathcal{P}(\mathcal{Y})$, el evento es $(a, \mathcal{R}, D)$ con $a = [Y\mathcal{R}D]$. Ejemplo de relación probabilista es como la (1.69):

$$a(e) = [Y(e) \sim D] = [(z_1 q_e(z_1), ..., z_x q_e(z_x)) \sim D] = \textstyle\sum_{z_i \in D} q_e(z_i) \tag{1.73}$$

  que representa la probabilidad del subconjunto D, según ley de probabilidad q_e que describe el individuo e.

- Si $d = q \equiv (z_1 q(z_1), ..., z_x q(z_x)) \in \mathcal{M}^{\Pr ob}(\mathcal{Y})$, el evento es $(a, \mathcal{R}, q)$ con $a = [Y\mathcal{R}q]$. Ejemplos de relaciones probabilistas $\mathcal{R}$ son relaciones que se establecen entre dos distribuciones de probabilidad definidas sobre un mismo conjunto $\mathcal{Y}$. Se pueden definir diferentes tipos de relación entre q_e y q:

$$\begin{array}{rlll} \mathcal{R}: & \mathcal{M}^{\Pr ob}(\mathcal{Y}) \times \mathcal{M}^{\Pr ob}(\mathcal{Y}) & \longrightarrow \mathcal{L}(\subseteq [0,1]) \\ & (q_e, q) & \longmapsto [q_e \mathcal{R} q] := [q_e \sim q] \end{array} \tag{1.74}$$

Si la relación entre dos distribuciones es el *producto escalar*, entonces:

$$a(e) = [Y(e) \sim q] = [q_e \sim q] =< (q_e(z_1), ..., q_e(z_x)), (q(z_1), ..., q(z_x)) > \tag{1.75}$$

representa la probabilidad de que la categoría observada en dos experiencias aleatorias de distribuciones q y q_e sea la misma. Hay antecedentes de esta relación en Diday (1991, 1993a).

Otra relación que puede establecerse es una medida de similaridad entre distribuciones de probabilidad. Ésta puede derivarse de una medida de divergencia, como por ejemplo, la *medida de divergencia de Kullback-Leibler* (Kullback y Leibler, 1951). En este caso, es:

$$\begin{aligned} a(e) &= [Y(e) \sim q] = [q_e \sim q] = MAX_{KL} - d_{KL}(q, q_e) \\ &= MAX_{KL} - \sum_{i=1,...,x} q(z_i) \log \frac{q(z_i)}{q_e(z_i)} \end{aligned} \tag{1.76}$$

con MAX_{KL} el máximo de las divergencias posibles para dos distribuciones de probabilidad y $d_{KL}(.)$ la divergencia de Kullback-Leibler[6].

También a partir de la *divergencia de la* χ^2 se puede establecer una relación de dominio:

$$[q_e \sim q] = MAX_{\chi^2} - d_{\chi^2}(q, q_e) = MAX_{\chi^2} - \sum_{i=1,...,x} \frac{(q(z_i) - q_e(z_i))^2}{q(z_i)} \tag{1.77}$$

con MAX_{χ^2} el máximo de las divergencias posibles para dos distribuciones de probabilidad y $d_{\chi^2}(.)$ la divergencia de la χ^2.

[6] La medida de divergencia de Kullback-Leibler entre dos distribuciones q y q_e es no simétrica y mide la información media necesaria para discriminar a favor de una ley de probabilidad q, contra una ley de probabilidad q_e, supuesta q verdadera.

Ambas relaciones de dominio son no simétricas y, por tanto, son de tipo *matching*.

Algunos autores (Titterington et al., 1985, Gil et al., 1993, Gower, 19 y Bock, 2000b,), presentan medidas de similaridad, disimilaridad, distancias y divergencias entre distribuciones de probabilidad. A partir de una medida de disimilaridad, divergencia o distancia se deriva fácilmente una medida de similaridad que representa la relación de dominio correspondiente.

Sea Y una variable modal posibilista definida por:

$$\begin{aligned} Y: \quad & E \longrightarrow \mathcal{M}^{Pos}(\mathcal{Y}) \\ & e \longmapsto q_e \equiv (z_1 q_e(z_1), ..., z_x q_e(z_x)) \end{aligned} \tag{1.78}$$

Se pueden definir los siguientes tipos de eventos relacionados con esta variable:

- Si $d = z \in \mathcal{Y}$, el evento es $(a, \mathcal{R}, z)$ con $a = [Y\mathcal{R}z]$. Ejemplos de relación difusa es como (1.64):

$$a(e) = [Y(e) \sim z] = [(z_1 q_e(z_1), ..., z_x q_e(z_x)) \sim z] = q_e(z) \tag{1.79}$$

- Si $d = D \in \mathcal{P}(\mathcal{Y})$, el evento es $(a, \mathcal{R}, D)$ con $a = [Y\mathcal{R}D]$. Ejemplo de relación difusa es como (1.70):

$$a(e) = [Y(e) \sim D] = [(z_1 q_e(z_1), ..., z_x q_e(z_x)) \sim D] = \max_{z_i \in D} q_e(z_i) \tag{1.80}$$

- Si $d = q \equiv (z_1 q(z_1), ..., z_x q(z_x)) \in \mathcal{M}^{Pos}(\mathcal{Y})$, el evento es $(a, \mathcal{R}, q)$ con $a = [Y\mathcal{R}q]$. Ejemplos de relaciones difusas $\mathcal{R}$ son relaciones que se establecen entre dos distribuciones de posibilidad definidas sobre un mismo conjunto

$\mathcal{Y}$. Ejemplo de relación difusa es:

$$\begin{aligned} a(e) = [Y(e) \sim q] &= [(z_1 q_e(z_1), ..., z_x q_e(z_x)) \sim (z_1 q(z_1), ..., z_x q(z_x))] \\ &= \sup_{i=1,...,x}\{\min\{q_e(z_i), q(z_i)\}\} \end{aligned} \tag{1.81}$$

En este caso, el operador min puede ser sustituido por otra $T-$norma (véase definición 1.9, Diday, 1995c). Antecedentes a esta medida se encuentran en Zadeh (Zadeh, 1978) aplicada a las funciones de pertenencia de dos conjuntos difusos. Su aplicación a objetos simbólicos en Diday (1991, 1993b). Diday también introduce la relación difusa:

$$\begin{aligned} a(e) = [Y(e) \sim q] &= [(z_1 q_e(z_1), ..., z_x q_e(z_x)) \sim (z_1 q(z_1), ..., z_x q(z_x))] \\ &= \inf_{i=1,...,x}\{\max\{q_e(z_i), q(z_i)\}\} \end{aligned} \tag{1.82}$$

y llama a la aserción a, *de necesidad*. Esta medida es introducida también por Zadeh (Zadeh, 1978).

1.4.3 Aserciones

Una aserción es un objeto simbólico referido a varias variables. Se compone de varios eventos y está dotada de una función combinación de niveles de relación. Esta función combina los niveles de relación de cada uno de los eventos aplicados a un elemento del conjunto sobre el cual está definida la aserción.

Definición 1.19 ***Objeto simbólico de tipo aserción.*** *Un objeto simbólico de tipo aserción definido en E es una tupla $(a, \mathcal{R}, d)$ donde:*

- *a es una función, denotada por $a = [Y\mathcal{R}d]$, con $Y = (Y_1, ..., Y_p)$ un vector de variables monoevaluadas o simbólicas con dominios $\mathcal{Y}_1, ..., \mathcal{Y}_p$ definido*

por $Y : E \longrightarrow \mathcal{D}$, *y* $\mathcal{D} = \mathcal{D}_1 \times ... \times \mathcal{D}_p$ *un conjunto de descripciones de elementos de* E, $\mathcal{D}_j$ *asociado al dominio* $\mathcal{Y}_j$. *La aserción a se define como:*

$$\begin{aligned} a: \quad E &\longrightarrow \mathcal{L} \\ e &\longmapsto a(e) = [Y(e)\mathcal{R}d] = g(\{[Y_j(e)\mathcal{R}_j d_j], j = 1, ..., p\}) \\ &= \wedge_{j=1,...,p}[Y_j(e)\mathcal{R}_j d_j] \end{aligned} \tag{1.83}$$

que asocia a cada elemento de E *el nivel de relación de su descripción dada por* Y *con la descripción* $d = (d_1, ..., d_p)$, *según la relación producto* $\mathcal{R}$ *con la función de combinación de niveles de relación* g.

- $\mathcal{R} = \mathcal{R}_1 \times ... \times \mathcal{R}_p$ *es un producto de relaciones de dominio definido en* $\mathcal{D} \times \{d\}$ *como* $[d'\mathcal{R}d] = g(\{[d'_j\mathcal{R}_j d_j], j = 1, ..., p\})$ *para* $d' = (d'_1, ..., d'_p) \in \mathcal{D}$ *(véase definición 1.14).*

- $d = (d_1, ..., d_p)$ *es una descripción de un conjunto de descripciones asociado a los dominios* $\mathcal{Y}_1, ..., \mathcal{Y}_p$ *(véase 1.3.5).*

Se llama indistintamente a la tupla $(a, \mathcal{R}, d)$ *y a* $a = [Y\mathcal{R}d]$ *aserción. Así mismo, la aserción a se denota por* $a = \wedge_{j=1,...,p}[Y_j\mathcal{R}_j d_j]$, *si bien* $\wedge$ *no es la conjunción boolena necesariamente.*

$a(e) = [Y(e)\mathcal{R}d]$ *es el* ***nivel de relación del individuo*** **e** ***con la aserción*** **a** *o nivel de relación de la descripción de e en* $\mathcal{D}$ *dada por* Y *con la descripción* d.

Una **aserción de tipo individuo** es $\wedge_{j=1,...,p}[Y_j = z_j]$, con $Y_j : \Omega \longrightarrow \mathcal{Y}_j$ una variable monoevaluada, $z_j \in \mathcal{Y}_j$ y $\wedge$ el operador conjuntivo estándar.

Definición 1.20 ***Aserción booleana****. Un objeto simbólico de tipo aserción es booleano si el conjunto de comparación de descripciones es* $\mathcal{L} = \{0, 1\}$. *En este caso, dado* $e \in E$,

- *si* $a(e) = [Y(e)\mathcal{R}d] = 1$*, entonces la descripción de* e *en* $\mathcal{D}$ *(dada por el vector* Y*) se relaciona con la descripción* d*,* o e *se relaciona con la aserción* a*;*

- *y, si* $a(e) = [Y(e)\mathcal{R}d] = 0$*, entonces la descripción de* e *en* $\mathcal{D}$ *(dada por el vector* Y*) no se relaciona con la descripción* d*,* o e *no se relaciona con la aserción* a*.*

Definición 1.21 ***Extensión de una aserción*** *. Sea* $(a, \mathcal{R}, d)$ *con* $a = [Y\mathcal{R}d]$*, una aserción booleana definida en* E*. Se llama* ***extensión de la aserción booleana*** a *en* E*,* $Ext_E(a)$*, al subconjunto de elementos de* E *cuya descripción en* $\mathcal{D}$ *(dada por el vector* Y*) se relaciona con la aserción* a*:*

$$Ext_E(a) = \{e \in E / a(e) = [Y(e)\mathcal{R}d] = 1\} \tag{1.84}$$

Sea $(a, \mathcal{R}, d)$ *con* $a = [Y\mathcal{R}d]$*, una aserción definida en* E *y* $\delta \in [0, 1]$*. Se llama* ***extensión de nivel*** δ ***de la aserción*** a ***en*** E*,* $Ext_{E,\delta}(a)$*, al subconjunto de elementos de* E *cuya descripción en* $\mathcal{D}$ *(dada por el vector* Y*) tiene un nivel de relación con la aserción* a *igual a superior a* δ*:*

$$Ext_{E,\delta}(a) = \{e \in E / a(e) = [Y(e)\mathcal{R}d] \geq \delta\} \tag{1.85}$$

Se pueden definir las extensiones de aserciones booleanas sobre subconjuntos de E. Sea $S \in \mathcal{P}(E)$, la extensión de a en S es:

$$Ext_S(a) = \{e \in S / a(e) = [Y(e)\mathcal{R}d] = 1\} \tag{1.86}$$

y extensiones de nivel de aserciones sobre subconjuntos de E. Sea $\delta \in [0, 1]$, la extensión de nivel δ de a en S es:

$$Ext_{S,\delta}(a) = \{e \in S / a(e) = [Y(e)\mathcal{R}d] \geq \delta\} \tag{1.87}$$

Diday (Diday, 1991) distingue entre las aserciones probabilistas, posibilistas y de creencia según las descripciones simbólicas y las relaciones de dominio que definen las aserciones.

La importancia de la definición 1.21 refleja la importancia de los objetos simbólicos ya que éstos permiten la vuelta a bases de datos. Es decir, un objeto simbólico además de representar una intención, permite obtener la extensión de esta intención en una base de datos.

Definición 1.22 ***Equivalencia de aserciones.*** *Sean* $(a, \mathcal{R}, d)$ *y* $(b, \mathcal{R}', d')$ *con* $a = [Y\mathcal{R}d]$, $b = [Y'\mathcal{R}'d']$, *dos aserciones definidas en* E. *Las aserciones* a *y* b *son equivalentes, y se denota por* $a \equiv b$ *si:*

$$a(\omega) = b(\omega), \forall \omega \in E \tag{1.88}$$

Ejemplos de aserciones según tipos de eventos y funciones de combinación de niveles de relación.

Ejemplo 1.3 ***Aserciones booleanas.*** *A partir de la matriz de datos simbólicos (1.10) del ejemplo 1.1 se definen sobre* Ω *las aserciones siguientes:*

$$\begin{aligned} a_1 &= [\widetilde{sexo} \in \{mujer\}] \wedge [\widetilde{profesión} \in \{informática, secretaria\}] \\ a_2 &= [\widetilde{sexo} \in \{varón\}] \wedge [\widetilde{profesión} \in \{informática, administrativa\}] \end{aligned}$$

con las variables $\widetilde{sexo}$ *y* $\widetilde{profesión}$ *definidas en* Ω *(véase ejemplo 1.1) y* $\wedge$ *el operador lógico conjuntivo.*

La aserción a_1 *es de la siguiente forma:*

$$\begin{aligned} a_1 : \ & \Omega \longrightarrow \{0, 1\} \\ & \omega \longmapsto a_1(\omega) = [\widetilde{sexo}(\omega) \in \{mujer\}] \\ & \qquad \wedge [\widetilde{profesión}(\omega) \in \{informática, secretaria\}] \end{aligned} \tag{1.89}$$

Se tiene que

$$a_1(\omega) = 1 \Longleftrightarrow \widetilde{sexo}(\omega) = mujer \ y \ \widetilde{profesión}(\omega) \in \{informática, secretaria\}$$

La aserción a_2 se define de forma similar.

Las aserciones a_1 y a_2 representan la intención de dos subconjuntos de individuos, S_1 y S_2, respectivamente (en (1.9)). Cada una de las aserciones a_1 y a_2 son también un medio de obtención de individuos que verifican dicha intención. Las extensiones en Ω de las aserciones a_1 y a_2 son:

$$\begin{aligned} Ext_\Omega(a_1) &= \{\omega \in \Omega / a_1(\omega) = 1\} = \{\omega_1, \omega_2, \omega_3\} \\ Ext_\Omega(a_2) &= \{\omega \in \Omega / a_2(\omega) = 1\} = \{\omega_4, \omega_5, \omega_6, \omega_7\} \end{aligned}$$

La expresión de las aserciones a_1 y a_2 se puede simplificar sin más que considerar equivalentes $[\widetilde{sexo} \in \{mujer\}]$ con $[\widetilde{sexo} = mujer]$ y $[\widetilde{sexo} \in \{varón\}]$ con $[\widetilde{sexo} = varón]$ ya que respectivamente dan los mismos valores al aplicarlos sobre los individuos de Ω, por ser la variable $\widetilde{sexo}$ monoevaluada.

Ejemplo 1.4 ***Aserciones probabilistas.*** *De forma similar al ejemplo anterior, a partir de la matriz de datos simbólicos (1.26) en el ejemplo 1.2 se definen sobre Ω las aserciones siguientes:*

$$\begin{aligned} a_3 &= [\widetilde{sexo} \sim (mujer1)] \wedge [\widetilde{profesión} \sim (informática\frac{1}{3}, secretaria\frac{2}{3})] \\ a_4 &= [\widetilde{sexo} \sim (varón1)] \wedge [\widetilde{profesión} \sim (informática\frac{1}{2}, administrativa\frac{1}{2})] \end{aligned}$$

con las variables $\widetilde{sexo}$ y $\widetilde{profesión}$ definidas en Ω (véase ejemplo 1.1) y $\wedge$ el operador producto. La relación de dominio $\sim$ definida en $\mathcal{Y} \times \mathcal{M}^{Prob}(\mathcal{Y})$ es la definida en (1.63).

La aserción a_3 es de la siguiente forma:

$$\begin{array}{rl} a_3: & \Omega \longrightarrow \{0,1\} \\ & \omega \longmapsto a_3(\omega) \quad = [\widetilde{sexo}(\omega) \sim (mujer1)] \\ & \qquad\qquad\quad \wedge[\widetilde{profesión}(\omega) \sim (inform\acute{a}tica\tfrac{1}{3}, secretaria\tfrac{2}{3})] \end{array} \tag{1.90}$$

La aserción a_4 se define de forma similar.

En este caso las extensiones en Ω de las aserciones a_3 y a_4 dependen de un umbral de adecuación ya que las relaciones de dominio $\sim$ no son booleanas. Los niveles de relación de los individuos $\omega \in \Omega$ de la matriz de individuos (1.8) con a_3 y a_4 son:

$$\begin{aligned} a_3(\omega_1) &= \frac{1}{3}; a_3(\omega_2) = \frac{2}{3}; a_3(\omega_3) = \frac{2}{3}; a_3(\omega_4) = 0; a_3(\omega_5) = 0; a_3(\omega_6) = 0; a_3(\omega_7) = 0; \\ a_4(\omega_1) &= 0; a_4(\omega_2) = 0; a_4(\omega_3) = 0; a_4(\omega_4) = \frac{1}{2}; a_4(\omega_5) = \frac{1}{2}; a_4(\omega_6) = \frac{1}{2}; a_4(\omega_7) = \frac{1}{2}; \end{aligned}$$

Así mismo, dado un umbral $\delta \in [0,1]$ se obtienen las extensiones de nivel δ en Ω de las aserciones a_3 y a_4. Varios ejemplos son:

$$\begin{aligned} Ext_{0.33,\Omega}(a_3) &= \{\omega \in \Omega / a_3(\omega) \geq 0.33\} = \{\omega_1, \omega_2, \omega_3\} \\ Ext_{0.5,\Omega}(a_3) &= \{\omega \in \Omega / a_3(\omega) \geq 0.5\} = \{\omega_2, \omega_3\} \\ Ext_{0.5,\Omega}(a_4) &= \{\omega \in \Omega / a_4(\omega) \geq 0.5\} = \{\omega_4, \omega_5, \omega_6, \omega_7\} \\ Ext_{0.66,\Omega}(a_4) &= \{\omega \in \Omega / a_4(\omega) \geq 0.66\} = \emptyset \end{aligned}$$

La expresión de las aserciones a_3 y a_4 se puede simplificar sin más que considerar equivalentes $[\widetilde{sexo} \sim (1mujer)]$ con $[\widetilde{sexo} = mujer]$ y $[\widetilde{sexo} \sim (1var\acute{o}n)]$ con $[\widetilde{sexo} = var\acute{o}n]$ ya que respectivamente dan los mismos valores al aplicarlos sobre los individuos de Ω, por ser la variable $\widetilde{sexo}$ monoevaluada.

A continuación se muestran otros dos ejemplos de aserción probabilista y aserción posibilista.

Sea la aserción probabilista $(a, \mathcal{R}, q)$ definida en E, con $a = \wedge_{j=1,...,p}[Y_j \sim q_j]$, $q = (q_1, ..., q_p) \in \mathcal{M}^{\Pr ob}(\mathcal{Y})$ y el vector $Y = (Y_1, ..., Y_p)$ de variables modales probabilistas:

$$\begin{array}{rl} Y: & E \longrightarrow \mathcal{M}^{\Pr ob}(\mathcal{Y}) \\ & e \longmapsto q_e = (q_{e,1}, ..., q_{e,p}) \end{array} \tag{1.91}$$

con $q_{e,j} \in \mathcal{M}^{\Pr ob}(\mathcal{Y}_j)$, $\mathcal{Y}_j = \{z_1, ..., z_{l_j}\}$, $j \in \{1, ..., p\}$. La aserción a es:

$$\begin{array}{rl} a: & E \longrightarrow [0,1] \\ & e \longmapsto a(e) = [Y(e) \sim q] = \wedge_{j=1,...,p}[Y_j(e) \sim q_j] = \wedge_{j=1,...,p}[q_{e,j} \sim q_j] \end{array} \tag{1.92}$$

Tomando como relación de dominio $\sim$, el producto escalar entre dos distribuciones (véase (1.75)), dependiendo de la función $g(.)$ de combinación de niveles de relación, la aserción a puede ser:

$$a(e) = \frac{1}{p} \sum_{j=1,...,p} \sum_{l=1,...,l_j} q_{e,j}(z_l) q_j(z_l) \tag{1.93}$$

$$\text{o bien:} \qquad a(e) = \prod_{j=1,...,p} \sum_{l=1,...,l_j} q_{e,j}(z_l) q_j(z_l) \tag{1.94}$$

(véase Diday (1991, 1995b)).

Sea la aserción posibilista $(a, \mathcal{R}, q)$ definida en E, con $a = \wedge_{j=1,...,p}[Y_j \sim q_j]$, $q = (q_1, ..., q_p) \in \mathcal{M}^{Pos}(\mathcal{Y})$ y sea el vector $Y = (Y_1, ..., Y_p)$ de variables modales posibilistas:

$$\begin{array}{rl} Y: & E \longrightarrow \mathcal{M}^{Pos}(\mathcal{Y}) \\ & e \longmapsto q_e = (q_{e,1}, ..., q_{e,p}) \end{array} \tag{1.95}$$

con $q_{e,j} \in \mathcal{M}^{Pos}(\mathcal{Y}_j)$, $\mathcal{Y}_j = \{z_1, ..., z_{l_j}\}$, $j \in \{1, ..., p\}$. La aserción a es:

$$\begin{array}{rlll} a: & E & \longrightarrow [0,1] & \\ & e & \longmapsto a(e) = [Y(e) \sim q] = \wedge_{j=1,...,p}[Y_j(e) \sim q_j] = \wedge_{j=1,...,p}[q_{e,j} \sim q_j] & (1.96) \end{array}$$

Tomando como relación de dominio $\sim$ la expresada en (1.81) y en (1.82), respectivamente, esta aserción puede ser:

$$a(e) = \max_{j=1,...,p} (\sup_{l=1,...,l_j} (\min(q_{e,j}(z_l), q_j(z_l)))) \quad (1.97)$$

$$\text{ó} \quad a(e) = \min_{j=1,...,p} (\inf_{l=1,...,l_j} (\max(q_{e,j}(z_l), q_j(z_l)))) \quad (1.98)$$

Se encuentran antecedentes de ambas en Zadeh (Zadeh, 1978) y su aplicación a objetos simbólicos en Diday ((1991, 1995a)). En el primer caso, deben redefinirse las propiedades (1.51) y (1.52).

Sea la aserción posibilista $(a, \mathcal{R}, D)$ definida en E, con $a = \wedge_{j=1,...,p}[Y_j \sim D_j]$, $D = (D_1, ..., D_p) \in \mathcal{P}(\mathcal{Y})$ y sea el vector $Y = (Y_1, ..., Y_p)$ de variables modales posibilistas (1.95). Tomando como relación de dominio $\sim$ la expresada en (1.80), la aserción a definida en E (véase Diday (1991, 1995a)) puede ser:

$$a(e) = \max_{j=1,...,p} (\max_{z_l \in D_j} (q_{e,j}(z_l))) \quad (1.99)$$

En (1.97) y (1.99), la función de combinación de niveles de relación es el operador max que puede sustituirse por otra $T-$conorma y deben redefinirse las propiedades (1.51) y (1.52). En (**??**), el operador min puede sustituirse por una $T-$norma.

1.4.4 Otros tipos de datos y objetos simbólicos

Existen otros tipos de variables, datos y objetos simbólicos que exceden el ámbito de esta Memoria:

- Las variables y datos simbólicos que se refieren a variables con dominio en un continuo. Este es el caso de las variables *de intervalo*. Por ejemplo, un *dato de intervalo* puede estar representado por el intervalo $[156, 170]$ para la variable simbólica de intervalo *edad*.

- Las variables y datos simbólicos *de creencia* y las correspondientes aserciones definidas con estas variables y/o descripciones. Esta representación del conocimiento se puede consultar en Shafer, 1976 que extiende, en su *teoría de la evidencia,* los conceptos introducidos por Dempster. Estos tipos de objetos simbólicos han sido introducidos por Diday (Diday, (1995a, 1995b)).

- Las variables y objetos simbólicos asociados a una generalización de modos que expresen *grados de certeza* sobre las categorías de una variable (Diday, (1991, 1995b)). En este caso, el conjunto de modos es un conjunto ordenado. Por ejemplo, este conjunto puede ser {siempre, a menudo, a veces, rara vez, nunca}.

- *Objetos simbólicos modales exteriores* (Diday (1990, 1991)). Para ilustrarlos se incluye un ejemplo. Sea $ME = \{M_1 =$ Necesario, $M_2 =$ Posible, $M_3 =$ Imposible$\}$ un conjunto de modos definidos sobre los eventos de una aserción y sea $me = \{m_1 =$ Verdad, $m_2 =$ Verdad condicionada, $m_3 =$ Falso$\}$ un conjunto de modos definidos sobre los eventos de una aserción de tipo individuo. Sean a una aserción y e una aserción de tipo individuo modales

exteriores definidas como:

$$a = M_1[diploma \in \{D_1, D_2\}] \wedge M_3[nacion = extranjero] \quad (1.100a)$$

$$e = m_1[diploma = D_2] \wedge m_3[nacion = no_extranjero] \quad (1.100b)$$

Para obtener el nivel de relación del objeto simbólico a y un individuo e, además de las relaciones de dominio que se establecen entre las descripciones de los eventos de la aserción y las correspondientes descripciones del individuo (véase definición 1.19), ha de establecerse una aplicación g_m de adecuación entre los modos de los conjuntos ME y me y las condiciones que deben verificarse para ser aplicada. Esta aplicación en general no se aplicará (será nula) si el nivel de relación del evento de la aserción y la descripción correspondiente del individuo es nula.

La imagen de la aplicación $g_m(.)$ de comparación de los modos de los conjuntos ME y me puede ser $\mathcal{L} = [0, 1]$ o un conjunto de modos (por ejemplo, $\mathcal{L} = \{L_1 = \text{Conviene}, L_2 = \text{Puede_convenir}, L_3 = \text{No_conviene}\}$). Por extensión (de nivel de relación en (1.57)), el resultado de esta comparación es el *nivel de relación* de los eventos modales exteriores con las descripciones de los individuos en las variables correspondientes, dotadas además de sus modos respectivos.

Posteriormente, se aplica una aplicación $g(.)$ de combinación de niveles de relación. De este modo, para a y e de (1.100a) y (1.100b), el nivel de relación puede ser:

$$a(e) = g(g_m(M_1, m_1), 0) = g(g_m(\text{Necesario}, \text{Verdad}), 0) = g(\text{Conviene}, 0) = 0 \quad (1.101)$$

- *Objetos simbólicos síntesis* (Diday, 1991). Un objeto de síntesis es una conjunción de varios *objetos simbólicos horda*. Un objeto horda se define

sobre una potencia del conjunto de individuos. Por ejemplo el objeto horda:

$$[Y_1(u_1) = 1] \wedge [Y_2(u_2) = 2] \tag{1.102}$$

afecta a pares de elementos del conjunto de individuos. La extensión de la misma se compone de pares de elementos (ω_1, ω_2) tales que $Y_1(\omega_1) = 1$ e $Y_2(\omega_2) = 2$.

- *Objetos simbólicos regla* (Diday, 1991, Bock y Diday, 2000b). Por ejemplo:

$$\text{Si } [semanas < 42] \wedge [sexo = mujer] \Longrightarrow [peso < 2.5]] \tag{1.103}$$

- *Objetos simbólicos de tipo disyunción* (Bock y Diday, 2000b). Por ejemplo:

$$[color \in \{naranja, rojo\}] \vee [hojas = redondas] \tag{1.104}$$

- Objetos simbólicos resultantes de la aplicación de una *aplicación de filtro* a un vector de variables simbólicas y las correspondientes descripciones simbólicas (Bock y Diday, 2000b). Se utilizan en el capítulo 3 para caracterizar la representación de los nodos del árbol en el conjunto de predictores (nota 3.2).

Los datos y objetos simbólicos pueden representar información adicional acerca de los datos, es decir, metadatos. Los metadatos que se consideran en el Análisis de Datos Simbólicos son:

- Las variables taxonómicas representan *taxonomías* o estructuras jerárquicas entre categorías de una variable. Por ejemplo, la variable *Alimentos* de cinco categorías es *verdura* si es *acelgas*, *espinacas* o *judías*; y es *legumbres*, si es *garbanzos* y *lentejas*.

- Las *dependencias jerárquicas* entre variables representan variables que no son aplicables para determinados valores de otra variable. Por ejemplo, si la variable $fumador$ es igual a no, entonces la variable $marca_de_cigarrillos$ es *no aplicable*. La forma de representar este metadato es mediante el objeto simbólico:

$$\text{Si } [fumador = no] \Longrightarrow [marca_cigarrillos = No_aplicable] \quad (1.105)$$

  Este tipo de objeto simbólico recibe el nombre de *regla de no aplicabilidad*.

- Las *dependencias lógicas* entre variables representan valores posibles de una variable en función de los valores de otra. Por ejemplo, si la variable *animal* es *ratón*, entonces la variable *longitud* es menor o igual que 20 centímetros. Este metadato se puede representar por el objeto simbólico:

$$\text{Si } [animal = rata] \Longrightarrow [longitud \leq 25] \quad (1.106)$$

1.4.5 Generalización

El proceso de generalización de un conjunto de individuos descritos por datos monoevaluados consiste en la obtención de datos y objetos simbólicos que los describan agregadamente.

Antecedentes de generalización pueden encontrarse en Michalski, 1973 inspirados en la lógica de primer orden que aplica técnicas de Inteligencia Artificial mediante la búsqueda heurística de *complejos* (Michalski, 1969, Michalski y Larson, 1983, Michalski et al., 1986, Clark y Nibblet, 1989), dada una clasificación inicial $\{c_1, ..., c_s\}$. Un complejo no es más que un predicado lógico definido en los predictores con los operadores conjuntivo y disyuntivo. Clark y Nibblet combinan esta búsqueda heurística con la *entropía de Shannon* para evaluar la calidad de los complejos y con el *estadístico de la razón de verosimilitud* (Kalbfleish, 1979)

para evaluar su significatividad. Este estadístico es:

$$2 \sum_{i=1,\ldots,s} n_i^k \log(\frac{n_i^k}{e_i^k}) \tag{1.107}$$

con: n_i^k frecuencia de elementos que cumplen el complejo k y son de la clase c_i y e_i^k frecuencia esperada bajo la hipótesis de distribución aleatoria de los elementos que cumplen el complejo. En determinadas circunstancias, este estadístico se distribuye aproximadamente como un estadístico χ^2 con $s-1$ grados de libertad.

La generalización por datos y objetos simbólicos puede aplicarse a consultas de una base de datos (Stéphan et al., 2000), es decir, a los conjuntos de individuos resultantes de las consultas, a clases obtenidas por una técnica de Análisis de Conglomerados (Gettler-Summa et al., 1994, Goupil et al., 2000) o a cualquier otra clasificación.

En 1.3.2 y 1.3.3 se ha presentado la descripción agregada de clases de individuos a partir de las descripciones monoevaluadas de los individuos de la clase (pag. 21 y 27 y ejemplos 1.3 y 1.4), introducida por Stéphan et al., (Stéphan, 1996, Stéphan et al., 2000). Proponen para cada una de las clases, realizar un proceso de *generalización* aplicado a cada variable para obtener descripciones simbólicas multievaluadas o modales probabilistas y objetos simbólicos de un conjunto o clase de individuos, seguido de un proceso de *especificación* que evite una sobregeneralización. Este proceso de especificación, se realiza de forma univariante mediante la calidad de la descripción obtenida. La medida de calidad que proponen combina la homogeneidad de los individuos de la clase con la extensión del objeto simbólico que se obtiene en la fase de generalización.

Gettler-Summa et al. (Gettler-Summa et al., 1994) proponen una generalización de conglomerados de individuos con variables originales categóricas, por objetos multievaluados y modales probabilistas descritos por las probabilidades empíricas. Realizan la generalización de cada conglomerado por un proceso iterativo añadiendo sucesivamente eventos con mayor poder generalizante y dis-

criminante frente a los demás conglomerados. Comparan los resultados de la clasificación original con las extensiones a un determinado umbral de los objetos simbólicos obtenidos. La determinación del umbral la realizan según el poder generalizante y discriminante de estos objetos. Con esta misma idea, Gettler-Summa (Gettler-Summa, 1999) propone un proceso de *marcaje*, en un proceso iterativo que generaliza todas las variables originales. Este proceso comienza por eventos booleanos de descripciones monoevaluadas, combina criterios de homogeneidad de cada clase con criterios de discriminación con respecto a las demás clases y obtiene objetos simbólicos multievaluados y modales probabilistas.

El proceso de generalización puede aplicarse también a la obtención de objetos simbólicos que ayuden a la interpretación de ejes factoriales obtenidos en una técnica de Análisis de Datos a individuos de datos monoevaluados (Gettler-Summa, 1992, Chavent, 1996). Se dividen los ejes en tres clases según las proyecciones de los individuos. Se procede a la descripción simbólica de las tres clases por generalización (Chavent, 1996) o por aplicación de una técnica de Segmentación (Gettler-Summa, 1996).

Ferraris et al. (Ferraris et al., 1995) proponen un método de obtención de objetos simbólicos a partir de series cronológicas.

1.5 Operaciones sobre conjuntos de aserciones

Esta sección introduce algunas operaciones sobre conjuntos de aserciones necesarias en el capítulo 4. En 1.5.1 se definen la unión, intersección y complementariedad de aserciones, así como la aserción total y la aserción nula y en 1.5.2 se define la conjunción de aserciones. En 1.5.1, se desciende al caso particular de conjuntos de aserciones que se utilizan en esta Memoria.

1.5.1 Unión, intersección y complementariedad

Se definen estas operaciones para los conjuntos de aserciones que se introducen en el capítulo 4. Sea

$$\mathcal{A}' = \{(a, \mathcal{R}, d) | a = [Y \mathcal{R} d], a : \Omega \rightarrow [0,1], \text{ con } d \in \mathcal{D}\} \tag{1.108}$$

un conjunto de aserciones definidas en el conjunto Ω con $Y : \Omega \longrightarrow \mathcal{D}'$ una variable o vector de variables monoevaluadas o modales probabilistas, con $\mathcal{D}'$ y $\mathcal{D}$ dos conjuntos de descripciones (univariantes o multivariantes) relativos a un dominio $\mathcal{Y}$. Se asume que en $\mathcal{D}$ están definidas las operaciones de unión, intersección y complementariedad. En particular, se considera en esta sección que $\mathcal{D} = \mathcal{Y}$ o $\mathcal{D} = \mathcal{P}(\mathcal{Y})$ y que la unión, intersección y complementariedad definidas son las conjuntistas en $\mathcal{D}_i$ (para $\mathcal{D} = \mathcal{D}_1 \times ... \times \mathcal{D}_p$) o en $\mathcal{D}$ univariante[7]. Así mismo, se asumen las propiedades (1.45) y (1.46) para la relación (o relación producto) $\mathcal{R}$ y las propiedades de (1.49) a (1.52) para la función $g(.)$ de combinación de niveles de relación.

A continuación se definen la unión, intersección y complementariedad de aserciones[8]. Por simplificación en la notación, se denotan la unión, intersección y

[7] Por ejemplo sean $\mathcal{D}_1 = \mathcal{P}(\{l_1, ..., l_3\})$, $\mathcal{D}_2 = \mathcal{P}(\{l'_1, ..., l'_5\})$, y sean las descripciones $d, e \in \mathcal{D} = \mathcal{D}_1 \times \mathcal{D}_2, d = (\{l_1, l_2\}, \{l'_1, l'_3\}), e = (\{l_1, l_3\}, \{l'_2, l'_4\})$, entonces:

$$\begin{aligned} d \cup e &= (\{l_1, l_2, l_3\}, \{l'_1, l'_2, l'_3, l'_4\}) \\ d \cap e &= (\{l_1\}, \emptyset) \\ d^c &= (\{l_3\}, \{l'_2, l'_4, l'_5\}) \end{aligned}$$

[8] Diday (Diday, 1991) define la unión, intersección y complementariedad de otros tipos de aserciones modales probabilistas tales como:

$$a = [Y \sim q]$$

con Y un vector de variables monoevaluadas y $q \in \mathcal{M}^{\Pr ob}(\mathcal{Y})$.

Sea $\mathcal{M}^{Cr}(\mathcal{Y})$ el conjunto de descripciones modales de creencia (véase Diday, 1991). Así mismo, Diday (Diday, 1991) define las operaciones de unión, intersección y complementario para aserciones de posibilidad y aserciones de creencia, definiendo previamente estas operaciones para elementos de los conjuntos $\mathcal{M}^{\Pr ob}(\mathcal{Y})$, $\mathcal{M}^{Pos}(\mathcal{Y})$ y $\mathcal{M}^{Cr}(\mathcal{Y})$, respectivamente. Estas

complementariedad de la misma manera en el conjunto $\mathcal{D}$ y en el conjunto $\mathcal{A}'$.

Definición 1.23 ***Unión de dos aserciones.*** *Sean $a_1 = (Y\mathcal{R}d) \in \mathcal{A}', a_2 = (Y\mathcal{R}d') \in \mathcal{A}'$, dos aserciones de $\mathcal{A}'$. Se define la unión de las aserciones a_1 y a_2 como la aserción:*

$$a_1 \cup a_2 = (Y\mathcal{R}(d \cup d')) \tag{1.109}$$

*La **unión** de a_1 y a_2 es la **aserción total** si $d \cup d' = \mathcal{Y}$. La aserción total se denota por $t^{\mathcal{A}'}$.*

La aserción total verifica que:

- $\forall \omega \in \Omega, t^{\mathcal{A}'}(\omega) = 1$, por (1.46) y (1.51).
- $\forall S \subseteq \Omega, Ext_S(t^{\mathcal{A}'}) = S$

Se deduce de (1.46) y de la condición (1.51), la siguiente proposición.

Proposición 1.2 *Si uno de los eventos que componen una aserción es el evento total, entonces esta aserción es equivalente a la aserción que se compone de todos los eventos excluidos el evento total.*

Definición 1.24 ***Intersección de dos aserciones.*** *Sean $a_1 = (Y\mathcal{R}d) \in \mathcal{A}'$, $a_2 = (Y\mathcal{R}d') \in \mathcal{A}'$, dos aserciones de $\mathcal{A}'$. Se define la intersección de las aserciones a_1 y a_2 como la aserción:*

$$a_1 \cap a_2 = (Y\mathcal{R}(d \cap d'))$$

*La **intersección** de a_1 y a_2 es la **aserción vacía** si $d \cap d' = \emptyset^p = (\emptyset, ...^p ..., \emptyset)$. La aserción vacía se denota por $\emptyset^{\mathcal{A}'}$.*

definiciones en $\mathcal{M}^{\Pr ob}(\mathcal{Y})$, $\mathcal{M}^{Pos}(\mathcal{Y})$ y $\mathcal{M}^{Cr}(\mathcal{Y})$, no siempre proporcionan elementos en estos conjuntos.

Hay antecedentes de la aserción total y la aserción vacía en Diday, 1995b. La aserción vacía verifica que:

- $\forall \omega \in \Omega, \emptyset^{\mathcal{A}'}(\omega) = 0$, por (1.45) y (1.52).
- $\forall S \subseteq \Omega, Ext_S(\emptyset^{\mathcal{A}'}) = \emptyset$

Por (1.45) y la condición (1.52) se deduce la siguiente proposición.

Proposición 1.3 *Si uno de los eventos que componen la aserción es el evento vacío, la aserción es equivalente a la aserción vacía.*

Definición 1.25 ***Complementario de una aserción.*** *Sea $a = (Y\mathcal{R}d) \in \mathcal{A}'$ una aserción de $\mathcal{A}'$. Se define la aserción complementaria de la aserción a como la aserción:*

$$a^c = (Y\mathcal{R}d^c)$$

Proposición 1.4 *Sea la aserción $a = (Y\mathcal{R}d)$ se tiene que:*

$$\begin{aligned} a \cup a^c &= t^{\mathcal{A}'} \\ a \cap a^c &= \emptyset^{\mathcal{A}'} \end{aligned}$$

Demostración. Se comprueba fácilmente que:

$$\begin{aligned} a \cup a^c &= (Y\mathcal{R}(d \cup d^c)) = (Y\mathcal{R}\mathcal{Y}) = t^{\mathcal{A}'} \\ a \cap a^c &= (Y\mathcal{R}(d \cap d^c)) = (Y\mathcal{R}\emptyset) = \emptyset^{\mathcal{A}'} \end{aligned}$$

■

La proposición siguiente demuestra, bajo determinadas circunstancias, la relación entre la unión, intersección y el complementario de aserciones con la extensión de las aserciones resultantes de dichas operaciones, respectivamente.

Proposición 1.5 *Si $\mathcal{R}$ es una relación de dominio definida en $\mathcal{D}' \times \mathcal{D}$, que verifica $\forall d' \in \mathcal{D}', \forall d_1, d_2 \in \mathcal{D}$,*

$$d'\mathcal{R}(d_1 \cup d_2) = \max\{d'\mathcal{R}d_1, d'\mathcal{R}d_2\} \tag{1.110a}$$

$$d'\mathcal{R}(d_1 \cap d_2) = \min\{d'\mathcal{R}d_1, d'\mathcal{R}d_2\} \tag{1.110b}$$

$$d'\mathcal{R}d^c = 1 - d'\mathcal{R}d \tag{1.110c}$$

entonces se cumple para $a_1 = [Y\mathcal{R}d_1], a_2 = [Y\mathcal{R}d_2], a = [Y\mathcal{R}d] \in \mathcal{A}'$:

$$Ext_\Omega(a_1 \cup a_2) = Ext_\Omega(a_1) \cup Ext_\Omega(a_2) \tag{1.111a}$$

$$Ext_\Omega(a_1 \cap a_2) = Ext_\Omega(a_1) \cap Ext_\Omega(a_2) \tag{1.111b}$$

$$Ext_\Omega(a^c) = \Omega - Ext_\Omega(a) \tag{1.111c}$$

Demostración.

$$\omega \in Ext_\Omega(a_1 \cup a_2) = Ext_\Omega([Y\mathcal{R}d_1 \cup d_2]) \Longleftrightarrow [Y(\omega)\mathcal{R}d_1 \cup d_2] = 1 \Longleftrightarrow$$
$$\max\{[Y(\omega)\mathcal{R}d_1], [Y(\omega)\mathcal{R}d_2]\} = 1 \Longleftrightarrow$$
$$\omega \in Ext_\Omega([Y\mathcal{R}d_1]) \cup Ext_\Omega([Y\mathcal{R}d_2]) = Ext_\Omega(a_1) \cup Ext_\Omega(a_2) \tag{1.112}$$

$$\omega \in Ext_\Omega(a_1 \cap a_2) = Ext_\Omega([Y\mathcal{R}d_1 \cap d_2]) \Longleftrightarrow [Y(\omega)\mathcal{R}d_1 \cap d_2] = 1 \Longleftrightarrow$$
$$\min\{[Y(\omega)\mathcal{R}d_1], [Y(\omega)\mathcal{R}d_2]\} = 1 \Longleftrightarrow [Y(\omega)\mathcal{R}d_1] = [Y(\omega)\mathcal{R}d_2] = 1$$
$$\omega \in Ext_\Omega([Y\mathcal{R}d_1]) \cap Ext_\Omega([Y\mathcal{R}d_2]) = Ext_\Omega(a_1) \cap Ext_\Omega(a_2) \tag{1.113}$$

$$\omega \in Ext_\Omega(a^c) = Ext_\Omega([Y\mathcal{R}d^c]) \Longleftrightarrow [Y(\omega)\mathcal{R}d^c] = 1 \Longleftrightarrow$$
$$[Y(\omega)\mathcal{R}d] = 0 \Longleftrightarrow \omega \in \Omega - Ext_\Omega([Y\mathcal{R}d]) \Longleftrightarrow \omega \in \Omega - Ext_\Omega(a) \tag{1.114}$$

■

Se comprueba fácilmente que (1.111a a 1.111c) se cumplen para extensiones en subconjuntos de Ω.

Esta proposición se verifica en el caso particular de que los conjuntos de descripciones sean $\mathcal{D}'=\mathcal{Y}$ y $\mathcal{D}=\mathcal{P}(\mathcal{Y})$ o $\mathcal{D}=\mathcal{Y}$, la relación de dominio $\mathcal{R}$ sea la de pertenencia ($\in$) y la función $g(.)$ de combinación de niveles de relación sea la conjunción lógica.

1.5.2 Conjunción

A continuación, se define la conjunción de aserciones distinguiéndose la conjunción de aserciones definidas sobre distintos vectores de variables de la conjunción de aserciones definidas sobre el mismo vector de variables.

Definición 1.26 ***Conjunción de aserciones definidas sobre distintos vectores de variables***[9]**.** *Sean $\mathcal{A}_1$ y $\mathcal{A}_2$ dos conjuntos de objetos simbólicos de tipo aserción definidos sobre el mismo conjunto Ω y asociados a los respectivos vectores de variables monoevaluadas o simbólicas Y_1, Y_2 definidos sobre Ω, tales que los vectores de variables Y_1, Y_2 no comparten ninguna variable. Y sea $g(.)$ una función de combinación de niveles de relación (véase la definición 1.14 y (1.48)), definida en $[0,1]\times[0,1]$.*

Sean $a_1 \in \mathcal{A}_1, a_2 \in \mathcal{A}_2$ dos aserciones de $\mathcal{A}_1$ y $\mathcal{A}_2$ respectivamente, entonces la conjunción de a_1 y a_2, se denota por $a_1 \wedge a_2$ y se define como:

$$\begin{aligned} a_1 \wedge a_2 : \quad & \Omega \longrightarrow [0,1] \\ & \omega \longmapsto a_1 \wedge a_2(\omega) := a_1(\omega) \wedge a_2(\omega) = g(a_1(\omega), a_2(\omega)) \end{aligned} \tag{1.115}$$

[9] *En esta Memoria, se utiliza esta definición para la definición de los nodos del árbol como conjunción de tres aserciones: la relativa al vector de variables predictoras, el evento relativo a la variable estrato y el evento relativo a la variable clase.*

Se define el conjunto $\mathcal{A}_1\hat{\wedge}\mathcal{A}_2$ *como:*

$$\mathcal{A}_1\hat{\wedge}\mathcal{A}_2 := \{a_1 \wedge a_2 | a_1 \in \mathcal{A}_1, a_2 \in \mathcal{A}_2\} \tag{1.116}$$

el conjunto de las conjunciones de elementos de $\mathcal{A}_1$ *y* $\mathcal{A}_2$*.*

Esta definición hace referencia a vectores de variables distintos para los conjuntos $\mathcal{A}_1$ y $\mathcal{A}_2$. La función $g(.)$ de combinación de niveles de relación entre dos aserciones a_1 y a_2 puede coincidir o no con las funciones de combinaciones de niveles de relación definidas internamente en las aserciones a_1 y a_2.

En realidad, esta definición permite la construcción de aserciones a partir de eventos y aserciones, dada una función de combinación de niveles de relación. El caso más elemental es la construcción de una aserción a partir de dos eventos y una función de combinación de niveles de relación (véase definición 1.19).

Se deduce fácilmente la siguiente proposición.

Proposición 1.6 *Si* $a_1 = [Y_1\mathcal{R}_1d_1] \in \mathcal{A}_1$, $a_2 = [Y_2\mathcal{R}_2d_2] \in \mathcal{A}_2$*, son dos aserciones definidas como en la definición 1.26 y* $\wedge$ *es una función de combinación de niveles de relación definida en* $[0,1] \times [0,1]$*, entonces* $a_1 \wedge a_2$ *es una aserción. Además, se identifica la conjunción de las dos aserciones con* $a_1 \wedge a_2 = [Y_1\mathcal{R}_1d_1] \wedge [Y_2\mathcal{R}_2d_2]$ *y con* $(a_1 \wedge a_2, \mathcal{R}_1 \times \mathcal{R}_2, (d_1, d_2))$*.*

Definición 1.27 ***Conjunción de aserciones definidas sobre el mismo vector de variables***[10]**.** *Sea* $\mathcal{A} = \{[Y\mathcal{R}d] : \Omega \longrightarrow [0,1], d \in \mathcal{D}(\mathcal{Y})\}$ *un conjunto de objetos simbólicos de tipo aserción asociados al vector de variables monoevaluadas o simbólicas* Y *definido sobre* Ω*, y con descripciones en un conjunto* $\mathcal{D}(\mathcal{Y})$ *que tiene definida la intersección,* $\cap$*.*

[10] *En esta Memoria, se utiliza esta definición para variables predictoras* Y_j*, definidas en los individuos* $\omega \in \Omega$*, no binarias, monoevaluadas o modales probabilistas asociadas a los conjuntos de descripciones* $\mathcal{D}_j = \mathcal{Y}_j$ *o* $\mathcal{D}_j = \mathcal{P}(\mathcal{Y}_j)$*. Y, más concretamente, para representar las conjunciones de eventos con predictores repetidos en un mismo nodo, cuando estos predictores no binarios forman parte de la representación del nodo.*

Sean $a_1 = [Y\mathcal{R}d_1] \in \mathcal{A}$, $a_2 = [Y\mathcal{R}d_2] \in \mathcal{A}$, *dos aserciones de* $\mathcal{A}$, *entonces la conjunción de* a_1 *y* a_2, *se denota por* $a_1 \wedge a_2$ *y se define como:*

$$\begin{aligned} a_1 \wedge a_2 : \quad & \Omega \longrightarrow [0,1] \\ & \omega \longmapsto a_1 \wedge a_2(\omega) := [Y\mathcal{R}d_1 \cap d_2](\omega) \end{aligned} \tag{1.117}$$

Se define el conjunto $\mathcal{A}\hat{\wedge}\mathcal{A}$ *como:*

$$\mathcal{A}\hat{\wedge}\mathcal{A} := \{a_1 \wedge a_2 | a_1, a_2 \in \mathcal{A}\} \tag{1.118}$$

el conjunto de las conjunciones de elementos de $\mathcal{A}$.

Diday (Diday, 1993a) introduce la conjunción de aserciones definidas sobre el mismo vector de variables en el caso general[11]. Según la definición 1.27, se deduce fácilmentente la siguiente proposición.

Proposición 1.7 *Si* $a_1 = [Y\mathcal{R}d_1]$, $a_2 = [Y\mathcal{R}d_2] \in \mathcal{A}$, *son dos aserciones definidas como en la definición 1.27, entonces* $a_1 \wedge a_2$ *es un aserción. Además, se identifica la conjunción de las dos aserciones con* $a_1 \wedge a_2 = [Y\mathcal{R}d_1 \cap d_2]$ *y con* $(a_1 \wedge a_2, \mathcal{R}, d_1 \cap d_2)$.

La extensión de las definiciones 1.26 y 1.27 a un conjunto de aserciones en un número mayor que dos es trivial. Así mismo, es fácil la extensión de la conjunción de dos aserciones definidas sobre vectores que comparten sólo algunas de las variables.

Ejemplo 1.5 ***Conjunción de aserciones definidas sobre el mismo vector de variables.***

[11] Sean $a_1 = [Y\mathcal{R}d] = \wedge_{i=1,..,p}[Y\mathcal{R}d_i]$, $a_2 = [Y\mathcal{R}d'] = \wedge_{i=1,..,p}[Y\mathcal{R}d'_i]$ dos aserciones definidas sobre Ω, Diday define la conjunción de a_1 y a_2, denotada por $a_1 \wedge a_2$ como:

$$\begin{aligned} a_1 \wedge a_2 : \quad & \Omega \longrightarrow [0,1] \\ & \omega \longmapsto a_1 \wedge a_2(\omega) := g(c([Y_i(\omega)\mathcal{R}_i d_i], [Y_i(\omega)\mathcal{R}_i d'_i]), i = 1, ..., p) \end{aligned}$$

$$\begin{aligned} Si \quad & a_1 = [profesión \in \{secretario, administrativo, manual\}] \\ & a_2 = [profesión \in \{secretario, dependiente\}] \\ & entonces \quad a_1 \wedge a_2 = [profesión \in \{secretario\}] \end{aligned} \tag{1.119}$$

$$\begin{aligned} Si \quad & a_1 = [profesión \sim \{secretario, administrativo, manual\}] \\ & a_2 = [profesión \sim \{secretario, dependiente\}] \\ & entonces \quad a_1 \wedge a_2 = [profesión \sim \{secretario\}] \end{aligned} \tag{1.120}$$

$$\begin{aligned} Si \quad & a_1 = [profesión \in \{secretario, administrativo, manual\}] \\ & a_2 = [profesión \in \{manual, dependiente\}] \\ & entonces \quad a_1 \wedge a_2 = [profesión \in \emptyset] = \emptyset^{\mathcal{A}} \end{aligned} \tag{1.121}$$

Proposición 1.8 *Sea $a = [Y\mathcal{R}D]$ una aserción definida sobre Ω, relativa a un vector de variables Y, $D = (D_1, ..., D_p) \in \mathcal{D}$ con $\mathcal{D} = \mathcal{Y}$ o $\mathcal{D} = \mathcal{P}(\mathcal{Y})$ y sea Y_j una variable definida sobre Ω. Se tiene que las aserciones a y $a \wedge [Y_j\mathcal{R}\mathcal{Y}_j]$ son equivalentes.*

Demostración. Si la variable Y_j es una variable distinta a las variables contenidas en el vector Y, entonces se tiene que:

$$a \wedge [Y_j\mathcal{R}\mathcal{Y}_j](\omega) = a(\omega) \wedge [Y_j(\omega)\mathcal{R}\mathcal{Y}_j] = g(a(\omega), 1) = a(\omega) \tag{1.122}$$

para cualquier función $g(.)$ de combinación de niveles de relación que verifique (1.51).

Si la variable Y_j es una variable del vector Y, entonces se tiene que:

$$a \wedge [Y_j\mathcal{R}\mathcal{Y}_j](\omega) = [Y\mathcal{R}D] \wedge [Y_j\mathcal{R}\mathcal{Y}_j](\omega) = [Y\mathcal{R}D](\omega) = a(\omega) \tag{1.123}$$

dado que

$$[Y_j \mathcal{R} D_j] \wedge [Y_j \mathcal{R} \mathcal{Y}_j] = [Y_j \mathcal{R} D_j \cap \mathcal{Y}_j] = [Y_j \mathcal{R} D_j] \tag{1.124}$$

por la definición 1.27. ■

1.6 Conclusión

En este capítulo se han introducido los datos y objetos simbólicos y se han relacionado con los datos monoevaluados. De un conjunto Ω de individuos descrito por un conjunto de descripciones de $\mathcal{Y}$ a partir de un vector de variables monoevaluadas Y se pueden obtener, para subconjuntos $E \subseteq \mathcal{P}(\Omega)$ del conjunto de sus clases, descripciones simbólicas de clases de individuos en un conjunto de descripciones $\mathcal{D}(\mathcal{Y})$. Se han presentado antecedentes de generalización por datos simbólicos de estas clases. Se ha mostrado cómo a partir de estas descripciones simbólicas se pueden definir objetos simbólicos dotándolos de relaciones de dominio que permiten la *comparación* de las descripciones de los individuos (dadas por Y) con las descripciones simbólicas. Las aserciones definidas en este capítulo, por variables, relaciones de dominio y descripciones simbólicas permiten obtener clases de individuos cuyas descripciones se relacionan con dichas descripciones simbólicas. Es decir, las aserciones son instrumentos que permiten la vuelta a la base de datos original mediante la obtención de sus extensiones.

Las aserciones son independientes de las bases de datos a partir de las cuales fueron creadas. Esto significa que las extensiones de las mismas pueden aplicarse a bases de datos diferentes o a una misma base de datos en distintos instantes de tiempo. Por lo cual, permiten la propagación de conceptos.

Por otra parte, las intenciones de las aserciones pueden ser definidas por un experto sin necesidad de ser creadas a partir de bases de datos. También de esta manera se puede acceder a una base de datos y obtener mediante una consulta a

la misma, los individuos que se relacionan con esas intenciones.

Los objetos simbólicos constituyen un nuevo sistema de representación del conocimiento que engloba en un mismo formalismo tanto conocimientos obtenidos de una base de datos como conocimientos aportados por un experto, siendo de mayor complejidad que los datos habituales. En este sentido, este sistema de representación es más rico que otros sistemas de representación del conocimiento anteriores.

En este capítulo, se ha destacado también la importancia de un marco común para los tres enfoques de expresión de incertidumbre: la probabilidad, la posibilidad y la creencia. Y se han formalizado, en este contexto, las dos primeras. Antecedentes a la creación de un marco común de representación de la incertidumbre pueden verse en Ruspini, 1990 con un modelo unificado semántico que permite comparar las tres expresiones del razonamiento aproximado y en Dubois y Prade, 1989 que proponen un marco común de combinación de información de diversas fuentes y tipos.

El formalismo presentado en este capítulo, incluye además otros tipos de datos como son los conjuntos de valores, los intervalos y la inclusión de metadatos, lo que enriquece aún más este tipo de representación. Se han formalizado los primeros e introducido los demás. Por último, destacar también que este formalismo permite analizar conjuntamente todos estos tipos de datos, siempre que se establezcan las correspondientes relaciones producto de las aserciones. Se ha introducido que este modo de representación admite formalizar datos y objetos aún de mayor complejidad como son los objetos simbólicos modales exteriores, horda, síntesis, disyunciones, etc..., que exceden el ámbito de esta Memoria.

Finalmente se han definido operaciones entre objetos simbólicos necesarias en la parte II de esta Memoria.

Algunas referencias en relación al Análisis de Datos Simbólicos, no indicadas anteriormente, son: la visualización de variables y datos simbólicos (Rouard et al., 1998, Noirhomme-Fraiture y Rouard, 2000), la Estadística descriptiva univariante

para variables simbólicas (Bertrand y Goupil, 2000), el Análisis de Conglomerados (Gowda y Diday, 1992, Esposito, 1994, Chavent, 2000, Brito, 2000, Polaillon, 2000), el Análisis de Componentes Principales y los métodos Factoriales (Cazes et al., 1997, Lauro y Palumbo, 1998, Chouakria et al., 2000, Lauro et al., 2000, Verde et al., 2000), la ayuda a la interpretación de conglomerados y ejes factoriales (Gettler-Summa, 1992, Gettler-Summa et al., 1994, Smadhi, 1994). En los artículos referenciados, la interpretación de los resultados se realiza así mismo por objetos simbólicos. Además, el conjunto de objetos simbólicos completos forman un retículo de Galois. Un objeto simbólico es completo si su extensión cubre exactamente la clase que describe (Diday y Emilion, 1996, Pollaillon y Diday, 1997).

Capítulo 2

Segmentación

En este capítulo se introducen los árboles de Segmentación en 2.1 y los árboles de Segmentación con incertidumbre en 2.2. Aunque están tratados extensamente en la literatura, se introduce aquí la representación de los árboles por objetos simbólicos que se formaliza en el capítulo 3. Además, se hace una breve presentación bibliográfica de la Segmentación en 2.1.7 y de la Segmentación con incertidumbre en 2.2.3, por ser tratada posteriormente en los capítulos siguientes. Esta última recopilación se realiza desde un punto de vista de datos y objetos simbólicos, referenciando conceptos introducidos en el capítulo 1.

2.1 Árboles de Segmentación

La primera parte del capítulo introduce los árboles de Segmentación en 2.1.1, presenta los datos de entrada en 2.1.2, el método y los pasos de los algoritmos en 2.1.3 y 2.1.4, la descripción de los nodos del árbol en 2.1.5, los criterios de elección en estos algoritmos en 2.1.6 y una revisión bibliográfica en 2.1.7.

2.1.1 Introducción

Los árboles de Segmentación o los métodos recursivos de construcción de árboles (Breiman et al. 1984, Quinlan 1986a, Cuesta, 1989, Ciampi, 1992) son en general técnicas de Análisis de Datos no paramétricas de Clasificación. Se parte de un conjunto de individuos subdivididos en *clases* conocidas y descritos por un vector de variables *explicativas* categóricas monoevaluadas. El objetivo de estas técnicas es doble, de una parte discriminar o explicar las clases de los individuos y de otra predecir la clase de nuevos individuos. Conocidas las descripciones asociadas a las variables explicativas, la interpretación de las clases se obtiene mediante la búsqueda de dependencias lógicas entre las variables explicativas y las clases en un proceso recursivo. Estas dependencias lógicas se traducen en *reglas de predicción* de las clases por los valores de algunas variables explicativas. Las clases predefinidas iniciales son así explicadas, gracias a estas reglas, en un lenguaje fácilmente interpretable. En los sucesivo, la variable de identificación de las clases es la *variable clase*.

En el proceso recursivo de obtención de dependencias lógicas se obtiene en cada paso una partición más fina del conjunto de individuos, definida por el predictor que mejor discrimina las clases, es decir, que maximiza la *calidad de predicción o explicación* de la variable clase en el nuevo corte o que minimiza los errores de predicción. La medida de calidad de predicción o explicación es una medida del *contenido de información* del árbol (de la nueva partición) con respecto a las clases predefinidas en los individuos. El proceso recursivo termina al cumplirse alguna *condición de parada*.

El origen de las técnicas de Segmentación se situa en el campo de la Estadística (Belson, 1959, Morgan y Sonquist, 1963, Cellard et al., 1967, Bouroche y Tenehaus, 1970, Messenger y Mandel, 1972). Quinlan (Quinlan, 1979) introduce la inducción de reglas de clasificación a partir de ejemplos en el año 1979 y presenta su método ID3 (Quinlan (1979, 1986a)) con lo que extiende estas técnicas

al campo de la Inteligencia Artificial donde se conocen con el nombre de Aprendizaje Automático en base a unos ejemplos de los que se conoce sus clases de pertenencia. El objetivo último de las mismas es la asignación al conjunto de clases de nuevos ejemplos de clases desconocidas. Estudios recopilatorios sobre la Segmentación pueden verse en Cuesta (Cuesta, 1989) y Périnel (Périnel, 1996).

2.1.2 Datos de partida

Sea $\Omega = \{\omega_1, ..., \omega_n\}$ un conjunto de individuos y sean los individuos $\omega \in \Omega$ descritos:

- Por las variables explicativas categóricas monoevaluadas $Y_1, \ldots, Y_p$, con dominios $\mathcal{Y}_1, ..., \mathcal{Y}_p$, $\mathcal{Y}_j$=$\{1, ..., l_j\}$, de tal forma que $Y_j(\omega) = l$ si y solo si la descripción de ω (dada por Y_j) es la categoría l-ésima del dominio $\mathcal{Y}_j$ (véase definición 1.1). Estas variables son las *variables independientes* o *predictores*.

- Y, por la variable de identificación de las clases o variable clase que es una variable categórica monoevaluada Z con dominio $\mathcal{Z} = \{1, ..., s\}$ de tal forma que $Z(\omega) = l$ si y solo si el individuo ω pertenece a la clase c_l.

El conjunto $\mathcal{Y}_1 \times ... \times \mathcal{Y}_p \times \mathcal{Z}$ es el conjunto de descripciones de individuos de Ω. Se parte de la matriz de datos monoevaluados $[X^Y|X^Z]$ asociada al conjunto de individuos Ω y al vector de variables (Y, Z) cuyas filas $(Y^{(i)}|Z^{(i)}), i = 1, .., n$ representan observaciones de n unidades experimentales. El conjunto de individuos Ω recibe el nombre de *muestra diseño*. En la fase de predicción, se parte de una matriz de datos monoevaluados $[X^Y]$ asociada al conjunto de individuos Ω_2 y referida al vector de variables Y.

2.1.3 Objetivo y Método

El objetivo es discriminar, estimar o predecir la variable clase Z en función de los predictores Y_j a partir de la matriz $[X^Z|X^Y]$ realizando particiones sucesivamente más finas del conjunto de individuos según los valores de los predictores, construyéndose de esta forma el *árbol de Segmentación*[1] y maximizando en cada paso una medida de contenido de información del árbol con respecto a las clases (Ciampi, 1992).

Es un método de clasificación *top-down* o de arriba-abajo que no supone conocimiento acerca de las distribuciones de probabilidad ('a priori') de (Y, Z) y $Y|Z$. Se parte de un nodo padre que contiene toda la población y una estimación de la variable clase derivada de dicha población. Cada nueva partición se obtiene de la anterior particionando un elemento o *nodo* generalmente en dos nuevos elementos de la partición (*nodos hijos*) por los valores de una de las variables predictoras. El corte óptimo es áquel, de entre los elementos posibles y admisibles de partición, que maximiza la *medida de contenido de información* del árbol (de la partición), $IC(T, \Omega)$, con respecto a las clases, estimada la variable clase en los nodos a partir de la matriz de datos $[X^Y|X^Z]$. En general, se estiman las

[1] En realidad un árbol es un grafo orientado y acíclico (N, A) definido por un conjunto de N nodos y un conjunto de A arcos. Cada nodo de este grafo representa un subconjunto de la población. En un árbol, se distinguen tres tipos de nodos:

- El nodo raíz que representa toda la población y que no tiene arcos entrantes.
- Los nodos terminales que no tienen arcos salientes y que representan la partición final.
- Los nodos intermedios que tienen un único arco entrante y dos arcos salientes, en el caso de un árbol binario. Los arcos salientes apuntan a los nodos hijos de este nodo, que es el nodo padre.

Un árbol cumple la propiedad que existe un camino único entre el nodo raiz y cada uno de los demás nodos.

El nivel de profundidad de un nodo se mide por la longitud del camino entre el nodo raíz y él mismo. Un nodo tiene como descendientes los nodos con los que puede establecer un camino desde él y como antecedentes los nodos desde los que puede establecer un camino hasta él.

Los nuevos subconjuntos de la partición más fina obtenida en un paso del algoritmo son nodos hijos del subconjunto o nodo del que proceden y se obtienen según los valores de algún predictor.

probabilidades ('a posteriori') condicionadas de las clases a los nodos por las probabilidades empíricas. El criterio de maximizar la *medida de contenido de información* es equivalente a minimizar la *falta de ajuste* entre la estimación de la variable clase y las clases conocidas dadas por Z. Los cortes en la población son descritos por predicados lógicos relativos a un predictor del tipo $sexo = varón$, $edad \in [0, 35]$, etc... y sus correspondientes complementarios, $sexo = mujer$, $edad \in [35, 150]$, etc...

El proceso recursivo se repite para los nuevos nodos hijos sucesivamente hasta que se cumple alguna *condición de parada* (véase 2.1.6)

Las reglas de predicción son aportadas por los nodos *terminales* o hojas del árbol que son los elementos de la partición. Por esta razón, se referirá una partición como el árbol y los elementos de la partición como los nodos. Los nodos se describen por los sucesivos cortes obtenidos del conjunto de predictores y por las estimaciones de la variable clase. Es decir, un nodo de la partición se puede ver simultáneamente como un conjunto de individuos y un conjunto de las variables predictoras y tiene asociado una estimación o predicción de la variable clase.

El árbol obtenido en el proceso recursivo se puede expresar como el conjunto de sus hojas, descritas por las variables predictoras y las estimaciones respectivas de la variable clase Z:

$$T = \{t_k\}_{k=1,\ldots,K} = \{(I_k(Y), \gamma_k)\}_{k=1,\ldots,K} \tag{2.1}$$

con:

- t_k o k, indistintamente, el nodo k-ésimo de la partición.
- K el número de hojas del árbol o elementos de la partición.
- $I_k(Y)(.)$ una función definida de Ω en $\{0, 1\}$. Dado $\omega \in \Omega$ si su descripción dada por el vector de predictores $Y(\omega)$ cumple los predicados lógicos que

describen el elemento k de la partición, entonces $I_k(Y)(\omega) = 1$. En caso contrario, $I_k(Y)(\omega) = 0$.

- γ_k la estimación de la variable clase en el elemento k de la partición. Sea $\mathcal{A}$ el conjunto de elementos posibles de estimación de la variable clase en los nodos.

La predicción de la variable clase Z para un conjunto de individuos Ω_2 descritos por los predictores $Y_1, \ldots, Y_p$ se expresa mediante (Ciampi, 1992, Ciampi et al. (1993, 1994, 1996)):

$$\gamma(Y) = \sum_{k=1,\ldots,K} I_k(Y)\gamma_k \tag{2.2}$$

visto como una aplicación:

$$\begin{array}{rccl} \gamma(Y): & \Omega_2 & \longrightarrow & \mathcal{A} \\ & \omega & \longmapsto & \sum\limits_{k=1,\ldots,K} I_k(Y)(\omega)\gamma_k \end{array} \tag{2.3}$$

De esta forma, la predicción de la variable clase en el elemento k de la partición es γ_k. Es decir, el elemento k de la partición constituye una *regla de predicción* del tipo: si un individuo $\omega \in \Omega_2$ pertenece al elemento k de la partición, entonces la estimación de la variable clase para ese individuo es γ_k.

Previamente al proceso recursivo, debe establecerse un conjunto de *elementos posibles de partición* o cortes posibles por los que realizar las particiones en la población (véase 2.1.6). Estos elementos posibles de partición vienen descritos por conjuntos de valores de los dominios de las variables predictoras. Asimismo se debe establecer una *condición de admisibilidad* que los elementos posibles de partición deben verificar desde un nodo para que sean cortes admisibles (véase 2.1.6).

También se hace necesario establecer un conjunto $\mathcal{A}$ de elementos que des-

criben la predicción o *estimación de la variable clase* para los nodos del árbol T (véase 2.1.6).

2.1.4 Esquema del algoritmo

Muy esquemáticamente el proceso recursivo se traduce en la sucesión de los siguientes pasos para cada nodo (hoja) del árbol:

- Comprobar la *condición de admisibilidad* de los cortes posibles desde el nodo. Un nodo que no tiene cortes admisibles es un nodo que no sigue el proceso recursivo y es un nodo *terminal* u hoja del árbol. Si un nodo no tiene cortes admisibles, se procede este paso de obtención de cortes admisibles con otro nodo del árbol. Si no existen nodos con cortes admisibles, se acaba el proceso recursivo.

- Obtener el mejor corte admisible desde el nodo, optimizando la *medida de contenido de información* del árbol con respecto a las clases.

- Realizar el corte, lo que da lugar a otros dos nuevos nodos del árbol. La descripción de estos nodos contiene la descripción de estos cortes.

- Aplicar *condición de parada* a los nuevos nodos.

- Estimar la predicción de la variable clase Z en los nuevos nodos obtenidos.

- Dar medida del contenido de información del nuevo árbol con respecto a las clases.

- Comenzar el proceso con un nuevo nodo del árbol, que no haya satisfecho la condición de parada.

En general, la secuencia de los nodos por los que se realizan las particiones en los sucesivos pasos del algoritmo sigue una secuencia de máximo incremento en la medida de contenido de información del nuevo árbol con respecto al conjunto de

individuos. Para simplificar el esquema del algoritmo, se ha obviado este hecho que no aporta diferencias sustanciales, salvo la secuencia de árboles que se van obteniendo.

2.1.5 Nodos del árbol

Como se ha mencionado, cada partición se identifica con un árbol (o con un nivel del árbol) y los elementos de cada partición son las hojas del árbol. Un nodo del árbol es:

1. Un elemento de la partición del conjunto de individuos y por tanto un subconjunto de individuos.

2. Un vector de descripciones del conjunto de descripciones $\mathcal{P}(\mathcal{Y}_1)\times ...\times \mathcal{P}(\mathcal{Y}_p)$ (véase definición 1.2 y criterio 1 en 2.1.6) asociado al vector de predictores Y. Más especificamente, un elemento del conjunto de descripciones $\mathcal{P}(\mathcal{Y}_{k_1}) \times ... \times \mathcal{P}(\mathcal{Y}_{k_l})$ asociado al vector de predictores $Y^k = (Y_{k_1}, ..., Y_{k_l})$ con $k_j \in \{1, ..., p\}$ y l *nivel de profundidad* del nodo. Estas descripciones son las descripciones de los nodos en el conjunto de predictores.

3. Una *descripción* de un conjunto de descripciones asociado al dominio $\mathcal{Z}$ que representa la estimación de la variable clase o predicción para la variable Z en el nodo (véase criterio 4 en 2.1.6). Esta descripción puede ser una descripción monoevaluada o simbólica. Este conjunto de descripciones se establece inicialmente para la descripción de la estimación de la variable clase (véanse 2.1.3 y 2.1.6).

Un nodo k se puede representar por una *aserción booleana* (véase definición 1.20), asociada al vector de predictores Y^k, cuya descripción es el vector de descripciones referido en el punto (2) y dotada de una de relación de dominio producto de pertenencia (véase definición 1.15). La *extensión* de esta aserción

booleana (véase definición 1.21) es el conjunto de individuos al que hace referencia el punto (1). En el capítulo siguiente se presenta la representación de un árbol de Segmentación como un conjunto de objetos simbólicos, omitida aquí su formalización por simplicidad.

2.1.6 Criterios

Como se ha mencionado en secciones anteriores, los criterios que deben establecerse en cualquier algoritmo de Segmentación son:

Criterio 1: Un conjunto B de *elementos posibles de partición* o cortes posibles. Estos cortes posibles se definen sobre los predictores Y_j y se caracterizan por la pertenencia de una observación $Y_j(\omega)$ a un subconjunto $D_j \subset \mathcal{Y}_j$ o a su complementario $\mathcal{Y}_j - D_j$. La selección de los cortes posibles es independiente de las descripciones de los individuos $\omega \in \Omega$ dada por la variable clase Z y debe realizarse al principio del algoritmo. En general son de la forma ¿$Y_j \in D_j$? e ¿$Y_j \in D_j^c$?. La descripción de estos dos cortes establecen qué individuos pertenecen a cada uno de los hijos. Los elementos posibles de partición se describen por las descripciones referidas en el punto (2) de 2.1.5.

Criterio 2: Una *condición de admisibilidad* para que los elementos posibles de partición puedan ser explorados desde un nodo del árbol. Generalmente, la condición de admisibilidad para los cortes del conjunto B desde un nodo está relacionada con:

- La presencia de predictores no descritos en el nodo.
- Si un predictor Y_j describe el nodo, la posibilidad de dividir el conjunto D_j o D_j^c en dos subconjuntos.
- La *condición de parada* del proceso recursivo para un nodo, aplicada a los nodos hijos derivados de los cortes.

Criterio 3: Una *medida de contenido de información* $IC\{T, \Omega\}$ del árbol T con respecto al conjunto de individuos Ω. IC mide la calidad de predición de la variable clase por los predictores.

Criterio 4: Un conjunto de descripciones $\mathcal{A}$ que componen la *descripción de la estimación de la variable clase*. Estas descripciones, que son referidas en el punto (3) de 2.1.5, pertenecen a un conjunto de descripciones asociado al dominio $\mathcal{Z}$ y describen la predicción o estimación de la variable clase para los nodos del árbol T. Estas descripciones pueden ser:

- Una distribución de probabilidad sobre el conjunto de clases, es decir, la estimación γ_k de la variable clase en los nodos de la partición es una distribución probabilidad sobre el conjunto de las clases.
- Una clase (Breiman et al., 1984). La estimación γ_k de la variable clase en los nodos de la partición es una clase, en lugar de las probabilidades de pertenencia a las clases. En general, se asigna a un nodo la clase de mayor probabilidad estimada, si bien se puede asignar aleatoriamente una clase según la distribución de probabilidad estimada para la variable clase en el nodo.

Criterio 5: Una *condición de parada* para un nodo del árbol. Generalmente, esta condición se traduce en que:

- El peso del nodo o número de individuos en el nodo es inferior a un umbral.
- O, que la calidad de predicción de dicho nodo, es decir, la contribución del nodo a la calidad del árbol es mayor que otro umbral.

2.1.7 Antecedentes

El criterio más importante en la construcción de árboles de Segmentación es el concerniente a la elección de la medida de contenido de información del árbol

con respecto a las clases ya que es la elección de este criterio lo que diferencia esencialmente unos algoritmos de Segmentación de otros (Breiman et al., 1984, Quinlan, 1986a, Cuesta, 1989, Shlien, 1990, Ciampi, 1992, Lerman y Da Costa, 1995). En esta sección, se detallan algunos de los criterios más utilizados, si bien esta revisión bibliográfica no es extensa ni exhaustiva.

Para muchos de los criterios descritos en la literatura se establece la medida de contenido de información como una medida de contenido de información ponderada en los nodos del árbol y expresada como:

$$IC\{T,\Omega\} = -\sum_{k=1,...,K} q^k H(p_1^k,...,p_s^k) \tag{2.4}$$

con:

- q^k el peso relativo del nodo k, como la proporción del total de individuos que son del nodo k.
- $p_i^k := P(c_i|t_k), i = 1,...,s$ la probabilidad estimada en el nodo k para la clase c_i de Z, como la proporción de individuos del nodo k que son de la clase c_i.
- $H(.)$ una función de incertidumbre aplicada a una distribución de probabilidad.

El incremento de la medida de contenido de información del árbol con respecto a las clases de un paso al siguiente, expresado en (2.4), es equivalente al incremento de la *medida de información mutua*[2]:

$$IM\{t_k,\Omega\} = H(p_1^k,...,p_s^k) - \sum_{\tilde{k}} (\frac{nq^{\tilde{k}}}{n^k}) H(p_1^{\tilde{k}},...,p_s^{\tilde{k}}) \tag{2.5}$$

[2] *La medida de información mutua ha sido referida cuando $H(.)$ es una medida de entropía.*

cuando el corte se realiza desde el nodo padre k para formar los nodos hijos $\tilde{k}$ siendo n^k la frecuencia del nodo k y n el número total de individuos.

Algunos autores en lugar de maximizar en cada paso la medida (2.4) o su equivalente (2.5), maximizan:

$$\frac{IM\{t_k, \Omega\}}{H(p_1^k, ..., p_s^k)} \tag{2.6}$$

Quinlan (Quinlan (1988, 1990)) maximiza el criterio de la *ganancia de la razón* definido como:

$$\frac{IM\{t_k, \Omega\}}{H(q^{k_1}, ..., q^{k_{l_j}})} \tag{2.7}$$

con $(q^{k_1}, ..., q^{k_{l_j}})$, las probabilidades empíricas de las categorías $(1, ..., l_j)$ del predictor Y_j condicionadas por los predicados que definen el nodo k y considerando que el nodo k se divide en l_j nodos hijos. Es decir, para un predictor Y_j cuyo nodo hijo se caracteriza por $Y_j = l$, entonces $q^{k_l} := P(Y_j = l|t_k)$. De igual modo, si los nodos hijos de k se obtienen por agrupación de categorías, el denominador de (2.7) se modifica en correspondencia.

Los criterios que puede expresarse según (2.4) dan distintas definiciones de la función $H(.)$:

- El criterio de máxima reducción de la *entropía de Shannon* de un paso al siguiente (Casey y Nagy, 1984, Quinlan[3], 1986a, Kononenko y Bratko, 1987, Selby y Porter, 1988, Manago, 1991, Shlien, 1992). Bergomier y Boucharenc introducen este criterio en 1966 y lo hacen para árboles binarios y $s = 2$ (Bergomier y Boucharenc, 1966). También fué indicado por Messenger y Mandel, 1972.

[3]Quinlan propone particiones no necesariamente binarias, por lo que el criterio favorece los predictores con más categorías.

Es conocido que la *entropía de Shannon* (Shannon, 1948) para una distribución de probabilidad $(p_1, ..., p_s)$ es:

$$H(p_1, ..., p_s) = -\sum_{i=1,...,s} p_i \log_2(p_i) \tag{2.8}$$

y es una medida de la incertidumbre de un experimento aleatorio de s sucesos de probabilidades $(p_1, ..., p_s)$.

La medida de contenido de información es el opuesto de la entropía media en el árbol, estimadas las probabilidades de las clases por las probabilidades empíricas en cada uno de los nodos. Es decir :

$$IC\{T, \Omega\} = \sum_{k=1,...,K} q^k \sum_{i=1,...,s} p_i^k \log_2(p_i^k) \tag{2.9}$$

En este caso, $H(p_1^k, ..., p_s^k)$ representa la entropía estimada de la variable clase en el nodo k e $IC\{T, \Omega\}$ el opuesto de la entropía ponderada media estimada de la variable clase en los nodos del árbol.

- Criterio de máxima reducción de la *entropía modificada de Shannon*:

$$H(p_1, ..., p_s) = -\sum_{i=1,...,s} p_i \log_2 \sqrt{p_i} \tag{2.10}$$

- Criterio de máxima reducción de la *función de incertidumbre*[4] (Breiman et al., 1984):

$$H(p_1, ..., p_s) = 1 - \max_{i=1,...,s} \{p_i\} \tag{2.11}$$

- Criterio de máxima reducción del *índice de diversidad de Gini*[5] (Breiman

[4] La función de incertidumbre aplicada a un nodo coincide con la probabilidad estimada de mal clasificados cuando se asigna en un nodo la clase de mayor probabilidad estimada.

[5] El índice de diversidad de Gini aplicado a un nodo coincide con la probabilidad estimada de

et al., 1984, Shlien, 1990) que sustituye al criterio anterior por dar mejores resultados:

$$H(p_1, ..., p_s) = 1 - \sum_{i=1,...,s} p_i^2 = \sum_{i \neq j} p_i p_j \tag{2.12}$$

que mide la probabilidad de que dos individuos caídos al azar en un nodo provengan de clases diferentes.

Otros criterios considerados son criterios asociados al estadístico χ^2:

- El estadístico χ^2 (Hunt et al., 1966, Hart, 1984, Mingers, 1987) o asociados seleccionan como variable de segmentación aquella que tenga un valor mayor del estadístico derivado de la tabla de contingencia que se construye con la población de un nodo padre cruzando las descripciones de los cortes dadas por un predictor con las clases. El estadístico χ^2 es:

$$\chi^2 = n^k \sum_{\tilde{k}} \sum_{i=1,...s} \frac{(n_i^{\tilde{k}} - \frac{n_i^k n^{\tilde{k}}}{n^k})^2}{n_i^k n^{\tilde{k}}} \tag{2.13}$$

  con: n^k la frecuencia del nodo padre k, $n_i^{\tilde{k}}$ la frecuencia en el nodo hijo $\tilde{k}$ de la clase c_i, n_i^k la frecuencia en el nodo padre k de la clase c_i y $n^{\tilde{k}}$ la frecuencia del nodo hijo $\tilde{k}$.

- Cellard et al. (Cellard et al., 1967) y Bouroche y Tenehaus, (Bouroche y Tenehaus, 1970), proponen el estadístico

$$\phi^2 = \frac{\chi^2}{n^k} \tag{2.14}$$

  que es independiente del tamaño del conjunto de individuos en el nodo k.

mal clasificados cuando se asigna aleatoriamente una clase según la distribución de probabilidad estimada de la variable clase en el nodo.

- Cellard et al. también proponen el estadístico *coeficiente de Tschuprow*

$$T^2 = \frac{\phi^2}{\sqrt{(s-1)}} \tag{2.15}$$

que es normalizado en $[0, 1]$, para árboles binarios. En (2.27) puede verse una definición más general de este estadístico.

- Algunos autores (Kaas, 1980, Mingers, 1987) apuntan el uso del *estadístico exacto de Fisher* (Fisher, 1934) en lugar del estadístico χ^2 cuando el numero de observaciones en un nodo es pequeño y en el caso de $s = 2$.

Otros criterios considerados en la literatura son:

- Criterio de la mayor disminución relativa de la *probabilidad estimada del error de predicción* (Messenger y Mandel, 1972). La probabilidad del error de predicción en un nodo se estima por 1 menos el máximo de las probabilidades estimadas de las clases. El criterio es la minimización se expresa como:

$$\lambda = \frac{\sum_{\tilde{k}} p^{\tilde{k}} - p^k}{1 - p^k} \tag{2.16}$$

con:

  * p^k el máximo de las probabilidades estimadas de las clases en un nodo padre k.
  * $p^{\tilde{k}}$ el máximo de las probabilidades estimadas de las clases en el nodo hijo $\tilde{k}$.

El criterio λ mide la reducción proporcional del error de predicción cuando se realiza la predicción modal, al realizar el corte desde el nodo k que proporciona los nodos hijos $\tilde{k}$.

- Criterio de la máxima *distancia media de la distribución estimada para la variable clase* en el nodo padre y los nodos hijos ponderadas por los pesos de los nodos hijos (Messenger y Mandel, 1972). Es decir, maximizar:

$$\sum_{\tilde{k}} q^{\tilde{k}} d((p_1^k, ..., p_s^k), (p_1^{\tilde{k}}, ..., p_s^{\tilde{k}})) \tag{2.17}$$

con $d(.,.)$ una distancia definida entre distribuciones de probabilidad. Una distancia propuesta es la del valor absoluto.

- Spangler et al. (Spangler et al., 1988) proponen los árboles binarios y minimizar:

$$R(\tilde{k}) = \frac{\sqrt{\sum_{i=1,...,s} \frac{\frac{n^{\tilde{k}}}{n^k} n_i^k (n_i^k - \frac{n^{\tilde{k}}}{n^k} n_i^k)}{n_i^k}}}{\sqrt{\sum_{i=1,...,s} (\frac{n^{\tilde{k}}}{n^k} n_i^k - n_i^{\tilde{k}})^2}} \tag{2.18}$$

el cociente entre el error estandar asociado al vector de frecuencias esperadas $(\frac{n^{\tilde{k}}}{n^k} n_1^k, ..., \frac{n^{\tilde{k}}}{n^k} n_s^k)$ de las clases en el nodo hijo $\tilde{k}$ suponiendo no asociación con el predictor (Quinlan, 1986b) y la distancia geométrica entre este vector y el vector de frecuencias de las clases obtenidas en el hijo $\tilde{k}$.

Si un predictor Y_j tiene l_j categorías, $R(\tilde{k})$ se refiere al corte que se prueba con una de las categorías de $\mathcal{Y}_j$ frente al resto. Se tiene que $R(\tilde{k}) = R(\tilde{k}^c)$, con $\tilde{k}^c$ el nodo hijo definido por las categorías que no definen el nodo $\tilde{k}$.

- Maximización de otras *medidas de asociación* para matrices de contingencia (Belson, 1959, Hugues et al.[6], 1970, Cuesta, 1989, Mola y Siciliano, 1992, Lerman y Da Costa, 1995). Así, por ejemplo:

[6] Proponen una generalización del criterio de Belson para $s > 2$.

– Belson propone para árboles binarios y $s = 2$, maximizar:

$$\max_{i=1,2;\tilde{k}} |\frac{n_i^{\tilde{k}}}{n^k} - \frac{n_i^k}{n^k}\frac{n^{\tilde{k}}}{n^k}| = \max_{i=1,2;\tilde{k}} |p_i^{\tilde{k}}\frac{n^{\tilde{k}}}{n^k} - p_i^k\frac{n^{\tilde{k}}}{n^k}| = \max_{i=1,2;\tilde{k}} |(p_i^{\tilde{k}} - p_i^k)\frac{n^{\tilde{k}}}{n^k}| \tag{2.19}$$

– Mola y Siciliano proponen maximizar el *índice τ de predictabilidad de Goodman-Kruskal* (Goodman y Kruskal, 1954), haciendo posible en su modelización otros índices de predictabilidad. El índice de predictabilidad aplicado a las distribuciones de probabilidad estimadas en los nodos hijos se define como una función no negativa, simétrica, con un único valor máximo cuando una de las probabilidades es 1 y un único valor mínimo cuando todas las probabilidades son iguales a $\frac{1}{s}$.

- Breiman et al., 1984 proponen maximizar el criterio:

$$q^{k_1}q^{k_2}\sum_{i=1,\ldots,s}(p_i^{k_1} - p_i^{k_2})^2 \tag{2.20}$$

con k_1 y k_2 los dos nodos hijos, $q^{\tilde{k}}$ peso relativo del nodo $\tilde{k}$ y $(p_1^{\tilde{k}}, ..., p_s^{\tilde{k}})$ distribución estimada para la variable clase en el nodo $\tilde{k}$, $\tilde{k} = k_1, k_2$.

- Mingers (Mingers, 1987) propone maximizar *el estadístico G para tablas de contingencia* basado en la teoría de la información, introducido por Kullback (Kullback, 1967) que coincide con el *estadístico G*, introducido por Sokal y Rohlf, 1981:

$$G = 2\sum_{\tilde{k}}\sum_{i=1,\ldots s} n_i^{\tilde{k}}\log_2\frac{n^k n_i^{\tilde{k}}}{n_i^k n^{\tilde{k}}} \tag{2.21}$$

Kullback demuestra que la distribución del estadístico G, bajo hipótesis de independencia, sigue una distribución χ^2.

Además (Mingers, 1987), se tiene que si $H(.)$ es la entropía de Shannon (véase (2.8)) en la expresión de $IM\{t_k, \Omega\}$ en (2.5), entonces se verifica:

$$G = 2n^k IM\{t_k, \Omega\} \tag{2.22}$$

siendo $IM\{t_k, \Omega\}$ en este caso, la *medida de información discriminante de Kullback-Leibler*[7] (Kullback*, 1981):*

$$IM\{t_k, \Omega\} = \sum_{\tilde{k}} \sum_{i=1,...s} \frac{n_i^{\tilde{k}}}{n^k} \log_2 \frac{\frac{n_i^{\tilde{k}}}{n^k}}{\frac{n^{\tilde{k}}}{n^k}\frac{n_i^k}{n^k}} \tag{2.23}$$

aplicada a la distribución de probabilidad empírica conjunta para el vector de variables $(Y_j, Z|t_{\tilde{k}})$ y al producto de las distribuciones de probabilidad empíricas para las variables $Y_j|t_k$ y $Z|t_k$. Este estadístico mide la divergencia entre la distribución conjunta y la distribución producto.

Mingers compara árboles construidos maximizando el criterio G (en (2.21)) y el criterio IM (en (2.23)), resultando estos últimos más cortos.

- Ciampi (Ciampi, 1992, 1994) propone maximizar el criterio:

$$n^k IM\{t_k, \Omega\} \tag{2.24}$$

con $IM\{t_k, \Omega\}$ la *medida de información discriminante de Kullback-Leibler* (véase (2.23)), para árboles binarios. Ciampi asume una modelización en los nodos del árbol de la distribución condicional de Z con respecto a Y, binomial en el caso de $s = 2$ y multinomial en el caso de $s > 2$, dependientes de un parámetro δ. Propone la maximización del *estadístico de la razón de verosimilitud*[8]. La modelización general de Ciampi permite la inclusión de

[7] Esta medida coincide con la divergencia de Kullback-Leibler de (1.76).

[8] para la hipótesis de que el parámetro δ varía en las subpoblaciones de la partición creada, frente a la alternativa de que esto no es así.

más parámetros y la maximización de otros criterios.

- Nuñez (Nuñez, 1991) propone incorporar conocimiento en la fase de creación del árbol. Propone que la medida que debe minimizarse en cada paso del algoritmo sea el cociente de una función de coste aplicada al predictor y una función de la eficiencia discriminadora del predictor, como un *cociente coste/beneficio*.

- Otros[9].

Pueden verse estudios de comparación de criterios de corte en Mingers, 1989, Cuesta, 1989, Buntine y Niblett, 1992, Lerman y Da Costa, 1995.

Estimación de las probabilidades de las clases

Muestreo aleatorio simple. Los criterios de maximización precedentes consideran las probabilidades estimadas de las clases en los nodos, proporcionales a los ejemplos observados de las clases en esos nodos. Esto tiene su justificación en el caso de que la muestra diseño para la construcción del árbol sea tomada mediante muestreo aleatorio simple.

Asimismo, Casey y Nagy, 1984 y Shlien, 1992 proponen estimar las probabilidades de las clases en un nodo k como:

$$p_i^k = \frac{n_i^k + 1}{n^k + 2}, i = 1, ..., s \tag{2.25}$$

[9]Dada la importancia que las Redes Neuronales adquieren en el Análisis de Datos, aunque las referencias aportadas sólo estudian variables continuas, se incluye en esta nota a pie de página.

Más recientemente, se han combinado los árboles de Segmentación con las redes neuronales debido a similitudes en ambas estructuras y aprovechando las ventajas de ambas: la fácil interpretación de los árboles de Segmentación y la precisión de las reglas de decisión producidas por las redes neuronales (Guo y Gelfand, 1992, Chabanon et al., 1992, Sethi, 1995).

Algunos autores construyen árboles de redes neuronales obteniéndose en cada nodo una función no lineal de clasificación. Otros autores transforman el árbol obtenido en una red neuronal para mejorar la clasificación.

con:

- n_i^k número de individuos en el nodo k que son de la clase c_i.
- n^k número de individuos en el nodo k.

El estimador $p_i^k \in (0,1)$ de (2.25) es un estimador insesgado (véase Casey y Nagy, 1984, Shlien, 1992) y evita los problemas de singularidades al ser sus valores distintos de 0 y 1.

Otros tipos de muestreo. En caso de muestreo en cada una de las clases o en cualquier otro tipo de muestreo se proporcionan las probabilidades *a priori* $(\pi_1, ..., \pi_s)$ de las clases. En consecuencia, la estimación de las probabilidades de las clases en un nodo k se determinan por la fórmula de Bayes como:

$$p_i^k = \frac{\frac{n_i^k}{n^k}\pi_i}{\sum\limits_{l=1,..s} \frac{n_l^k}{n^k}\pi_l}, i = 1, ..., s \tag{2.26}$$

Predicción en los nodos y número de clases

Por lo general, el número de clases más extendido en la literatura es dos. No obstante, casi todos los criterios presentados en este capítulo son fácilmente extensibles para un número de clases superior y por este motivo, su expresión en este capítulo está basada en un número de clases s genérico.

En cuanto a la predicción de la variable clase en los nodos del árbol, existen tres corrientes mayoritarias:

- Se proporciona como valor de predicción en cada nodo, la distribución de probabilidad estimada de la variable criterio.
- O, se proporciona como valor de predicción en cada nodo, la clase de mayor probabilidad estimada en el nodo.
- O, se asigna aleatoriamente una clase según la distribución de probabilidad estimada en el nodo para la variable clase.

El caso de los predictores no binarios

Esta Memoria se centra en la *Segmentación binaria* como la mayor parte de las referencias bibliográficas de la literatura. En general, en el caso de predictores categóricos no binarios, se realiza una binarización de los mismos para realizar los cortes desde un nodo padre. La binarización más extendida en la literatura es la obtención de todas las particiones posibles en dos conjuntos de categorías. Excepciones a esto puede verse en Kononenko y Bratko, 1987, Splanger et al., 1988, Mola y Siciliano, 1991, Chow, 1991, Lerman y Da Costa, 1995, de utilidad si el número de categorías es elevado.

Chow propone, cuando el número de categorías es elevado, para cada binarización una clasificación de tipo K-medias usando como distancia una generalización de la divergencia de información de Kullback (véase (1.76)) aplicada a las distribuciones de probabilidad estimadas para las clases en cada una de las categorías. Lerman y Da Costa proponen también para variables con muchas categorías un algoritmo de clasificación jerárquica obteniendo dos conglomerados.

Para la binarización de variables ordinales, el número de particiones que se prueban es inferior al de las variables categóricas. Este número es igual al número de categorías de la variable menos 1, correspondientes a los cortes posibles en la escala de valores de la variable ordinal.

Tambien es de destacar que se introducen variantes a la Segmentación binaria cuando los predictores son categóricos no binarios:

- Desde un nodo padre, crear tantos nodos hijos como categorías tiene un predictor (Quinlan, 1979, Mingers, 1987). Los criterios de corte expuestos anteriormente en esta sección favorecen los predictores con mayor número de categorías. Por este motivo, algunos autores proponen alternativas a los criterios de máxima reducción de la *entropía de Shannon* y mínimo valor del *estadístico* χ^2:
  - Quinlan (Quinlan (1988, 1990)) propone maximizar el criterio de la

ganancia de la razón[10] definido en (2.7).

– Cellard et al. (Cellard et al., 1967) proponen el estadístico

$$T^2 = \frac{\chi^2/n^k}{\sqrt{(l_j - 1)(s-1)}} \tag{2.27}$$

coeficiente de Tschuprow, que es normalizado, para un predictor de l_j categorías. Proponen cortes no necesariamente binarios.

– Mingers (Mingers, 1987) propone normalizar el estadístico χ^2 dividiéndolo por su desviación típica.

- Algunos algoritmos (Kaas, 1980, Cuesta, 1989) combinan la división en tantos hijos como categorías de una variable, con la unión de varios de ellos para obtener menor número de cortes. En algunos de ellos, se llega incluso a una binarización del predictor posterior a la división (Lerman y Da Costa, 1995).

  – Kaas (Kaas, 1980) propone un algoritmo jerárquico de Análisis de Conglomerados, usando la distancia de Benzecri (Benzecri, 1973), aplicado a las filas de la tabla de contingencia que cruza un predictor (en las filas) con la variable clase (en las columnas). El algoritmo determina el número de hijos.

  – Con esta misma idea, Cuesta (Cuesta, 1989) presenta la aplicación del algoritmo de las nubes dinámicas de Diday (Diday et al., 1980) de Análisis de Conglomerados a los perfiles fila de dicha tabla. Se fija previamente el número nh deseado de nodos hijos y se seleccionan al azar nh representantes, que son los perfiles de sendas categorías. El resto de categorías se asignan a estos hijos según la distancia χ^2 sea

[10] Este criterio favorece tamaños muy desiguales en los nodos hijos, produciendo nodos de tamaño muy pequeño (véase Mingers, 1989).

menor a sus representantes. Una vez asignadas todas las categorías, se obtienen nuevos representantes (los centros de gravedad) en estos nh hijos y se procede sucesivamente a la reasignación de las categorías a los hijos con centros de gravedad más próximos hasta que alguna condición de parada se verifique.

2.2 Árboles de Segmentación con incertidumbre

En esta segunda parte del capítulo se introduce la incertidumbre[11] en los árboles de Segmentación en 2.2.1. En 2.2.2 se presentan brevemente los datos de entrada, el método, la descripción de los nodos y los criterios de consideración en árboles de Segmentación con incertidumbre. En 2.2.3 se presentan algunos antecedentes de árboles de Segmentación con incertidumbre.

2.2.1 Introducción

En la sección anterior se introducen los árboles de Segmentación y los criterios de obtención del corte óptimo más frecuentes. En esta sección, se presentan algunos antecedentes del tratamiento de la presencia de incertidumbre en los algoritmos de Segmentación. Estudios recopilatorios sobre la Segmentación con incertidumbre pueden verse en Périnel (Périnel, 1996). Por lo general, la presencia de incertidumbre es descrita en los predictores aunque también ha sido considerada en ocasiones la incertidumbre en las clases de los individuos.

Puede suceder en la clasificación de ejemplos que la descripción de los individuos en los predictores no sea discreta, es decir, que los individuos no pertenezcan de forma precisa a ninguna de las categorías del predictor, sino que la pertenencia a las categorías sea descrita por una distribución de posibilidad o por conjuntos difusos. En estos casos, la incertidumbre introduce cierta flexibilidad en las fron-

[11] Se engloba en el término incertidumbre también la imprecisión y la vaguedad.

teras de diversas categorías. Esta expresión de incertidumbre ha sido también tratada para la variable clase Z.

Puede acontecer, así mismo, que el conocimiento de la categoría de un individuo sea incierto y que la pertenencia a las categorías sea descrita por una distribución de probabilidad. Esta expresión de incertidumbre también puede representar una variación de un conjunto de individuos respecto de las categorías.

En este marco, los datos de partida pueden ser expresados en diversas semánticas como conjuntos difusos, distribuciones de posibilidad, distribuciones de probabilidad, etc... Sin embargo, la literatura ha tratado más frecuentemente la expresión de la incertidumbre por conjuntos difusos.

En general, la incertidumbre es considerada tanto en la fase de creación del árbol como en la de predicción. Sin embargo, algunos autores sólo consideran la incertidumbre en la fase de predicción.

Las diferencias esenciales con respecto a los árboles de Segmentación tradicionales son de una parte, que las particiones obtenidas son en general particiones difusas o con incertidumbre y de otra que las medidas de contenido de información toman en consideración la incertidumbre.

2.2.2 Método con incertidumbre

Datos de partida

La matriz de entrada de los datos $[X^Y|X^Z]$ es una matriz de datos simbólicos asociada al conjunto de individuos u objetos $E = \{e_1, ..., e_n\}$ y al vector de variables simbólicas (Y, Z). El vector de variables simbólicas Y es un vector de variables probabilistas o variables posibilistas. En general, el conjunto de descripciones de elementos de E es el conjunto $\mathcal{M}(\mathcal{Y}_1) \times ... \times \mathcal{M}(\mathcal{Y}_p) \times \mathcal{Z}$ cuando la *variable clase* es una variable categórica monoevaluada. Si bien, aunque raramente, el conjunto de descripciones de elementos de E puede ser el conjunto $\mathcal{M}(\mathcal{Y}_1) \times ... \times \mathcal{M}(\mathcal{Y}_p) \times \mathcal{M}(\mathcal{Z})$ si la expresión de la incertidumbre también afecta

a las clases. Para mayor detalle de la expresión de la incertidumbre en las diversas semánticas puede consultarse el capítulo anterior.

Si la incertidumbre afecta sólo a la fase de predicción, la matriz de entrada de los datos en la fase de creación del árbol es una matriz de datos como en 2.1.2.

En la fase de predicción, se parte de una matriz de datos simbólicos $[X^Y]$ referida al vector de variables simbólicas Y, asociada a un conjunto de individuos E_2, con $\mathcal{M}(\mathcal{Y}_1) \times ... \times \mathcal{M}(\mathcal{Y}_p)$ el conjunto de descripciones de sus elementos.

Objetivo y método

El objetivo y método son los mismos que los de los árboles de Segmentación (véanse 2.1.3 y 2.1.4), sólo que ahora las reglas de predicción obtenidas son en general difusas o con incertidumbre. Los árboles de Segmentación obtenidos producen por lo general, particiones difusas o con incertidumbre, ya que los individuos:

- No se reparten de forma discreta entre los nodos hijos de un padre, sino que cada individuo puede *pertenecer* a todos los hijos de un padre con un determinado *grado de pertenencia*.

- O bien, tienen una *probabilidad* o *posibilidad* de pertenencia a los nodos hijos.

En general, los individuos son *distribuídos* por el árbol en función de estos grados de pertenencia o de estas probabilidades o posibilidades.

La representación del árbol expresada en (2.1) queda ahora modificada en el sentido que $I_k(Y)(.)$ es ahora una función definida de E en $[0, 1]$. La descripción de los cortes o bien se realiza ahora con diversas semánticas distintas de la monoevaluada o multievaluada, como los conjuntos difusos; o bien, la semántica de descripción de los cortes es la monoevaluada o multievaluada y los individuos son

distribuídos a los hijos según distribuciones de posibilidad o probabilidad, etc..., dependiendo de la semántica de representación de los individuos.

La predicción de la variable clase Z a partir del vector de predictores, según la expresión (2.2) y (2.3), es ahora una ponderación de las predicciones estimadas en los nodos del árbol. Estas ponderaciones son los grados de pertenencia o las probabilidades y posibilidades comentadas anteriormente.

Generalizando la expresión (2.2), se puede establecer la predicción de la variable clase Z como:

$$\gamma(Y) = h(\{(I_k(Y), \gamma_k)\}_{k=1,\ldots,K}) \tag{2.28}$$

la aplicación de $h(.)$ a las ponderaciones en los nodos y a sus predicciones respectivas, es decir, vista como una aplicación:

$$\begin{array}{rlll} \gamma(Y): & E_2 & \longrightarrow \mathcal{A} \\ & e & \longmapsto h(\{(I_k(Y)(e), \gamma_k)\}_{k=1,\ldots,K}) \end{array} \tag{2.29}$$

con $\mathcal{A}$ el conjunto de elementos posibles de estimación de la variable clase en los nodos. Este puede ser el caso por ejemplo de particiones con incertidumbre en el que la predicción de un individuo es la de la rama más probable y así, se define $\gamma(Y)(.)$ como:

$$\gamma(Y)(e) = h(\{(I_k(Y)(e), \gamma_k)\}_{k=1,\ldots,K}) = \gamma_{k_0} | k_0 = \arg\max_{k=1,\ldots,K} \{I_k(Y)(e)\} \tag{2.30}$$

donde el operador $\arg\max(.)$ devuelve el índice k que hace máxima la expresión que encierra.

En cada paso del algoritmo se maximiza una medida de contenido de información del árbol con respecto a las clases que toma en consideración los grados de pertenencia, las distribuciones de posibilidad o de probabilidad de los individuos sobre los nodos del árbol y las predicciones de la variable clase Z en los mismos.

Particiones

A continuación se definen las particiones difusas y con incertidumbre.

Definición 2.1 ***Partición difusa.*** *Sea E un conjunto finito de individuos, sea $T = \{t_1, ..., t_K\}$ una familia de conjuntos difusos de E con funciones de pertenencia*

$$\begin{aligned} \phi_{t_k} : \quad & E \longrightarrow [0,1] \\ & e \longmapsto \phi_{t_k}(e) \end{aligned} \tag{2.31}$$

que a cada elemento $e \in E$ le asocia el grado de pertenencia $\phi_{t_k}(e)$, para $k \in \{1, ..., K\}$.

T es una ***partición difusa*** *(Ruspini, 1969) de E si se cumple:*

$$\forall e \in E, \text{ se tiene que } \sum_{t \in T} \phi_t(e) = 1 \tag{2.32}$$

El ***peso*** *del* ***elemento de la partición*** *$t \in T$ es:*

$$\sum_{e \in E} \phi_t(e) \tag{2.33}$$

Se tiene que la suma de los pesos de los elementos de una partición difusa definida en E es el cardinal de E. Es decir, se cumple que:

$$\sum_{t \in T} \sum_{e \in E} \phi_t(e) = Card(E) \tag{2.34}$$

La partición difusa en el sentido de Ruspini, es decir la que verifica (2.32), que la suma de grados de pertenencia de los individuos a los elementos de la partición es uno, está muy extendida en la literatura. Sin embargo, este requisito resulta muy restrictivo y en el caso de la semántica difusa o posibilista es innecesario. En general, cuando las categorías de un predictor o de la variable clase son conjuntos

difusos, se cumple que las funciones de pertenencia a las categorías aplicadas a un individuo suman uno[12]. Sin embargo, existen aplicaciones de los árboles de Segmentación con particiones difusas que no cumplen este requisito. Ejemplos de estas aplicaciones pueden verse en Séchet, 1995 para incertidumbre en los predictores y en Rives, 1990 y Araya, 1995 para incertidumbre en las clases. Y más aún, en Rives[13], 1990, la distribución de posibilidad de un individuo sobre las clases no es ni normalizada ni representa una partición difusa en el sentido de Ruspini. Más recientemente, Del Amo (Del Amo, 1999) opina también que la propuesta inicial de Ruspini representa una situación muy restrictiva en la práctica de la modelización difusa, siendo con frecuencia que las clases difusas no definen una partición de Ruspini y que pueden existir situaciones en las que sea deseable que la clasificación no cumpla estos requisitios.

Definición 2.2 ***Partición con incertidumbre.*** *Sea E un conjunto finito de individuos y sea T una variable de clasificación de K clases. Cada elemento $e \in E$ lleva asociada una distribución de probabilidad $(p_1^e, ..., p_K^e)$ sobre las K clases. Se dice que T es una* ***partición de E con incertidumbre***

Nodos del árbol

Un nodo del árbol en presencia de incertidumbre es ahora:

1. Un elemento de la partición difusa o con incertidumbre del conjunto de individuos.

[12] Y, por consiguiente, las particiones definidas por los predictores correspondientes, en la fase de creación del árbol son particiones difusas en el sentido de Ruspini, al ser la descripción de las ramas expresada por subconjuntos de categorías difusas del predictor. Esto se verifica siempre y cuando, el grado de pertenencia (o nivel de relación) de los individuos a una rama sea la suma de los grados de pertenencia de los individuos a las categorías que la definen. Para que se siga manteniendo la partición según Ruspini, en el árbol global, la función de combinación de niveles de relación (que se establece entre los eventos que definen las ramas) debe conservar esta propiedad (Ej. producto). Véase una aplicación con estas propiedades en árboles de Segmentación en Verde, 1995.

[13] Rives opera con las distribuciones de posibilidad marginales de las clases difusas de Z.

2. Un vector de descripciones del conjunto de descripciones $\mathcal{P}(\mathcal{Y}_1)\times ...\times \mathcal{P}(\mathcal{Y}_p)$ o del conjunto $\mathcal{M}(\mathcal{Y}_1) \times ... \times \mathcal{M}(\mathcal{Y}_p)$, asociado al vector de predictores Y. Más especificamente, un vector de un conjunto de descripciones $\mathcal{P}(\mathcal{Y}_{k_1}) \times ... \times \mathcal{P}(\mathcal{Y}_{k_l})$ o $\mathcal{M}(\mathcal{Y}_{k_1}) \times ... \times \mathcal{M}(\mathcal{Y}_{k_l})$, asociado al vector de predictores $Y^k = (Y_{k_1}, ..., Y_{k_l})$ con $k_j \in \{1, ..., p\}$ y l nivel de profundidad del nodo. Estas descripciones son las descripciones de los nodos en el conjunto de predictores.

3. Al igual que en 2.1.5, una *descripción* monoevaluada o simbólica de un conjunto de descripciones asociado al dominio $\mathcal{Z}$ que representa la estimación de la variable clase o predicción para la variable Z en el nodo.

Al igual que en 2.1.5, en el caso de árboles con incertidumbre, un nodo k se puede representar por una *aserción* (véase definición 1.19), asociada al vector de predictores Y^k, cuya descripción es el vector de descripciones referido en el punto (2) y dotada de un vector de relaciones de dominio difusas o probabilistas (véase la definición 1.14) y de una función g de combinación de niveles de relación (véase la definición 1.19). La *extensión* de esta aserción (véase definición 1.21) es el conjunto de individuos al que hace referencia el punto (1). Véase el capítulo siguiente para más detalles sobre este modo de representación.

Los grados de pertenencia a los nodos del árbol o las probabilidades / posibilidades de los nodos para los individuos referidos en 2.2.2 son los *niveles de relación* que presentan los individuos con los nodos del árbol, representados éstos por aserciones difusas o probabilistas. Los pesos de los nodos resultan de la suma en E de los niveles de relación de los elementos de E con los nodos respectivos.

Criterios

Los criterios que deben establecerse son los mismos que los de 2.1.6 solo que adaptados a las semánticas de incertidumbre utilizadas. Es decir,

- La descripción de los *elementos posibles de partición* puede venir dada en semántica difusa y no únicamente en semántica monoevaluada o multievaluada.

- La *medida de contenido de información* considera la incertidumbre contenida en los datos y, en su caso, la contenida en la descripción de los nodos.

- Las descripciones de la *estimación de la variable clase* Z cubren otras formas de representación de la incertidumbre, como por ejemplo distribuciones de posibilidad, conjuntos difusos, etc...

Además, en relación a las aserciones que describen los nodos, deben establecerse los siguientes criterios que consideran la incertidumbre de los datos y/o nodos:

- Las *relaciones de dominio* que se establecen entre las descripciones de los individuos en los predictores y las descripciones de los cortes, que establecen los niveles de relación de los individuos con los cortes.

- Una *función* $g(.)$ *de combinación de niveles de relación* que establece el nivel de relación de los individuos con los nodos, partiendo de los niveles de relación de los individuos con los cortes que definen cada nodo.

- Una aplicación $h(.)$ que proporciona un valor de predicción para un elemento $e \in E$, en función de los niveles de relación del elemento e con los nodos del árbol y las predicciones estimadas en los mismos.

2.2.3 Antecedentes

Como ya se ha introducido, la incertidumbre afecta en general a los predictores y no a las clases. Sin embargo, algunos autores consideran que los datos de partida

contienen incertidumbre en las clases (Rives, 1990, Yuan y Shaw[14], 1995, Araya, 1995, Séchet, 1995).

En ocasiones las categorías difusas de los predictores se construyen a partir de una variable de rango continuo que se subdivide en categorías y para las cuales se definen funciones de pertenencia (Wang y Mendel, 1992, Zeidler y Schlosser, 1994, Verde, 1995) (véase pie de página en pag. 36). Los grados de pertenencia de un individuo a las nuevas categorías difusas se obtienen aplicando las correspondientes funciones de pertenencia a la puntuación de la variable original.

Las funciones de pertenencia más utilizadas en los algoritmos de Segmentación con conjuntos difusos son trapezoidales (Dubois et al., 1991, Zeidler y Schlosser, 1994, Séchet, 1995) y triangulares (Maher y St.Clair, 1993). Verde (Verde, 1995) propone una función logística como función de pertenencia. Algunos algoritmos determinan automáticamente las cotas necesarias para construir estas funciones de pertenencia (Zeidler y Schlosser, 1994, Verde, 1995).

Algunos de los métodos que expresan la incertidumbre por conjuntos difusos, basan los criterios de Segmentación en la minimización de una medida de entropía definida en el marco de la teoría de las posibilidades (Rives, 1990, Yuan y Shaw, 1995). Muy extendido es el criterio de minimización de la *entropía difusa*[15] (Séchet, 1995, Verde, 1995).

Otros autores proponen el uso de incertidumbre en los datos para la asignación de los individuos a las ramas del árbol sólo en la fase de predicción y por tanto introducen una fuente de incertidumbre en la predicción (Quinlan, 1990, Maher y St.Clair, 1993).

La expresión de incertidumbre tratada en la literatura corresponde fundamentalmente a la expresión de la misma por conjuntos difusos y en menor medida a

[14]Rives y Yuan y Shaw consideran la incertidumbre sólo en las clases de los individuos.

[15]En esta Memoria se considera la entropía difusa (véase Randami, 1994, Séchet, 1995) la que se aplica a probabilidades difusas (Zadeh, 1968).

distribuciones de posibilidad[16] (Rives[17], 1990, Yuan y Shaw, 1995), distribuciones de probabilidad (Quinlan[18], 1990, Araya, 1995, Périnel, 1996) y expresada según la teoría de la evidencia (Shlien, 1990).

Los árboles de Segmentación con incertidumbre son tratados en la literatura por:

- Séchet (Séchet, 1995) presenta la incertidumbre con predictores de categorías difusas y clases difusas, es decir, los individuos tienen un grado de pertenencia a cada una de las categorías de los predictores y a las clases dadas por la variable Z. Las funciones de pertenencia utilizadas para las categorías de los predictores son trapezoidales. Los cortes definidos por un predictor definen tantos nodos hijos como categorías tiene y las ramas se representan por los conjuntos difusos de las categorías que las definen. El grado de pertenencia de un individuo a un nodo resulta de la combinación mediante una función g de los grados de pertenencia del individuo a las categorías que definen los cortes que definen el nodo. Séchet propone que la función g sea el mínimo. Esta función de combinación de valores de relación puede ser cualquier $T-$norma[19] (véase definición 1.9).

  En cada nodo, la estimación de las probabilidades de las clases se realiza estimando la *probabilidad difusa* de las mismas, introducida por Zadeh[20] (Zadeh, 1968). Las frecuencias del caso probabilista son sustituidas por grados de pertenencia a las clases difusas. La probabilidad de cada clase se

[16] Se distingue entre una variable descrita por distribuciones de posibilidad y las categorías difusas de las variables (véase 1.3.4). Como la función de pertenencia de un conjunto difuso es una distribución de posibilidad, también se usa la teoría de posibilidades en presencia de categorías difusas (Dubois et al, 1991, Maher y St. Clair, 1993).

[17] Rives opera con las distribuciones de posibilidad marginales de las clases difusas de Z y con una incertidumbre aplicada a distribuciones de posibilidad.

[18] Quinlan utiliza distribuciones de probabilidad únicamente en la fase de predicción.

[19] o una $T-$norma normalizada en el conjunto de datos E (véase Dubois y Prade, 1989).

[20] Probabilidad de conjuntos difusos. La probabilidad difusa permite asociar a la representacion difusa un grado de incertidumbre representado por una distribución de probabilidad.

Zadeh establece un modelo probabilístico asociado a un experimento aleatorio cuyos sucesos elementales tienen grados de pertenencia a un conjunto difuso A. La probabilidad de un suceso

estima ponderando el grado de pertenencia de los individuos a la clase por sus grados de pertenencia al nodo.

El criterio de máximización es el de *máxima reducción de la entropía difusa condicionada* adaptada al caso difuso, es decir, maximizar:

$$-\sum_{\tilde{k}} q^{*\tilde{k}} \sum_{i=1,...,s} p_i^{*\tilde{k}} \log_2(p_i^{*\tilde{k}}) \tag{2.35}$$

con:

- $\tilde{k}$ nodo hijo en un paso del algoritmo.
- $q^{*\tilde{k}} = \sum_{e \in E} q_{t_{\tilde{k}}}^{\min}(e)$ el peso o la *cardinalidad difusa* del nodo $t_{\tilde{k}}$.
- $p_i^{*\tilde{k}} := P^*(c_i|t_{\tilde{k}}) = \frac{\sum_{e \in E} q_{t_{\tilde{k}}}^{\min}(e) q_{c_i}(e)}{\sum_{e \in E} q_{t_{\tilde{k}}}^{\min}(e)}, i = 1, ..., s$, la *probabilidad difusa* estimada de la clase c_i en el nodo $t_{\tilde{k}}$.
- $q_{c_i} : E \longrightarrow [0,1]$ función de pertenencia a la clase c_i.
- $q_{t_{\tilde{k}}}^{\min} : E \longrightarrow [0,1]$ función[21] que aplica el mínimo a los niveles de relación de un individuo con las ramas que definen el nodo.

En el nodo inicial, la estimación de la probabilidad difusa de una clase es el cociente de la cardinalidad difusa de la misma y el cardinal de E, es decir, $p_i^* = \frac{\sum_{e \in E} q_{c_i}(e)}{n}, i = 1, ..., s$. La entropía difusa inicial es $-\sum_i p_i^* \log_2(p_i^*)$.

- Verde (Verde, 1995) utiliza también como criterio la *máxima reducción de la entropía difusa condicionada*, para obtener árboles binarios. Los predictores

difuso A es:

$$Pr^*(A) = \sum_{e \in E} \mu_A(e) p(e)$$

con $p(.)$ la ley de probabilidad y $\mu_A(.)$ la función de pertenencia.

[21] Para facilitar la notación en $p_i^{*\tilde{k}}$ y $q^{*\tilde{k}}$, se define $q_{t_{\tilde{k}}}^{\min}$ sobre E cuando en realidad está definida sobre tuplas en $[0,1] \times ..l.. \times [0,1]$ y es $q_{t_{\tilde{k}}}^{\min} \equiv \min$, con l el nivel de profundidad del nodo $t_{\tilde{k}}$. Los elementos de una tupla son los niveles de relación de un elemento e con las ramas que definen el nodo $t_{\tilde{k}}$.

originales son variables continuas, siendo introducida la incertidumbre en los mismos mediante dos conjuntos difusos cuyas funciones de pertenencia son la función logística $\phi(x) = \frac{1}{1+e^{\frac{-(x-a)}{b}}}$, obtenidos a y b para cada predictor con la muestra diseño y su complementaria $1 - \phi(x) = \frac{1}{1+e^{\frac{x-a}{b}}}$.

Ahora las probabilidades difusas de las clases en los nodos $\tilde{k}$ son estimadas por $p_i^{*\tilde{k}} = \frac{\sum_{e \in c_i} \phi_{t_{\tilde{k}}}(Y(e))}{\sum_{e \in E} \phi_{t_{\tilde{k}}}(Y(e))}, i = 1, ..., s$, siendo Y el vector de predictores y $\phi_{t_{\tilde{k}}} \circ Y : E \longrightarrow [0, 1]$ función que aplica el producto a los niveles de relación (o grados de pertenencia) de un individuo con las ramas que definen el nodo $t_{\tilde{k}}$.

- Rives (Rives, 1990) propone como criterio la *máxima reducción de una función de información transferible*, asociada a la medida de entropía *incertidumbre-U* definida sobre distribuciones de posibilidad (Klir y Folger, 1988). Los datos de partida son predictores categóricos monoevaluados y clases representadas por conjuntos difusos, con funciones de pertenencia $\phi_{c_1}, ..., \phi_{c_s}$.

  Dado un conjunto de distribuciones de posibilidad $q_i : E \longrightarrow [0, 1]$, $i = 1, ..., s$ la *incertidumbre-U* es una medida de posibilidad de no especifidad que se define por:

$$U(q_1, ..., q_s) = \frac{1}{s} \sum_{i=1,...,s} \{\max_{e \in E}(q_1(e)), ..., \max_{e \in E}(q_s(e))\} \tag{2.36}$$

  Se entiende por *incertidumbre-U* de un predictor Y_j de l_j categorías, el valor $U(q_1^Z, ..., q_{l_j}^Z)$, con q_i^Z la *distribución de posibilidad marginal* de la variable clase Z para los individuos que presentan la categoría $i-$ésima del predictor, es decir:

$$q_i^Z = \{\phi_{c_1/Y_j=i}, ..., \phi_{c_s/Y_j=i}\} = \{\max_{e \in E/Y_j(e)=i} \{\phi_{c_1}(e)\}, ..., \max_{e \in E/Y_j(e)=i} \{\phi_{c_1}(e)\}\} \tag{2.37}$$

La distribución de posibilidad marginal es dada por los grados de pertenencia a cada una de las clases que se obtienen como el máximo de los grados de pertenencia a las clases de los individuos que presentan la categoría correspondiente.

Define la *información transferible* de un nodo hijo $t_{\tilde{k}}$ como la suma de las *incertidumbre-U* de los predictores que no forman parte en la descripción del nodo y de la variable clase Z restada la *incertidumbre-U* de todas ellas conjuntamente (condicionadas al nodo padre todas ellas).

Desde un nodo, se obtienen tantos nodos hijos como categorías tiene el predictor que minimiza el máximo de la *información transferible* de los elementos de la nueva partición, $I(t_{\tilde{k}})$, condicionados por la distribución de posibilidad marginal de las clases estimada en ellos. Este condicionamiento propone que sea mediante la función mínimo, aunque indica que puede ser otra $T-$norma (véase definición 1.9). Es decir, se selecciona el predictor que minimiza:

$$\max_{\tilde{k} \text{ hijos}} I(t_{\tilde{k}})|U(q^Z_{\tilde{k}}) = \max_{\tilde{k} \text{ hijos}} \min\{U(q^Z_{\tilde{k}}), I(t_{\tilde{k}})\} \tag{2.38}$$

por ser el predictor más relacionado con la variable clase. Para más detalles del criterio de maximización, véase Rives, 1990.

En cada nodo, la estimación de la variable clase Z es la distribución de posibilidad marginal de Z para los individuos del nodo. Las reglas de predicción difusas que propone seleccionan la clase que tiene un *grado de distinguibilidad* (Trillas y Sanchís, 1979*)* mayor y le asocian la posibilidad dada por este grado de distinguibilidad, obviando el resto de las clases y sus grados de distinguibilidad. El grado de distinguibilidad de la clase c_i

en el nodo t_k es:

$$D(c_i|t_k) = \frac{\sum_{e \in t_k} |2q_{c_i}(e) - 1|}{\sum_{e \in t_k} q_{c_i}(e)} \tag{2.39}$$

y mide el nivel de distinción de la clase c_i frente al resto de las clases en el nodo t_k.

- Dubois et al. (Dubois et al., 1991) partiendo de descripciones q_P distribución de posibilidad de varios patrones y de q_D distribución de posibilidad de un individuo normalizadas (identificadas con los conjuntos difusos P y D respectivamente), proponen varios criterios de construcción de árboles binarios para asociar al individuo de entrada, uno o varios de los patrones. En uno de los algoritmos propuestos, el individuo puede no pertenecer a ningún nodo o pertenecer a varios nodos que contienen un único patrón. En otros algoritmos propuestos, el individuo pertenece a un único nodo que contiene uno o varios patrones. Definen distintas relaciones de dominio entre las descripciones posibilistas de P y D, utilizando simultáneamente las medidas de *posibilidad* y *necesidad* (véanse definiciones 1.5 y 1.6) definidas sobre dos conjuntos difusos (véase Zadeh, 1978). Dados dos conjuntos difusos P y D referidos sobre E, se define la necesidad de ambos como:

$$N(P, D) = \inf_{e \in E} \max\{q_P(e), 1 - q_D(e)\} \tag{2.40}$$

que mide cuánto es necesario que el conjunto difuso D esté contenido en P.

Asimismo, se define la posibilidad de P y D como:

$$P(P, D) = \sup_{e \in E} \min\{q_P(e), q_D(e)\} \tag{2.41}$$

que mide cuánto de posible es que sea $P \cap D \neq \emptyset$.

La necesidad de dos conjuntos difusos es una medida del grado de inclusión de los valores posibles de uno de ellos (el individuo) en el núcleo del otro (el nodo). La posibilidad de dos conjuntos difusos es una medida de cuánto de posible que ambos se refieran al mismo individuo.

Proponen varios criterios de selección del corte óptimo en función de estas medidas de necesidad y posibilidad.

- Maher y St. Clair (Maher y St.Clair, 1993) presentan el tratamiento de la incertidumbre expresada mediante conjuntos difusos solamente en la fase de predicción si bien los predictores en la fase de construcción del árbol son variables continuas monoevaluadas. En la fase de predicción, realizan una transformación de las ramas del mismo y de los predictores en conjuntos difusos.

  Parten de datos no inciertos o imprecisos y construyen el árbol no necesariamente binario por el criterio de minimización de la entropía, asociando cada nodo del árbol a una única clase, la de mayor probabilidad estimada. Posteriormente, para mejorar las predicciones de nuevos individuos, representan las descripciones de éstos y de los nodos con incertidumbre mediante numeros difusos[22] dotándolos de funciones de pertenencia triangulares.

  Haciendo uso de la teoría de las posibilidades (Zadeh, 1978), introducen la definición de *soporte de la similaridad de dos conjuntos difusos*, como el intervalo $[S_n, S_p] \subseteq [0,1]$ con S_n el soporte necesario (véase (2.42)) y S_p el soporte posible de la similaridad de los dos conjuntos difusos (véase (2.41)). El soporte necesario de dos conjuntos difusos P y D se define como:

$$S_n(P,D) = \min\{N(P,D), N(D,P)\} \tag{2.42}$$

[22] Los *números difusos* son un caso particular de los conjuntos difusos definidos en la recta real (véase Dubois y Prade, 1978).

con $N(.,.)$ definido en (2.40). El soporte de similaridad de dos conjuntos difusos es una extensión de una relación de dominio (véase definición 1.12) entre dos conjuntos difusos cuya imagen es un intervalo de valores, en lugar de un único valor.

Una vez construido el árbol e introducidos los conjuntos difusos, para predecir la clase de un individuo se calcula un intervalo de soporte de similaridad del individuo con cada nodo del árbol, que proporciona una medida del grado de pertenencia del individuo al nodo o de como el nodo *soporta* el individuo.

El grado de pertenencia del individuo a un nodo del árbol se expresa mediante el intervalo que resulta de la combinación mediante una aplicación, denotada por $\wedge^*$, de los intervalos de soporte de similaridad de las descripciones de los cortes que llevan al nodo con las descripciones de los predictores para el individuo. Esta aplicación entre dos intervalos de soporte, se define como:

$$[S_n, S_p] \wedge^* [S'_n, S'_p] = [S_n S'_n, S_p S'_p] \tag{2.43}$$

Además, este *intervalo de soporte* asociado a cada nodo es ponderado por el peso relativo del nodo, multiplicando las cotas inferior y superior del intervalo por el peso relativo del nodo.

La aplicación de combinación $\wedge^*$ presentada en (2.43) es una generalización de la función g de combinación de valores de relación o de adecuación introducida en la definición 1.14, aplicada en este caso a intervalos de valores y no a valores.

El subconjunto de nodos del árbol para los que el *grado de pertenencia* resulta ser distinta del intervalo $[0, 0]$ es el subconjunto de nodos que *soportan* el individuo. Este conjunto de nodos se subdivide en s subconjuntos

dependiendo de las clases asociadas a los nodos. Finalmente, combinando con una aplicación $h_l, l = 1, ..., s$, denotada por $\vee^*$, los intervalos de soporte de los nodos de cada uno de estos s subconjuntos, se obtiene un intervalo de soporte que determina el *intervalo de soporte de la clase* c_l para ese individuo. La aplicación de combinación $\vee^*$ entre dos intervalos de soporte, se define como:

$$[S_n, S_p] \vee^* [S'_n, S'_p] = [S_n + S'_n - S_n S'_n, S_p + S'_p - S_p S'_p] \tag{2.44}$$

El primer término del intervalo de soporte de cada clase es el grado de evidencia presente que soporta la clasificación a la clase y el segundo término del intervalo es el soporte posible de la clasificación a una clase, dada más evidencia.

Una vez obtenidos los intervalos de soporte de cada una de las clases para el individuo, proponen varios criterios de predicción. Por ejemplo, un criterio conservador predice la clase cuya cota inferior es la superior de las cotas inferiores de estos intervalos.

- Quinlan (Quinlan, 1990) introduce la incertidumbre en términos de distribuciones de probabilidad en los predictores en la fase de predicción. Una vez construido el árbol de Segmentación con datos precisos según el criterio de la *ganancia de la razón* (Quinlan, 1988) (véase (2.7)), *etiquetando* cada nodo con una única clase, propone un método para predecir la clase de nuevos individuos con ausencia de observación en los predictores y/o con expresión de incertidumbre en los mismos.

  Para cada individuo $e \in E$ con ausencia de observación en alguno de los predictores, se estima la probabilidad del individuo en cada uno de los nodos del árbol. Esta probabibilidad es nula si los demás predictores imposibilitan el *acceso* al nodo. Para los demás nodos, la probabilidad estimada de

pertenecer a las ramas correspondientes al predictor con ausencia de dato es la distribución marginal condicionada empírica obtenida con los individuos utilizados en la construcción del árbol. Las probabilidades de estos nodos son los productos de las probabilidades de estas ramas que definen el nodo.

Propone estimar las probabilidades para las clases de la variable Z, dado e, como:

$$P_e(c_i) = \sum_{k=1,...,K} P_e(t_k) p_i^k, i = 1, ..., s \tag{2.45}$$

con:

- $P_e(t_k)$ la probabilidad estimada del nodo $k-$ésimo dado e, según se detalla en el párrafo anterior.
- p_i^k la probabilidad estimada de la clase c_i en el nodo $k-$ésimo, dada por la probabilidad empírica.

Quinlan estima asimismo una probabilidad inferior y superior para cada una de las clases.

Propone como mejor aproximación al éxito obtenido en un nodo, la aproximación de Yates (Snedecor y Cochran, 1980) expresada por $\frac{n^k - e^k - 0.5}{n^k}$ o la aproximación de Laplace (Niblett y Bratko, 1986) expresada por $\frac{n^k - e^k - 1}{n^k + 2}$, con e^k el número de errores cometidos con la muestra diseño y n^k el tamaño del nodo (en lugar de la habitual proporción $\frac{n^k - e^k}{n^k}$ que es la probabilidad estimada para la clase que *etiqueta* el nodo). De esta forma, la probabilidad inferior de la clase asignada, la que *etiqueta* al nodo, se estima como $\frac{n^k - e^k - 0.5}{n^k}$.

De forma similar, si para un individuo los predictores se expresan con distribuciones de probabilidad, la asignación a los nodos se realiza de forma probabilista como en el caso de las distribuciones de probabilidad estimadas

en ausencia de observación, multiplicando las probabilidades de las ramas que definen los nodos. De forma similar se realiza la estimación de las clases de estos individuos, dotándolas de probabilidades inferiores y superiores como en el caso de la falta de observación.

- Araya (Araya, 1995) trata la expresión de la incertidumbre en los predictores expresada mediante distribuciones de probabilidad. Estudia el caso de dos clases, $s = 2$, y cortes binarios descritos por datos multievaluados. Propone una generalización del criterio de Kolmogorov-Smirnov que es aplicado por lo general a predictores continuos. El criterio para la realización del corte óptimo desde un nodo consiste en maximizar, en el conjunto de cortes admisibles, la función definida como:

$$KS(t_{k_1}) = |P(t_{k_1}|c_1) - P(t_{k_1}|c_2)| \tag{2.46}$$

con:

  - t_{k_1} uno de los nodos hijos.
  - las probabilidades condicionadas estimadas por:

$$P(t_{k_1}|c_i) = \frac{\sum_{e \in c_i} P_e(t_{k_1})}{card\{c_i\}}, i = 1, 2 \tag{2.47}$$

  - la probabilidad $P_e(t_{k_1})$ calculada a partir de las distribuciones de probabilidad de Y_j, $j = 1, ..., p$ para e, supuesta independencia en las variables.

El máximo de $KS(t_{k_1})$ es *la distancia generalizada de Kolmogorov-Smirnov* y mide el poder discriminante entre las clases c_1 y c_2 del corte que define el nodo hijo t_{k_1}. Se tiene que $KS(t_{k_2}) = KS(t_{k_1})$.

La distancia de Kolmogov-Smirnov mide la distancia máxima entre dos

funciones de distribución y se aplica en Segmentación de variables mono-evaluadas continuas a dos funciones de distribución empíricas. Friedman (Friedman, 1977) propone por primera vez este estadístico como medida de discriminación en árboles de Segmentación obteniendo el predictor y punto de corte óptimos en la maximización. Puede verse una aplicación en Celeux y Lechevallier (1982, 1990). Rounds, 1980 propone también este estadístico.

- Araya (Araya, 1995) propone asimismo, la extensión del criterio de la distancia generalizada de Kolmogorov-Smirmov presentada en (2.46) al caso de que exista, además de la incertidumbre probabilista en los predictores, incertidumbre en las clases, consideradas éstas como conjuntos difusos. En este caso, el criterio de maximización es:

$$KS(t_{k_1}) = |\sum_{e\in E} q_{c_1}(e)\frac{P_e(t_{k_1})}{\sum_{e\in E} q_{c_1}(e)} - \sum_{e\in E} q_{c_2}(e)\frac{P_e(t_{k_1})}{\sum_{e\in E} q_{c_2}(e)}| \qquad (2.48)$$

  siendo para $i = 1, 2$:

  - $q_{c_i} : E \longrightarrow [0, 1]$ la función de pertenencia a la clase c_i.
  - $P_e(t_{k_1})$ calculada a partir de las distribuciones de probabilidad de Y_j, $j = 1, ..., p$ para e, supuesta independencia en las variables.
  - $\sum_{e\in E} q_{c_i}(e)$ la *cardinalidad difusa* o peso de la clase c_i.

  También se cumple en este caso que $KS(t_{k_2}) = KS(t_{k_1})$.

- Périnel (Périnel (1996, 1999)) trata la expresión de la incertidumbre en los predictores expresada mediante distribuciones de probabilidad. Construye árboles binarios descritos por datos multievaluados. El criterio es la maximización de la log-verosimilitud de una mixtura de distribuciones:

$$\log \prod_{e\in E} \sum_{k=1,...,K} P_e(t_k) P_{t_k}(c_e) \qquad (2.49)$$

con $Z(e) = c_e$ y $P_e(t_k)$ el producto de las probabilidades del elemento e de pertenecer a las ramas que definen el nodo. El criterio se convierte en un paso del algoritmo en maximizar:

$$\log \prod_{e \in E} \sum_{\tilde{k}} P_e(t_{\tilde{k}}) P(c_e | t_{\tilde{k}}) \tag{2.50}$$

para los nodos hijos $t_{\tilde{k}}$.

Aplica el algoritmo EM (esperanza-estimación-maximización) (Dempster et al., 1977) para la estimación de las probabilidades $p_i^{\tilde{k}} = P(c_i | t_{\tilde{k}}), i = 1, ..., s$.

- Shlien, 1990 propone medidas de aplicación a datos con incertidumbre expresados según la teoría de la evidencia de Dempster y Shafer (Shafer, 1976).

2.3 Conclusión

En este capítulo se ha introducido la Segmentación y la Segmentación con incertidumbre, dado que ésta será tratada en la parte II de esta Memoria. Si bien se deja para el capítulo 3 la formalización que representa el árbol como un conjunto de objetos simbólicos ya se apunta aquí la representación de los datos y los nodos del árbol como datos y objetos simbólicos.

Se realiza también una recopilación de las referencias más importantes de la Segmentación y del tratamiento de la incertidumbre en Segmentación. Esta última recopilación está basada en la exposición de los criterios adoptados en términos de objetos simbólicos introducidos en el capítulo 1, es decir, se presentan las aproximaciones de la literatura en términos de variables y datos simbólicos y relaciones de dominio, niveles de relación, funciones de combinación de niveles de relación, etc... que serán más desarrollados en la parte II de esta Memoria.

Parte II

Segmentación y Análisis de Datos Simbólicos

Capítulo 3

Segmentación para Datos Estratificados

3.1 Introducción

En el capítulo anterior se introduce la Segmentación y la extensión de los métodos de Segmentación a la presencia de incertidumbre, que afecta en general a las variables explicativas. También se presentan criterios que se han utilizado extensamente en la Segmentación y antecedentes del tratamiento de la incertidumbre en los métodos de Segmentación.

Este capítulo extiende de forma novedosa los métodos de Segmentación a la presencia de estratos en la población de una parte y a la presencia de estratos e incertidumbre en los predictores de otra y se presenta una formalización generalizada del método propuesto, en términos de objetos simbólicos. Esta formalización y el marco común que los datos simbólicos presentan para la representación de la incertidumbre hacen extensible el método a otros tipos de datos simbólicos y, en particular, a otras semánticas de incertidumbre, no tratados en profundidad en esta Memoria.

Frecuentemente, en grandes volumenes de datos, como es el caso de las Ofici-

nas de Estadística, la población no sólo se encuentra dividida en clases conocidas sino que se encuentra estratificada en subpoblaciones. Un estrato no es más que un grupo de individuos que tiene entidad por sí mismo. Éste puede ser el caso de los municipios de una comunidad autónoma, de las comunidades autónomas de un país, de los países de la Unión Europea, sectores económicos de las empresas, ... o la combinación de características socio-demográficas de un conjunto de individuos, por ejemplo.

Los estratos pueden ser agrupaciones de individuos predeterminadas, generalmente de número elevado, o estratos en el sentido más habitual, es decir, considerando un estrato como una subpoblación de la población para la que la explicación de las clases en función de los predictores se espera similar. Es decir, un conjunto de individuos homogéneos en reglas de predicción. Y los datos de partida pueden venir de un muestreo aleatorio simple o bien de un muestreo aleatorio estratificado.

Como se introduce en el prólogo, el objetivo que se persigue con la generalización de los árboles de Segmentación para datos estratificados es triple: explicar o discriminar las clases de los individuos y predecir la clase de nuevos individuos de estratos conocidos; explicar y agrupar los estratos por reglas de predicción de las clases comunes; y, describir agregadamente los estratos o grupos de individuos por objetos simbólicos que representan las reglas de predicción de las clases que les son aplicables.

Además de los datos de partida que puede ser simbólicos, con el método propuesto se consigue uno de los objetivos del Análisis de Datos Simbólicos, que es el de la *generalización* por medio de objetos simbólicos de grupos de individuos. Por una parte, la *generalización de las clases* como en los métodos de Segmentación tradicionales (Gettler-Summa, 1996) y en la Segmentación de datos simbólicos (Périnel (1996,1999)) y por otra la *generalización de los estratos* mediante objetos simbólicos que representan reglas de predicción de una variable de interés en la población. Los objetos simbólicos representan de este modo las propiedades o

descripciones de un elemento genérico del estrato que describen (Gettler-Summa et al., 1994, Chavent, 1996, Gettler-Summa (1992, 1999), Stéphan et al., 2000).

De esta forma, se incorporan a los objetivos de los algoritmos de Segmentación tradicionales, los de describir un estrato por las reglas de predicción de la variable clase y obtener reglas de predicción comunes a conjuntos de estratos, realizándose una clasificación de estratos.

El método de Segmentación generalizado presentado en este capítulo incorpora la estructura de los estratos en el método de Segmentación tradicional. El algoritmo propuesto combina en cada iteración la maximización de una *medida de contenido de información* para la variable clase en una nueva partición binaria (con *incertidumbre, difusa* o tradicional) de la población teniendo en cuenta la pertenencia de los individuos a los estratos; y, la selección de *nodos decisionales*. Cada nodo decisional del árbol, se compone de un conjunto de estratos y una regla de predicción para los individuos de esos estratos para explicar simultaneamente la variable clase.

El marco simbólico encuadra los datos de entrada que pueden venir dados con incertidumbre y ser representados por otros datos simbólicos; que, al igual que en la Segmentación, los nodos del árbol se representan mediante objetos simbólicos; y, que, se obtiene además una descripción simbólica de los estratos o grupos de individuos mediante objetos simbólicos, obteniéndose una generalización de los mismos.

En este capítulo se presenta una formalización general del método propuesto, y para una simplificación en la exposición y evitar formalizaciones excesivas se desciende siempre que sea posible a los casos particulares de los datos de entrada monoevaluados y modales probabilistas que se tratan extensamente en el capítulo 4, si bien también en el capítulo 4 se extiende el método a otros datos simbólicos. En 3.1.1 se presentan los datos de entrada que pueden ser monoevaluados o modales probabilistas. En 3.1.2 se presentan los objetivos del método para los estratos. En 3.2 se introduce el método: se presenta el modo de representación

del árbol en 3.2.1, los nodos en 3.2.2 y los estratos en 3.2.3, y en 3.2.4 los criterios se dan de forma general. En 3.3, se detalla el método completo de 3.3.1 a 3.3.5. En 3.4, se da una breve conclusión.

3.1.1 Datos de partida

Al igual que en el capítulo anterior, se parte de una población de individuos subdivididos en clases conocidas $\{c_1, ..., c_s\}$ y descritos por un vector de variables explicativas o predictores. Estas variables explicativas son ahora categóricas monoevaluadas o modales probabilistas. La población de partida se compone además de grupos de individuos, los *estratos*. Estos estratos pueden estar compuestos por ejemplo por los habitantes de las regiones de un país, los habitantes de las ciudades de una región, los estudiantes de las escuelas de un sistema educativo, etc...

Sea $\Omega = \{\omega_1, ..., \omega_n\}$ un conjunto de individuos y sea $E = \{S_1, ..., S_m\} \subset \mathcal{P}(\Omega)$ una partición de Ω, es decir, $S_i \subset \Omega$ es un grupo de individuos $\omega \in \Omega$, que representa un estrato de Ω. Sean los individuos $\omega \in \Omega$ descritos por las variables predictoras $Y_1, \ldots, Y_p$, con dominios finitos $\mathcal{Y}_1, ..., \mathcal{Y}_p$, con $\mathcal{Y}_j$=$\{1, ..., l_j\}$ y la variable clase Z con dominio finito $\mathcal{Z} = \{1, ..., s\}$. La pertenencia de un individuo $\omega \in \Omega$ a un estrato S_i se caracteriza por $M(\omega) = i \iff \omega \in S_i$. La variable M se llama *variable estrato*.

Se consideran dos tipos diferentes de variables de entrada:

1. Ω *un conjunto de datos monoevaluados.* Las variables Y_j y Z son variables categóricas monoevaluadas (véase definición 1.1) de tal forma que $Y_j(\omega) = l_x$ si y solo si la descripción de ω (dada por Y_j) es la categoría x-ésima del dominio $\mathcal{Y}_j$; y $Z(\omega) = l$ si y solo si el individuo ω pertenece a la clase c_l.

- En este caso, $\mathcal{Y}_1 \times ... \times \mathcal{Y}_p \times \mathcal{Z} \times \mathcal{M}$ es el conjunto de descripciones de individuos de Ω.

2. Ω *un conjunto de datos modales probabilistas.* Las variables Y_j son variables modales probabilistas (véase definición 1.3) de dominio finito $\mathcal{Y}_j$, de tal forma que $Y_j(\omega) = q_{j\omega}$ representa una distribución de probabilidad sobre los elementos de $\mathcal{Y}_j$; y la variable Z es una variable categórica monoevaluada (véase definición 1.1). Se puede pensar en este caso que los elementos $\omega \in \Omega$ son *objetos de segundo nivel* y pueden representar un conjunto de individuos, con información agregada acerca de las observaciones en $\mathcal{Y}$ de un vector de variables monoevaluadas. Esta información agregada es la descripción probabilista de un vector de variables modales probabilistas, que representa la variabilidad en el conjunto. O bien, que los elementos $\omega \in \Omega$ son datos individuales y su descripción probabilista representa una incertidumbre.

- En este caso, $\mathcal{M}^{\mathrm{Prob}}(\mathcal{Y}_1) \times ... \times \mathcal{M}^{\mathrm{Prob}}(\mathcal{Y}_p) \times \mathcal{Z} \times \mathcal{M}$ es el conjunto de descripciones de individuos de Ω.

La matriz de datos de entrada es $[X^Y|X^Z|X^M]$ asociada al conjunto de individuos Ω y al vector de variables (Y, Z, M) de filas $(Y^{(i)}|Z^{(i)}|M^{(i)})$, $i = 1, .., n$ con datos monoevaluados o simbólicos. Aunque en lo sucesivo los elementos $\omega \in \Omega$, se llamen individuos, éstos pueden representar información agregada de grupos de individuos (véase la matriz (1.26) del ejemplo 1.2 y el ejemplo 3.1). En la fase de predicción, se parte de una matriz de datos $[X^Y|X^M]$ asociada al conjunto de individuos Ω_2 y referida al vector de variables (Y, M) y tales que para $\omega \in \Omega_2$, $M(\omega) \in \{1, ..., m\}$.

Ejemplo 3.1 Ω ***un conjunto de datos modales probabilistas.*** *Sea un conjunto de individuos dividido en el conjunto de ciudades de un país que son los estratos. Sea la variable clase Empleo que define las clases empleado y no empleado, la variable que se desea explicar. Sean Sexo, Profesión y Empresa los predictores (variables modales probabilistas) con los cuales se desea explicar*

el Empleo. Un individuo $\omega \in \Omega$ de entrada puede ser:

$$\omega : Sexo(\omega) = mujer, Profesión(\omega) = (mecanógrafa\, 0.8, publicidad\, 0.2),$$
$$Empresa(\omega) = (pública\, 0.6, privada\, 0.4), Empleo(\omega) = si,\ ciudad(\omega) = León \tag{3.1}$$

El individuo representado en (3.1) puede representar dos situaciones diferentes:
- un individuo de León cuyos valores de profesión, empresa y empleo han sido observadas en diversas ocasiones en los últimos años. En este caso, (3.1) representa una mujer de León empleada (en todas las observaciones) que ha trabajado como mecanógrafa el 80% de su vida laboral y en publicidad el 20% y además el 60% de su vida laboral ha trabajado en una empresa pública y el 40% en una empresa privada,
- o bien, un conjunto de individuos de León que son, por ejemplo del mismo distrito, y todos ellos mujeres empleadas. En este caso, (3.1) representa un conjunto de mujeres empleadas de León, de las cuales el 80% son mecanógrafas y el 20% trabajan en publicidad, y de ellas el 60% trabaja en una empresa pública y el 40% en una empresa privada.

3.1.2 Estratos y Objetivos

Se extienden aquí las ideas introducidas en el prólogo y en 3.1. El objetivo que se persigue es asociar los estratos con las reglas de predicción de la variable clase y clasificar los estratos por reglas de predicción comunes, determinando cómo los estratos influyen en las reglas de predicción, ya que algunas son aplicables a unos estratos y no a otros. Los objetivos son:

- Predecir y explicar la clase o el valor de la variable clase de un individuo por los predictores, condicionado al estrato al que pertenece.

- Explicar cómo estas explicaciones o predicciones se ven afectadas por la

pertenencia a un estrato.

- Obtener conjuntos de estratos donde esta explicación de la variable clase es la misma.

- Describir simbólicamente un estrato por el conjunto de reglas que le son válidas, junto con su importancia relativa.

La variable estrato es una variable sobre la que se tiene un interés específico en explicar dado que representa grupos de población que tienen entidad por sí mismos. Por lo general son variables con un número elevado de categorías. Estos grupos pueden ser:

- Sectores de actividades económicas, según la clasificación oficial de la Unión Europea desde 1990, NACE. Es una variable taxonómica con 22 categorías principales. Pueden verse dos aplicaciones en Bravo y García-Santesmases, 2000b y Bissdorf, 2000.

- Sectores de actividad profesional, según la clasificación oficial de la Unión Europea, IFSCO. Es una variable taxonómica con 8 categorías principales.

- Unidades territoriales: Comunidad Autónoma, región, provincia, municipio, etc... Pueden verse dos aplicaciones en Iztueta y Calvo, 2000 y Goupil et al., 2000.

Así mismo la estratificación puede venir dada por combinación de varias variables, como por ejemplo la combinación de características socio-demográficas de un conjunto de individuos como en los procesos de estratificación habituales. En Iztueta y Calvo, 2000 puede verse una aplicación de estos cruces combinados con las variables sexo, categorías de edad, nivel educativo y relación con la actividad laboral.

Aunque las dos situaciones anteriores pudieran ser las más comunes, una variable de estratificación puede ser cualquiera que se tenga interés en explicar

y, en particular, puede ser muy útil para variables con un número elevado de categorías.

Ejemplo 3.2 ***Los estratos son actividades económicas o sectores NACE.*** *En este ejemplo se describe el tipo de datos de una encuesta llevada a cabo entre individuos de distintas empresas e individuos desempleados. Las empresas se clasifican según diversas actividades económicas siguiendo la clasificación europea NACE. Para los individuos desempleados se recoge el dato de la actividad económica de la última empresa que ha trabajado. El objetivo del estudio es explicar el desempleo en la población con respecto a las variables observadas en los individuos: sexo, edad, estudios, profesión... y relacionar esta explicación con la actividad económica de la empresa. En este caso,* Ω *es el conjunto de personas observado,* $E \subset \mathcal{P}(\Omega)$ *contiene conjuntos de personas que trabajan en la misma actividad económica (NACE),* M *la variable que indica la pertenencia de un individuo a una actividad económica (NACE),* Z *recoge la información de empleado o desempleado y* $Y_1, \ldots, Y_p$ *recogen información acerca de sexo, edad, estudios, profesión, etc...*

En este caso, el objetivo es:

- *Obtener la explicación (en los individuos) del desempleo por los predictores sexo, edad, estudios, profesión... condicionado por la actividad económica (NACE) de la empresa en la que trabajan o han trabajado por última vez.*

- *Obtener grupos de actividades económicas (NACE) donde esta explicación es la misma.*

- *Describir una actividad económica (NACE) por el conjunto de reglas que se pueden aplicar para el desempleo en esa actividad económica (NACE) con la importancia que tienen en dicha actividad.*

Ejemplo 3.3 ***Los estratos son municipios.*** *Consideremos el caso de una encuesta llevada a cabo en un conjunto de municipios para conocer el grado de apreciación o satisfacción global que la población tiene acerca de su municipio con respecto a algunos aspectos concretos de apreciación o satisfacción parcial. En este caso,* Ω *es el conjunto de personas encuestadas, los elementos de* $E \subset \mathcal{P}(\Omega)$ *contienen personas del mismo municipio,* M *es la variable que indica la pertenencia de un individuo a un municipio,* Z *mide la apreciación global del municipio,* $Y_1, \ldots, Y_p$ *las apreciaciones parciales de algunos aspectos del municipio, como espacio y habitabilidad, integración, apreciación del trabajo, de la salud, optimismo social, etc ...*

En este caso, el objetivo es:

- *Explicar la apreciación global del municipio por los individuos derivada de las apreciaciones parciales y condicionada por el estrato al que pertenecen.*
- *Obtener conjuntos de municipios para los que la explicación de la apreciación global viene derivada por unas mismas apreciaciones parciales.*
- *Describir un municipio por la explicación general dada a la apreciación global, en función de las apreciaciones parciales.*

3.2 Método y representación

La incorporación de la estratificación en los datos de entrada y la finalidad de describir la información agregada de los estratos por objetos simbólicos, hacen que el objetivo ahora no sea sólo discriminar, estimar o predecir la variable clase Z en función de los predictores mediante búsqueda de dependencias lógicas entre los predictores y las clases como en los árboles de Segmentación, sino además discriminar y describir simbólicamente los estratos por las reglas de predicción de la variable clase que les son aplicables. Es decir, el método de Segmentación

incorpora ahora la información de los estratos en todos los pasos del algoritmo.

En cada iteración del método recursivo, se realiza el corte binario en la población definido por el predictor que maximiza el *contenido de información extendido* del árbol (de la partición) con respecto a las clases y a los estratos, dando lugar a dos nodos hijos *explorables.* El corte en la población se describe por un evento booleano o probabilista definido en un predictor. El *contenido de información extendido* mide la calidad de predicción para la variable clase en una nueva partición, teniendo en cuenta la pertenencia de los individuos a los estratos.

Una vez realizado el corte, se comprueba para subconjuntos de estratos la calidad de predicción de las clases en los nodos hijos (o el contenido de información del nodo con respecto a las clases en los estratos) para obtener *nodos decisionales*. Los estratos con *alta* calidad de predicción se escinden de los nodos explorables para crear uno o varios nodos decisionales. Un nodo decisional es un nodo terminal para algunos estratos, mientras los otros estratos (los que quedan en el nodo explorable) siguen el método recursivo. Para estos últimos estratos, se comprueba una *condición de parada.* La descripción de los estratos que se encuentran en un nodo se realiza mediante un evento booleano definido en la variable estrato M.

La estimación de la variable clase Z en los nodos se realiza a partir de la matriz de entrada $[X^Y|X^Z|X^M]$ y se representa por un evento definido en la variable clase Z. La medida de *contenido de información* del árbol con respecto a las clases mide la calidad de predicción de las clases por el árbol.

El proceso recursivo expuesto se repite para los nuevos nodos explorables si estos son no vacíos después de aplicar la condición de nodo decisional y la condición de parada para los estratos.

En cada iteración del algoritmo, los nodos terminales del árbol son los elementos de la nueva partición que se compone de nodos decisionales y nodos explorables. Al final del algoritmo, el árbol o partición final se compone de todos

los nodos decisionales.

En el capítulo anterior se presentan antecedentes de los árboles de Segmentación y del tratamiento de la incertidumbre en los algoritmos de Segmentación.

3.2.1 Árbol

A diferencia de los árboles de Segmentación expuestos en la literatura, el árbol para datos estratificados tiene nodos decisionales que no son nodos terminales del proceso de partición recursiva, sino que se separan de nodos intermedios del árbol porque algunos estratos verifican la *condición de nodo decisional*. Es decir, los nodos explorables se obtienen del proceso de partición binaria recursiva y son subceptibles de continuar el proceso recursivo, mientras que los nodos decisionales se obtienen de los nodos explorables y no continúan el proceso recursivo (Bravo y García-Santesmases (1997,1998)).

Un árbol de decisión puede representarse por un *conjunto organizado de aserciones* que representan los nodos del árbol. Esta sección presenta la formalización de este modo de representación ya introducido previamente por Ciampi et al. (1993, 1994, 1996), Périnel (1996, 1999), Bravo y García-Santesmases (1997, 1998, 2000a) y Bravo (Bravo, 2000b) y que ya se indicaba en 2.1.5 y 2.2.2.

Conjuntos de aserciones para los nodos

La representación de los nodos se realiza como *conjunción* de tres aserciones o eventos: una aserción definida en el vector de predictores Y, un evento booleano definido en la variable estrato M y un evento definido sobre la variable clase Z. A continuación se definen los conjuntos que contienen estas aserciones y en 3.2.4 se establecen los criterios que afectan a los tipos de descripciones, relaciones de dominio y funciones de combinación de niveles de relación de los elementos de estos conjuntos.

Sea B el conjunto de *elementos posibles de partición* compuesto por eventos

simbólicos definidos en las variables predictoras $Y_1, \ldots, Y_p$ (una descripción más detallada puede verse en 3.2.4). Sea

$$\mathcal{B} = \{\beta = \wedge_l \beta^l : \Omega \longrightarrow [0,1], \beta^l \in B, l = 1, \ldots, l_\beta, l_\beta \in \{1, \ldots, Card(B)\}\} \tag{3.2}$$

el conjunto de aserciones compuesto por conjunciones de eventos de B, con una función de combinación de niveles de relación. Elementos de $\mathcal{B}$ representan los nodos en el vector de predictores y definen las particiones del conjunto Ω definidas según las variables predictoras. Las definiciones 1.26 y 1.27 describen la conjunción de aserciones. Una caracterización de los elementos de $\mathcal{B}$ se describe en 3.2.4 en la nota 3.2.

Sea $\mathcal{A}$ un conjunto de eventos simbólicos definidos en Z :

$$\mathcal{A} = \{\alpha = [Z\mathcal{R}'d] : \Omega \longrightarrow [0,1], d \in \mathcal{D}(\mathcal{Z})\} \tag{3.3}$$

que representan los nodos en la variable clase Z, $\mathcal{D}(\mathcal{Z})$ un espacio de descripciones asociados al dominio $\mathcal{Z}$.

Y sea $\mathcal{N}$ un conjunto de eventos simbólicos booleanos definidos en la variable estrato M:

$$\mathcal{N} = \{\mu = [M \in S] : \Omega \longrightarrow \{0,1\}, S \subseteq \{1, \ldots, m\}\} \tag{3.4}$$

que representan los nodos en la variable estrato M. La función de combinación de niveles de relación que se establece entre los elementos de $\mathcal{N}$ y los elementos de $\mathcal{B}\hat{\wedge}\mathcal{A}$ es el producto de los niveles de relación.

Representación del árbol

El árbol puede representarse por el conjunto organizado de aserciones:

$$T = \{t_k\}_{k=1,\dots,K} = \{\beta_k \wedge \alpha_k \wedge \mu_k\}_{k=1,\dots,K} \tag{3.5}$$

con K el número de nodos terminales, que son decisionales en la última iteración del algoritmo. Cada nodo decisional k se describe por la aserción:

$$t_k = \beta_k \wedge \alpha_k \wedge \mu_k \in \mathcal{B}\hat{\wedge}\mathcal{A}\hat{\wedge}\mathcal{N} \tag{3.6}$$

con:

- $\beta_k \in \mathcal{B}$ una aserción definida en el vector de predictores Y de la forma $\beta_k = \wedge_l \beta_k^l, \beta_k^l \in B, l = 1, \dots, l_k$ siendo l_k el nivel de profundidad del nodo k.
- $\alpha_k \in \mathcal{A}$ un evento simbólico que describe la predicción para la variable clase Z.
- $\mu_k \in \mathcal{N}$ un evento Booleano en la variable estrato M.

Los objetos simbólicos β_k, α_k, μ_k son las descripciones del nodo k en los conjuntos de aserciones $\mathcal{B}$, $\mathcal{A}$ y $\mathcal{N}$ respectivamente. Estas aserciones se denotan por $t_k(\mathcal{B})$, $t_k(\mathcal{A})$ y $t_k(\mathcal{N})$, respectivamente. Así,

$$t_k = t_k(\mathcal{B}) \wedge t_k(\mathcal{A}) \wedge t_k(\mathcal{N}) \tag{3.7}$$

es la descripción del nodo k en $\mathcal{B}\hat{\wedge}\mathcal{A}\hat{\wedge}\mathcal{N}$.

Definición 3.1 ***Nivel de relación de un individuo con un nodo del árbol.*** *El nivel de relación del individuo $\omega \in \Omega$ con el nodo t_k es el nivel de relación del individuo ω con la aserción $\beta_k \wedge \mu_k$, es decir, $t_k(B\hat{\wedge}N)(\omega) = \beta_k \wedge \mu_k(\omega)$ (véase definición 1.19).*

Es decir, es el nivel de relación de la descripción de ω en $\mathcal{D}(\mathcal{Y}) \times \mathcal{M}$ con la descripción de $\beta_k \wedge \mu_k$ en $\mathcal{D}'(\mathcal{Y}) \times \mathcal{P}(\mathcal{M})$[1], siendo $\mathcal{D}'(\mathcal{Y})$ el conjunto de las descripciones de las aserciones de B.

El modo de representación del árbol mediante (3.5) es una generalización de la representación del árbol presentada en el capítulo anterior en (2.1) para datos de entrada monoevaluados y de su extensión para datos con incertidumbre en 2.2.2. La función $I_k(Y)(.)$ definida de Ω en $[0,1]$, se sustituye ahora por la aserción $\beta_k \wedge \mu_k$ y la estimación de la variable clase γ_k en el elemento k de la partición, por el evento α_k.

La predicción de la variable clase Z o del evento que la define para un conjunto de individuos Ω_2, es una generalización de (2.2) como en 2.2.2 y se expresa por la aplicación R:

$$\begin{array}{rccl} R: & \Omega_2 & \longrightarrow & \mathcal{A} \\ & \omega & \longmapsto & \sum\limits_{k=1,\ldots,K} \beta_k \wedge \mu_k(\omega)\alpha_k \end{array} \tag{3.8}$$

$\beta_k \wedge \mu_k(\omega)$ define el nivel de relación del individuo ω con $\beta_k \wedge \mu_k$ y representa una ponderación del nodo k dado $\omega \in \Omega$. Esta ponderación puede ser la pertenencia o no de ω al nodo k (para datos de entrada monoevaluados) o una probabilidad (véase (4.37) para datos de entrada modales probabilistas). Se puede extender a otros tipos de datos y representar una probabilidad difusa o un grado de pertenencia (véase capítulo 2 y extensiones en 4.8).

La predicción de la variable clase Z puede ser también una generalización de (2.28) expresada por:

$$\begin{array}{rccl} R: & \Omega_2 & \longrightarrow & \mathcal{A} \\ & \omega & \longmapsto & h(\{(\beta_k \wedge \mu_k(\omega), \alpha_k)\}_{k=1,\ldots,K}) \end{array} \tag{3.9}$$

[1] En esta Memoria, para datos de entrada monoevaluados, $\mathcal{D}(\mathcal{Y}) = \mathcal{Y}$, para datos de entrada modales probabilistas, $\mathcal{D}(\mathcal{Y}) = \mathcal{M}(\mathcal{Y})$; y, en ambos casos $\mathcal{D}'(\mathcal{Y}) = \mathcal{P}(\mathcal{Y})$.

Es decir, se aplica una aplicación $h(.)$ a las estimaciones de la variable clase en los nodos teniendo en cuenta las ponderaciones de los nodos o los niveles de relación del individuo ω con los nodos del árbol. Un ejemplo de regla de predicción puede ser elegir la predicción del nodo de mayor ponderación.

3.2.2 Nodos del árbol

Esta sección describe al igual que se hizo en 2.1.5 y 2.2.2 los nodos del árbol, destacando ahora la formalización de éstos como aserciones y la incorporación de la información de los estratos a las mismas. La aserción que representa un nodo contiene la descripción de los cortes en las variables predictoras, un subconjunto de estratos y una estimación de la variable clase en el nodo. Esta descripción junto con las relaciones correspondientes constituyen la *intención* del nodo. La *extensión* de la aserción se compone de los individuos que se relacionan a un cierto nivel con la descripción de la aserción según las *relaciones* especificadas en la misma. Un nodo del árbol es:

1. Un elemento de la partición del conjunto de individuos. En el caso de incertidumbre en los predictores, esta partición puede ser difusa, con incertidumbre o normal.

2. Un elemento de la partición (del conjunto de individuos) de los estratos y por tanto un subconjunto de individuos de un subconjunto de estratos. En el caso de incertidumbre en los predictores, esta partición en el conjunto de individuos puede ser difusa, con incertidumbre o no debido a los predictores; si bien no lo es en el conjunto de estratos.

3. Un vector de descripciones del conjunto de descripciones $\mathcal{D}(\mathcal{Y}_1) \times ... \times \mathcal{D}(\mathcal{Y}_p) \times \mathcal{P}(\mathcal{M})^2$ (véase 1.3.5 y definición 1.3.2) asociado al vector de va-

[2]En la aproximación de esta Memoria para datos monoevaluados y modales probabilistas, $\mathcal{D}(\mathcal{Y}_j) = \mathcal{P}(\mathcal{Y}_j)$, $j = 1, ..., p$. Sin embargo, para destacar la generalidad del método, se elige el

riables (Y, M). Más especificamente, es un vector de descripciones (d, D_M) de un conjunto de descripciones $\mathcal{D}(\mathcal{Y}_{k_1}) \times ... \times \mathcal{D}(\mathcal{Y}_{k_l}) \times \mathcal{P}(\mathcal{M})$, asociado al vector de variables $(Y^k, M) = (Y_{k_1}, ..., Y_{k_l}, M)$ con $k_j \in \{1, ..., p\}$ y l nivel de profundidad del nodo.

4. Una aserción $(\beta_k \wedge \mu_k, \mathcal{R}\times \in, (d, D_M))$ definida en Ω. La aserción $\beta_k \wedge \mu_k$ es una función de Ω en $[0, 1]$ que para cada individuo $\omega \in \Omega$ proporciona el nivel de relación $\beta_k \wedge \mu_k(\omega)$ del individuo con el nodo.

5. Se le asocia una descripción simbólica d' de un conjunto de descripciones asociado al dominio $\mathcal{Z}$. Esta descripción representa el valor de predicción para la variable Z en el nodo.

6. Una aserción $(\beta_k \wedge \alpha_k \wedge \mu_k, \mathcal{R} \times \mathcal{R}'\times \in, (d, d', D_M))$ definida en Ω.

7. Un objeto simbólico de tipo regla del tipo: Si $\beta_k \wedge \mu_k$ entonces α_k. Las extensiones de la regla anterior son las extensiones de la aserción $\beta_k \wedge \alpha_k \wedge \mu_k$ y por tanto se identifican ambos objetos simbólicos. Este objeto simbólico representa una regla de predicción para un conjunto de estratos. Para los estratos representados por μ_k : si β_k entonces α_k.

Se puede decir que un nodo es una *aserción (booleana, difusa o probabilista)* (véase definiciones 1.20 y 1.19) asociado al vector de variables (Y^k, M) cuya descripción es dada en el punto (3) y dotado de un vector de relaciones (véase punto (4)).

- En el caso de ser una aserción booleana, la *extensión* (véase definición 1.21) está expresada en el punto (1). El peso del nodo es el cardinal de esta extensión.

término $\mathcal{D}(\mathcal{Y}_j)$. En el capítulo 2, en 2.2.2, se presentan algunos antecedentes con conjuntos de descripciones $\mathcal{D}(\mathcal{Y}_j) = \mathcal{M}(\mathcal{Y}_j)$, $j = 1, ..., p$. Y, también en el capítulo 4, cuando se presentan extensiones del método a otros datos simbólicos.

- En el caso de ser una aserción difusa o probabilista, los individuos $\omega \in \Omega$ tienen un nivel de relación *o adecuación* (se puede pensar en términos de *pertenencia* o de *probabilidad*) con el nodo. En este caso, el peso del nodo es la suma de los niveles de *relación* de todos los individuos ω con el nodo, es decir, de las descripciones de los individuos ω con la descripción del nodo según las relaciones dadas en el punto (4).

El punto (5) asocia con el nodo una descripción monoevaluada o simbólica de un conjunto de descripciones asociado al dominio $\mathcal{Z}$.

Ejemplo 3.4 ***Un nodo decisional del ejemplo de los municipios (ejemplo 3.3).*** *Para los datos de entrada del ejemplo 3.3, la descripción de un nodo decisional puede ser:*

$$\begin{aligned} \beta \wedge \alpha \wedge \mu = ([Y_4 = (-)] \wedge [Y_5 = (-)]) \wedge \\ [Z \sim ((-)0.91, (+)0.09)] \wedge [M \in \{1, 19\}] \end{aligned} \tag{3.10}$$

describe para los individuos de los municipios S_1 y S_{19} la regla: si los valores de Y_4 y Y_5, dos aspectos de apreciación parcial de su municipio, son negativos entonces la probabilidad estimada para una apreciación global negativa del municipio es 0.91 y la probabilidad estimada para una apreciación global positiva es 0.09. Una descripción más completa de las salidas de este ejemplo puede verse en 4.9.3.

Ejemplo 3.5 ***Nodo decisional en el ejemplo de los sectores económicos (ejemplo 3.2).*** *Un nodo decisional puede ser:*

$$\begin{aligned} \beta \wedge \alpha \wedge \mu = [sexo = f] \wedge [salh25 = sí] \wedge [administrativo \sim (no(0.10), sí(0.90))] \\ \wedge [NACE \in \{servicios, electric\}] \end{aligned}$$

que para las unidades de datos en los sectores económicos $NACE$ servicios y electricidad, gas y agua, proporciona la regla: si sexo es mujer y salario bruto

hora medio en el primer cuartil, entonces la probabilidad de ser administrativo es 0.9*. Detalles de los datos y salidas de este ejemplo pueden verse en 4.9.1.*

3.2.3 Estratos

Una vez obtenido el árbol de Segmentación para datos estratificados, se describe la información agregada de los estratos, por las reglas que les son aplicables, así como por los pesos respectivos. Es decir, cada estrato se representa por un conjunto de objetos simbólicos que describen diferentes 'segmentos' de población descritos por los valores de sus variables (predictores y variable clase) así como los pesos de estos segmentos en la población del estrato (véase Bock y Diday, 2000b). Se presenta así un modo de generalización de grupos de individuos, los estratos, por conjuntos de objetos simbólicos con pesos respectivos.

Cada estrato se describe por un conjunto organizado de aserciones ponderadas (Bravo y García-Santesmases (1998, 2000a, 2000b)). Sea T el árbol de (3.5) y sea el estrato $S_i \in E$, entonces la descripción simbólica del mismo es:

$$S_i : \{w_k^i \ (\beta_k \wedge \alpha_k)\}_{k=1,...,K} \tag{3.11}$$

donde:

- $w_k^i \in [0,1]$ representa el peso del nodo decisional k en el estrato S_i;
- β_k es la descripción del nodo decisional k en $\mathcal{B}$ y representa un segmento de población del estrato S_i;
- α_k es la descripción del nodo decisional k en $\mathcal{A}$ y representa la estimación de la variable clase en el segmento de la población de S_i descrito por β_k.

Además, se verifica que para todos los estratos $S_i \in E$, $i \in \{1, ..., m\}$ se tiene que $\sum_{k=1}^{K} w_k^i = 1$.

Ejemplo 3.6 *En este ejemplo, los estratos son el tamaño de la empresa. Se desea caracterizar los profesionales manuales por predictores relacionados con el salario, tiempo de trabajo, sexo, etc..., derivada esta caracterización del tamaño de la empresa en la que trabajan. La descripción de las empresas de tamaño de 100 a 249 trabajadores se representa por un conjunto de objetos simbólicos ponderados:*

$$e100_249 : \{0.2[b50 = no] \wedge [manual = sí 0.16, no 0.84],$$
$$0.48[b50 = sí] \wedge [b25 = sí] \wedge [manual = sí 0.90, no 0.10],$$
$$0.12[b50 = sí] \wedge [b25 = no] \wedge [sexo = f] \wedge [manual = sí 0.10, no 0.90],$$
$$0.2[b50 = sí] \wedge [b25 = no] \wedge [sexo = h] \wedge [manual = sí 0.67, no 0.33]\}$$

que se traduce en las siguientes reglas: Si los bonos periódicos recibidos son mayores que la mediana, entonces el trabajador es no manual $(1 - p = 0.84)$[3]*, si los bonos periódicos son menores que el primer cuartil, entonces es un trabajador manual* $(p = 0.9)$*. Y, para el resto, es decir los que reciben bonos periódicos de cuantía entre el primer cuartil y la mediana, si el trabajador es mujer, entonces no es manual* $(1 - p = 0.9)$ *y si el trabajador es hombre, entonces es manual* $(p = 0.67)$*. Además, la importancia relativa de estas reglas en las empresas de tamaño de 100 a 249 trabajadores es de 0.2, 0.48, 0.12 y 0.2, respectivamente. Una descripción detallada del conjunto de datos del que se extrae este ejemplo puede consultarse en 4.9.1.*

3.2.4 Criterios

Los criterios que deben establecerse son los mismos que en 2.1.6 y 2.2.2, adaptados a la presencia de estratos y a la semántica de incertidumbre en el caso de datos modales probabilistas.

[3] *p: probabilidad estimada para trabajador manual*

Para formalizar estos criterios se definen previamente los conjuntos $\mathcal{T}$ y X. El conjunto $\mathcal{T}$ está compuesto de la sucesión de las particiones o árboles en las iteraciones sucesivas del proceso recursivo:

$$\mathcal{T} = \{T = \{t_k\}_{k=1...K} | T \text{ es un árbol en una iteración del algoritmo}\} \quad (3.12)$$

y el conjunto X se compone de los nodos explorables en una iteración del algoritmo, es decir, de los nodos terminales del árbol que deben seguir siendo explorados en sucesivas iteraciones del algorimo, por no haber satisfecho todos los estratos, las condiciones de nodo de nodo decisional o nodo terminal. En realidad, $r \in X \Leftrightarrow t_r \in T \in \mathcal{T}$ es nodo explorable.

Nota 3.1 *Para las funciones $Adm_\nu(.)$, $Decon_\Gamma(.)$ y $Stop_\tau(.)$ que se presentan en esta subsección con imagen en el conjunto $\{0,1\}$, el valor nulo se identifica con el valor $falso$ y el valor unidad con el valor verdad.*

Los criterios, adaptados aquí a la formalización del árbol mediante aserciones, son:

Criterio 1: **Un conjunto B de *elementos posibles de partición* o cortes posibles.**

Los cortes posibles se definen en las variables predictoras Y_j. La selección de cortes posibles es independiente de las descripciones de los individuos $\omega \in \Omega$ dada por Z y debe realizarse al principio del algoritmo. Los elementos de B son eventos definidos sobre las variables predictoras Y_j:

$$B = \{b_j = [Y_j \mathcal{R}_j d_j] : \Omega \longrightarrow [0,1], d_j \in \mathcal{D}(\mathcal{Y}_j)\} \quad (3.13)$$

Por ejemplo, en el caso de predictores monoevaluados suelen ser de la forma $b_j = [Y_j \in D_j]$ o $b_j^c = [Y_j \notin D_j]$, con $D_j \subset \mathcal{Y}_j$.

Nota 3.2 ***Caracterización de los elementos de $\mathcal{B}$.*** *Con respecto al conjunto $\mathcal{B}$ definido en (3.2)*[4]*, en un plano formal un elemento $\beta \in \mathcal{B}$ está definido por*

$$\beta = [f(Y)\, f(\mathcal{R})\, f(d)] \tag{3.14}$$

con $Y = (Y_1, ..., Y_p)$ el vector de predictores, $d \in \mathcal{D}(\mathcal{Y}) = (\mathcal{D}(\mathcal{Y}_1), ..., \mathcal{D}(\mathcal{Y}_p))$ una descripción y $\mathcal{R} = \mathcal{R}_1 \times ... \times \mathcal{R}_p$ la relación producto con la función de combinación de niveles de relación $g(.)$ de (3.2) y $f(.)$ una aplicación de filtro en los índices j que selecciona un subconjunto $J = \{j_1, ..., j_u\} \subseteq \{1, ..., p\}$ y permite permutaciones en los índices j de J (Bock y Diday, 2000b), es decir,

$$\beta = [f(Y)\, f(\mathcal{R})\, f(d)] = [(Y_{j_1}, ..., Y_{j_u})\mathcal{R}_{j_1} \times ... \times \mathcal{R}_{j_u}(d_{j_1}, ..., d_{j_u})] \tag{3.15}$$

En lo sucesivo, la descripción de un nodo en $\mathcal{B}$ refiere este modo de representación, aunque no se haga uso de la aplicación f en la representación.

La representación en (3.15) es equivalente a la representación alternativa que considera la aserción $\beta \in \mathcal{B}$ definida por:

$$\beta = [Y\mathcal{R}d] \tag{3.16}$$

cuando $\mathcal{D}(\mathcal{Y}) = \mathcal{Y}$ o $\mathcal{D}(\mathcal{Y}) = \mathcal{P}(\mathcal{Y})$ (véase 1.3.5) y donde:

- *Se obvia el órden de los predictores Y_j en la definición de la rama.*
- *Se considera que los eventos definidos sobre los predictores Y_j con $j \in \{1, ..., p\} - J$ son de la forma $[Y_j\mathcal{R}\mathcal{Y}_j]$.*

Proposición 3.1 *$[f(Y)\, f(\mathcal{R})\, f(d)]$ (de (3.15)) y $[Y\mathcal{R}d]$ (de (3.16)) son equivalentes.*

[4] *Se suponen resueltas las conjunciones de elementos de B que hacen referencia a un mismo predictor Y_j, según la definición 1.27. Por tanto, se supone definida la interseccion en $\mathcal{D}(\mathcal{Y}_j)$. En particular para $\mathcal{D}(\mathcal{Y}_j) = \mathcal{P}(\mathcal{Y}_j)$, $j = 1, ..., p$.*

Demostración. El orden de los eventos no afecta el nivel de relación que los mismos tienen sobre los individuos $\omega \in \Omega$, ya que las funciones de combinación de niveles de relación $g(.)$ son simétricas (véase definición 1.14).

Por (1.46), los eventos de la forma $[Y_j \mathcal{R}_j \mathcal{Y}_j]$ dan nivel de relación 1 para todos los individuos $\omega \in \Omega$.

Por (1.51), es equivalente aplicar la función de combinación de niveles de relación a un conjunto de valores que al mismo conjunto de valores aplicado un número arbitrario de elementos unidad.

Por tanto, el resultado de la proposición se sigue de forma trivial. ■

Criterio 2: **Una *condición de admisibilidad.***

Se debe establecer una condición de admisiblidad para que los elementos posibles de partición $b \in B$ puedan ser explorados desde un nodo explorable $r \in X$ como cortes admisibles. Esta condición de admisibilidad se establece mediante la función binaria Adm_ν:

$$\begin{array}{rll} Adm_\nu: & \mathcal{B}\hat{\wedge}\mathcal{N} \times B & \longrightarrow \{0,1\} \\ & (\beta_r \wedge \mu_r, b) & \longmapsto Adm_\nu(\beta_r \wedge \mu_r, b) \end{array} \tag{3.17}$$

Si bien esta función se define en $\mathcal{B}\hat{\wedge}\mathcal{N} \times B$, sólo se aplicará a nodos explorables $r \in X$.

Criterio 3: **Una medida de *contenido de información* y una medida de *contenido de información extendida.***

Una medida de *contenido de información* del árbol con respecto a Ω, $IC\{T, \Omega\}$. IC mide la calidad de predicción para la variable clase con respecto a los predic-

tores y a los estratos.

$$\begin{array}{rll} IC: & \mathcal{T} & \longrightarrow R \\ & T & \longmapsto IC\{T, \Omega\} \end{array} \tag{3.18}$$

Una medida de *contenido de información extendida* $EIC\{T, r, b, E\}$ del árbol T explorado el nodo explorable $r \in X$ con el corte $b \in B_{r,\nu}$ y con respecto a E. EIC mide la calidad de predicción para la variable clase en un árbol con un nuevo corte, teniendo en cuenta la pertenencia a los estratos en el corte.

$$\begin{array}{rll} EIC: & \mathcal{T} \times \mathcal{B}\hat{\wedge}\mathcal{N} \times B \times \mathcal{P}(\Omega) & \longrightarrow R \\ & (T, \beta_r \wedge \mu_r, b, E) & \longmapsto EIC\{T, \beta_r \wedge \mu_r, b, E\} \end{array} \tag{3.19}$$

Si bien esta función se define en $\mathcal{T} \times \mathcal{B}\hat{\wedge}\mathcal{N} \times B \times \mathcal{P}(\Omega)$, sólo se aplicará a nodos explorables $r \in X$. Por comodidad en la notación, se identifica $EIC\{T, r, b, E\}$ con $EIC\{T, \beta_r \wedge \mu_r, b, E\}$.

Criterio 4: Descripción de la estimación de la variable clase.

Un conjunto $\mathcal{A}$ de objetos simbólicos definidos sobre la variable Z y una aplicación $Calc_\alpha(.)$ que devuelve para una descripción de un nodo en $\mathcal{B}\hat{\wedge}\mathcal{N}$, un elemento de $\mathcal{A}$ como la estimación para Z en el nodo. Los objetos simbólicos de $\mathcal{A}$ son las predicciones para la variable clase Z en los nodos del árbol T. Esta aplicación es:

$$\begin{array}{rll} Calc_\alpha: & \mathcal{B}\hat{\wedge}\mathcal{N} & \longrightarrow \mathcal{A} \\ & \beta_k \wedge \mu_k & \longmapsto Calc_\alpha(\beta_k \wedge \mu_k) \end{array} \tag{3.20}$$

Criterio 5:* Una *condición de nodo decisional

Una *condición de nodo decisional*, $Deccon_\Gamma(\beta_k \wedge \mu_k, S_i)$, para un nodo k y un estrato $S_i \in E$ para que pertenezca éste a un nodo decisional.

$$\begin{array}{rll} Deccon_\Gamma: & \mathcal{B}\hat{\wedge}\mathcal{N} \times E & \longrightarrow \{0,1\} \\ & (\beta_k \wedge \mu_k, S_i) & \longmapsto Deccon_\Gamma(\beta_k \wedge \mu_k, S_i) \end{array} \tag{3.21}$$

Si bien esta función se define en $\mathcal{B}\hat{\wedge}\mathcal{N} \times E$, sólo se aplicará a nodos explorables $r \in X$.

***Criterio 6:* Una *condición de parada* para los estratos**

Una condición de parada, $Stop_\tau(\beta_r \wedge \mu_r, S_i)$, para un estrato $S_i \in E$ desde un nodo explorable $r \in X$.

$$\begin{array}{rll} Stop_\tau: & \mathcal{B}\hat{\wedge}\mathcal{N} \times E & \longrightarrow \{0,1\} \\ & (\beta_k \wedge \mu_k, S_i) & \longmapsto Stop_\tau(\beta_k \wedge \mu_k, S_i) \end{array} \tag{3.22}$$

Si bien esta función se define en $\mathcal{B}\hat{\wedge}\mathcal{N} \times E$, sólo se aplicará a nodos explorables $r \in X$.

Asimismo, como en 2.2.2 (pag. 107) deben establecerse los siguientes criterios:

- Las *relaciones de dominio* que se establecen entre las descripciones de los individuos en los predictores y las descripciones de los cortes. Estas relaciones establecen los niveles de relación de los individuos con los cortes.

- Una *función* $g(.)$ *de combinación de niveles de relación* que establece el nivel de relación de los individuos con los nodos, partiendo de los niveles de relación de los individuos con los cortes que definen cada nodo y la pertenencia o no a los estratos que definen los nodos.

- Una aplicación $h(.)$ que proporciona un valor de predicción para un elemento

$\omega \in \Omega$, en función de los niveles de relación del elemento ω con los nodos del árbol y las predicciones estimadas en los mismos.

Además, debe establecerse el criterio de descripción de la información agregada de los estratos por objetos simbólicos.

En el capítulo 4, se particularizan criterios y medidas de 3.2.4 para datos de entrada monoevaluados en 4.1 y modales probabilistas en 4.2. Y se extienden a otros datos simbólicos en 4.8.

3.3 Algoritmo

Esta sección detalla el algoritmo generalizado de árbol de Segmentación para datos estratificados.

El proceso recursivo consiste en realizar para cada nodo explorable la sucesión de los siguientes pasos: Comprobar la *condición de admisibilidad* de los cortes posibles desde el nodo, obtener el corte de entre los admisibles que maximice la *medida de contenido de información extendida*, realizar el corte dando lugar a otros dos nuevos nodos explorables, escindir de estos nodos, aquellos estratos que cumplen la *condición de nodo decisional* y aplicar la *condición de parada* a los estratos que quedan en los dos nuevos nodos explorables. El proceso recursivo termina cuando no existen nodos explorables con cortes admisibles. Un esquema del algoritmo puede verse en Bravo y García-Santesmases, 2000b y una versión previa del algoritmo en Bravo y García-Santesmases, 2000a.

En la presentación del algoritmo, el orden por el cual se van procesando los nodos explorables es aquel que maximiza la *medida de contenido de información* del árbol, si bien el árbol obtenido final es equivalente si se procesan los nodos explorables en cualquier orden.

3.3.1 Inicialización

Se describe el paso inicial del algoritmo que consiste en la *inicialización y evaluación de la medida de contenido de información del árbol* en el paso inicial.

$$\text{Sean} \qquad K = 1,\ S^K = \{1, ..., m\},\ X = \{1\} \tag{3.23}$$

S^1 es el conjunto de indicadores de estrato en el nodo inicial y la descripción multievaluada de la variable estrato M del nodo inicial.

El árbol en el paso inicial se describe por:

$$T = \{\beta_1 \wedge \alpha_1 \wedge \mu_1\} \tag{3.24a}$$

$$\text{con} \qquad \beta_1 = \wedge_j [Y_j \mathcal{R}_j \mathcal{Y}_j],\ \mu_1 = [M \in S^1],\ \alpha_1 = Calc_\alpha(\beta_1 \wedge \mu_1) \tag{3.24b}$$

El árbol inicial contiene un único nodo, explorable, que contiene toda la población y todos los estratos. El primer nodo es la aserción total en el conjunto Ω (véase definición 1.23).

El valor de la medida de información en el paso inicial es:

$$IC\{\beta_1 \wedge \alpha_1 \wedge \mu_1, \Omega\} \tag{3.25}$$

Sea N el conjunto de elementos de $\mathcal{B}\hat{\wedge}\mathcal{A}\hat{\wedge}\mathcal{N}$ que representan los nodos obtenidos en el proceso recursivo. Al final del proceso, N se compone de todos los nodos decisionales y todos los nodos explorados en alguna iteración del proceso. En el paso inicial, este conjunto es:

$$N = \{\beta_1 \wedge \alpha_1 \wedge \mu_1\} \tag{3.26}$$

3.3.2 Admisibilidad

Se describe el paso 1 del algoritmo consistente en la *evaluación de la condición de admisibilidad y actualización del conjunto de nodos explorables.*
Para cada nodo explorable $r \in X$:

Se comprueba la condición de admisibilidad para los elementos de B y se obtiene el conjunto $B_{r,\nu} \subseteq B$ de elementos admisibles de partición para ser explorados desde el nodo r que cumplen la condición de admisibilidad. Sea

$$B_{r,\nu} = \{b \in B | Adm_\nu(\beta_r \wedge \mu_r, b) = 1\} \tag{3.27}$$

el conjunto de cortes admisibles desde el nodo $r \in X$.

Un nodo que no tiene cortes admisibles es un *nodo* que no sigue el proceso recursivo y es *terminal*. Por tanto, se actualiza el conjunto de nodos explorables X eliminando los nodos que no tienen cortes admisibles, es decir, si $B_{r,\nu} = \emptyset$ entonces $X \leftarrow X - \{r\}$.
Fin

Nota 3.3 *La comprobación de la condición de admisibilidad para que los elementos de B sean cortes admisibles desde los nodos explorables, se realiza en cada iteración de algoritmo para los nuevos nodos explorables que se obtienen en la iteración anterior.*

Nota 3.4 *Al introducir nuevos criterios de parada del algoritmo en 4.6.2, se introduce asimismo una mejora en el algoritmo cuando un nodo deja de ser explorable. Esta mejora consiste en la posible escisión del nodo en dos nodos terminales. Por claridad en la exposición en esta sección, se remite al lector a la nota 4.3.*

Si el conjunto de nodos explorables queda vacío entonces se termina el proceso recursivo.

- Si $X = \emptyset$ entonces ***Salida del algoritmo*** al ser el conjunto de nodos explorables el conjunto vacío, $\emptyset$.

- Si $X \neq \emptyset$, entonces $K \leftarrow K + 1$. Es decir, se incrementa en uno el número de nodos del árbol, ya que se realiza la división en dos de uno de los nodos explorables en 3.3.3.

3.3.3 Maximización

Se describe el paso 2 del algoritmo consistente en la *maximización de la medida de contenido de información extendida.* Para cada nodo explorable, se obtiene el corte, de entre los admisibles, que maximiza la medida de contenido de información extendida del árbol con respecto a los cortes admisibles desde el nodo. De entre estos nodos se selecciona el que maximiza estas medidas, obteniéndose el nodo y el corte optimos. Se actualiza el conjunto de nodos explorables eliminando el nodo seleccionado y se crean dos nodos hijos que añaden a la descripción del nodo padre, la descripción del corte óptimo.

Para todos los nodos explorables $r \in X$, sea

$$\beta'_r = \arg \max_{b \in B_{r,\nu}} \{EIC\{T, r, b, E\}\} \tag{3.28}$$

el corte desde el nodo r que maximiza en $B_{r,\nu}$ la medida de contenido de información extendida, $EIC\{T, r, b, E\}$, del árbol T desde el nodo r con el corte admisible $b \in B_{r,\nu}$ con respecto a E. Se considera $\arg\max\{.\}$ la aplicación que devuelve el argumento donde se alcanza el máximo de la función que encierra.

Dados el nodo $r \in X$ y el corte $b \in B_{r,\nu}$, EIC es una medida de contenido de información extendida del árbol que se deriva del árbol T quitando el nodo r, y añadiendo dos nuevos nodos hijos de r definidos en $\mathcal{B}$ por $t_r(\mathcal{B}) \wedge b$ y $t_r(\mathcal{B}) \wedge b^c$,

respectivamente.

$$\text{Sea} \qquad r^+ = \arg\max_{r \in X}\{EIC\{T, r, \beta'_r, E\}\} \tag{3.29}$$

el nodo explorable que maximiza estas medidas.

El nodo r^+ se divide en dos nuevos nodos hijos r_1^+, r_2^+ descritos en B por:

$$\beta_{r_1^+} = \beta_{r^+} \wedge \beta'_{r^+} \tag{3.30a}$$

$$\beta_{r_2^+} = \beta_{r^+} \wedge \beta'^c_{r^+} \tag{3.30b}$$

Para simplificar la notación de los nodos, en la primera iteración del algoritmo, cuando $r^+ = 1$ entonces $\beta_{r_1^+}$, $\beta_{r_2^+}$ son:

$$\beta_{r_1^+} = \beta'_{r^+} \tag{3.31a}$$

$$\beta_{r_2^+} = \beta'^c_{r^+} \tag{3.31b}$$

ya que por la proposición 3.1 se tiene que:

$$\beta_{r_1^+} = \wedge_j [Y_j \mathcal{R} \mathcal{Y}_j] \wedge \beta'_{r^+} \equiv \beta'_{r^+} \tag{3.32a}$$

$$\beta_{r_2^+} = \wedge_j [Y_j \mathcal{R} \mathcal{Y}_j] \wedge \beta'^c_{r^+} \equiv \beta'^c_{r^+} \tag{3.32b}$$

El nodo r^+ deja de ser explorable y se elimina su descripción de T:

$$X \leftarrow X - \{r^+\} \tag{3.33a}$$

$$T \leftarrow T - \{\beta_{r^+} \wedge \alpha_{r^+} \wedge \mu_{r^+}\} \tag{3.33b}$$

3.3.4 Nodos decisionales

Se describe el paso 3 del algoritmo de *obtención de nodos decisionales.* Para los dos nuevos nodos hijos obtenidos en 3.3.3, posibles nodos explorables, se

obtiene el subconjunto de indicadores de estratos para los que la condición de nodo decisional se cumple. Se crean nodos decisionales para estos estratos. Si los nodos hijos queda no vacíos son nodos explorables cuya actualización se realiza en 3.3.5, una vez que se verifica la condición de parada para los estratos restantes. A continuación se detalla este paso.

Para los dos nuevos nodos hijos $k \in \{r_1^+, r_2^+\}$ de 3.3.3, sea:

$$S^k = \{i \in S^{r^+} | Deccon_\Gamma(\beta_k \wedge \mu_{r^+}, S_i) = 1\} \tag{3.34}$$

el conjunto de indicadores[5] de estratos en el nodo r^+ para los cuales la condición de nodo decisional se satisface en el nodo k. Para la descripción de $Deccon_\Gamma$, véase 3.2.4.

Si $S^k = S^{r^+}$, entonces el nodo k pasa a ser decisional. En el resto de los casos, para los estratos en S^k se construye un nodo decisional K y los estratos en $S^{r^+} - S^k$ quedan en un nuevo nodo explorable k. Es decir:

- *Si ningún estrato verifica la condición de decisional, todos los estratos quedan en el nuevo nodo explorable k (que se actualiza en 3.3.5).*

Es decir, si $S^k = \emptyset$ entonces:

$$X \leftarrow X \cup \{k\}$$
$$S^k \leftarrow S^{r^+}$$

Fin

- *Por el contrario, si todos los estratos verifican la condición de decisional, todos pasan a formar parte de un nodo decisional.*

Es decir, si $S^k = S^{r^+}$ entonces:

$$\mu_k = [M \in S^k],\ \alpha_k = Calc_\alpha(\beta_k \wedge \mu_k)$$

[5]El conjunto S^{r^+} es la descripción multievaluada de la variable M del nodo explorable k antes de la escisión de los estratos que cumplen la condición de nodo decisional.

Se añade el nodo decisional k al árbol T y al conjunto de nodos N:

$$T \leftarrow T \cup \{\beta_k \wedge \alpha_k \wedge \mu_k\}$$
$$N \leftarrow N \cup \{\beta_k \wedge \alpha_k \wedge \mu_k\}$$

Fin

- *En cualquier otro caso, se obtiene un nodo decisional K y un nodo explorable k (que se actualiza en 3.3.5).*

Es decir, cuando $S^k \neq S^{r^+}$:

$$K \leftarrow K + 1$$
$$X \leftarrow X \cup \{k\}$$
$$S^K \leftarrow S^k,\ S^k \leftarrow S^{r^+} - S^k$$
$$\mu_K = [M \in S^K],$$
$$\alpha_K = Calc_\alpha(\beta_k \wedge \mu_K)$$

Se añade el nodo K al árbol T y al conjunto de nodos N:

$$T \leftarrow T \cup \{\beta_k \wedge \alpha_K \wedge \mu_K\}$$
$$N \leftarrow N \cup \{\beta_k \wedge \alpha_K \wedge \mu_K\}$$

Fin

Para la descripción de la aplicación $Calc_\alpha$ véase 3.2.4.

3.3.5 Parada

Se describe el paso 4 del algoritmo consistente en la *aplicación de la condición de parada para los estratos.* Para los nodos explorables de 3.3.4 se obtiene el subconjunto de indicadores de estratos para los que la condición de parada se satisface. Se crean nodos terminales para ellos y se actualiza la descripción de los nodos explorables añadiéndose al conjunto de nodos explorables. Finalmente, se calcula la medida de contenido de información del nuevo árbol con respecto a Ω. A continuación se detalla este paso.

Para los nuevos nodos explorables $k \in \{r_1^+, r_2^+\} \cap X$ obtenidos en 3.3.4, se forma el conjunto:

$$R^k = \{i \in S^k | Stop_\tau(\beta_k \wedge \mu_k, S_i) = 1\} \tag{3.35}$$

de indicadores de estratos[6] del nodo k para los que se verifica la condición de parada de los estratos (véase 3.2.4).

Si $R^k = S^k$, entonces el nodo k deja de ser explorable y pasa a ser terminal. En el resto de los casos, para los estratos en R^k se obtiene un nodo terminal K y el nodo explorable k se actualiza eliminando los estratos en R^k que pasan a formar parte del nodo terminal K.

- *Si ningún estrato cumple la condición de parada, se añade el nodo explorable k obtenido en 3.3.4, al árbol T y al conjunto de nodos N.*

Es decir, si $R^k = \emptyset$, entonces:

$$\mu_k = [M \in S^k],\ \alpha_k = Calc_\alpha(\beta_k \wedge \mu_k)$$
$$T \leftarrow T \cup \{\beta_k \wedge \alpha_k \wedge \mu_k\}$$
$$N \leftarrow N \cup \{\beta_k \wedge \alpha_k \wedge \mu_k\}$$

Fin

- *Por el contrario, si todos los estratos cumplen la condición de parada, entonces el nodo k deja de ser explorable y se convierte en terminal.*

Es decir, si $R^k = S^k$ entonces:

$$X \leftarrow X - \{k\}$$
$$\mu_k = [M \in S^k],\ \alpha_k = Calc_\alpha(\beta_k \wedge \mu_k)$$
$$T \leftarrow T \cup \{\beta_k \wedge \alpha_k \wedge \mu_k\}$$
$$N \leftarrow N \cup \{\beta_k \wedge \alpha_k \wedge \mu_k\}$$

Fin

[6] El conjunto S^k es la descripción multievaluada de la variable M del nodo explorable k antes de eliminar los estratos que cumplen la condición de parada.

- *En cualquier otro caso, para los estratos en R^k se obtiene un nodo terminal K y el nodo explorable k se actualiza eliminando los estratos en R^k que pasan a formar parte del nodo terminal K y éstos se añaden al árbol T y al conjunto de nodos N.*

Es decir, si $S^k \supset R^k \neq \emptyset$, entonces:

$$K \leftarrow K + 1$$
$$S^K \leftarrow R^k,\ S^k \leftarrow S^k - R^k$$
$$\mu_k = [M \in S^k],\ \mu_K = [M \in S^K]$$
$$\alpha_k = Calc_\alpha(\beta_k \wedge \mu_k),\ \alpha_K = Calc_\alpha(\beta_k \wedge \mu_K)$$
$$T \leftarrow T \cup \{\beta_k \wedge \alpha_k \wedge \mu_k, \beta_k \wedge \alpha_K \wedge \mu_K\}$$
$$N \leftarrow N \cup \{\beta_k \wedge \alpha_k \wedge \mu_k, \beta_k \wedge \alpha_K \wedge \mu_K\}$$

Fin

Nota 3.5 *Los nodos terminales obtenidos en este paso son nodos decisionales que se obtienen al alcanzarse la condición de parada de los estratos.*

Exploración de otro nodo o final del algoritmo.

Se añade la nueva descripción del árbol a la sucesión $\mathcal{T}$ de árboles obtenidos y se calcula el nuevo valor de medida de contenido de información del árbol con respecto a Ω:

$$\mathcal{T} \leftarrow \mathcal{T} \cup T$$

Se calcula $IC\{T, \Omega\}$

- Si $X \neq \emptyset$ entonces ir a ***3.3.2.***
- si no ***Salida del algoritmo***.

Nota 3.6 *K es el número de nodos no intermedios del árbol, es decir, los decisionales y los explorables. En la iteración final, son los nodos decisionales.*

Nota 3.7 *Por parsimonia en la representación se considera que los nodos de un árbol están indexados de* 1 *a* K *en una iteración del algoritmo. Sea* $o(k)$ *el índice del nodo* $k-$*ésimo de* T*, entonces* $T = \{t_{o(k)}\}_{k=1,\dots,K}$*, aunque se ha considerado, por simplicidad,* $T = \{t_k\}_{k=1,\dots,K}$*.*

3.4 Conclusión

Se presenta en este capítulo una formalización generalizada del método propuesto en términos de objetos simbólicos de tal forma que el algoritmo general puede ser extendido a otros tipos de datos simbólicos sin más que aportar las relaciones de dominio correspondientes. La formalización del método aporta unas medidas o funciones generales que pueden ser definidas de distintas maneras dando lugar a diferentes criterios.

De una parte se crea un nuevo algoritmo de Segmentación generalizado en el que se incorpora el tratamiento de los estratos y de la incertidumbre en los predictores junto a los estratos. Se aporta además una forma novedosa de generalización de grupos de individuos: los estratos; además de la generalización habitual de las clases de los métodos de Segmentación.

De otra parte, se proporciona una formalización básica del algoritmo creándose el marco formal que permite una formalización general del método que lo hace extensible a otros datos simbólicos y otros tipos de expresión de la incertidumbre. Dicha extensión está formalizada en 4.8.

En 4.6 se proponen algunos criterios de parada adicionales y algunas mejoras al algoritmo para los nodos que dejan de ser explorables en una iteración del algoritmo y en 4.7 se proponen algunas extensiones del método.

Capítulo 4

Método Datos Monoevaluados, Modales Probabilistas y Extensiones

En el capítulo anterior se presenta un método de Segmentación para datos estratificados formalizado en forma de objetos simbólicos. En este capítulo se presentan dos aproximaciones al método para datos de entrada monoevaluados en 4.1 y para datos de entrada modales probabilistas en 4.2. En ambas secciones se particularizan los criterios expuestos en 3.2.4 y se incorpora al método el tratamiento de otros datos simbólicos como las reglas de no aplicabilidad. En 4.3 se presentan algunos resultados que caracterizan el árbol para datos estratificados según los criterios de 4.1 y 4.2. En 4.4 se presenta la descripción simbólica de los estratos y normas de interpretación de los mismos, destacando la importancia del método para la caracterización de los estratos y la clasificación de los mismos por reglas de predicción comunes, frente a los árboles de Segmentación tradicionales.

En 4.5 se proporciona un método de predicción que incluye a individuos con algunos predictores de valor desconocido. En 4.6 se presenta la calidad del árbol y se proponen algunos criterios de parada adicionales y algunas modificaciones al

algoritmo para los nodos que dejan de ser explorables en una iteración.

En 4.7 se proponen algunas extensiones del método que incluyen la incorporación de pesos en los individuos y las probabilidades 'a priori' de las clases. En 4.8, los criterios de 3.2.4 se presentan de una forma generalizada que extiende el algoritmo de Segmentación para datos estratificados a otros tipos de datos simbólicos, relaciones de dominio y funciones de combinación de niveles de relación, destacándose la importancia del marco simbólico para la formalización del método. Para distintos tipos de datos simbólicos de entrada, se proponen distintos tipos de aserciones de representación de los nodos, estimaciones de la variable clase y criterios concretos de elección. La extensión comprende también la consideración de una variable clase no monoevaluada.

En 4.9 se presentan algunas aplicaciones del método a modo de ejemplo, destacándose normas de interpretación de resultados y líneas generales de aplicación. Finalmente, en 4.10 se expone una pequeña conclusión al capítulo.

4.1 Criterios para datos monoevaluados

En esta sección, se presentan la elección de los criterios expuestos en 3.2.4 para el método de Segmentación para datos estratificados cuando los datos de entrada son monoevaluados (véase 3.1.1). La sección se estructura según estas elecciones. Véase Bravo y García-Santesmases, 2000a.

4.1.1 Elementos posibles de partición

Este conjunto B se compone de cortes binarios que se representan por eventos booleanos definidos en los predictores $Y_j : \Omega \longrightarrow \mathcal{Y}_j$ (véase 3.1.1, punto (1)) con relaciones de dominio de pertenencia, es decir,

$$B = \{b = [Y_j \in D_j], b^c = [Y_j \in \mathcal{Y}_j - D_j] | D_j \subset \mathcal{Y}_j, j \in \{1, \ldots, p\}\} \qquad (4.1)$$

Proposición 4.1 *El conjunto B se compone de eventos boolenos. Además, dados $\omega \in \Omega$ y $b = [Y_j \in D_j] \in B$, con $D_j \subset \mathcal{Y}_j$ se tiene una de las siguientes igualdades:*

- $b(\omega) = 1 \Longleftrightarrow b^c(\omega) = 0$
- $b(\omega) = 0 \Longleftrightarrow b^c(\omega) = 1$

Además, su evento complementario $[Y_j \in \mathcal{Y}_j - D_j]$ es equivalente al evento $[Y_j \notin D_j]$, por tanto se denotan ambos por b^c.

Demostración. Sean $\omega \in \Omega$ y $b = [Y_j \in D_j], D_j \subset \mathcal{Y}_j, j \in \{1, \ldots, p\}$. Como Y_j es una variable categórica monoevaluada entonces $Y_j(\omega) = l_s^\omega \in \{1, ..., l_j\}$ y por tanto, los únicos dos valores de relación de b con ω son:

$$b(\omega) = [Y_j(\omega) \in D_j] = \begin{cases} 1 \Longleftrightarrow Y_j(\omega) = l_s^\omega \in D_j \\ 0 \Longleftrightarrow Y_j(\omega) = l_s^\omega \notin D_j \end{cases} \tag{4.2}$$

y de b^c con ω son:

$$b^c(\omega) = [Y_j(\omega) \in \mathcal{Y}_j - D_j] = \begin{cases} 0 \Longleftrightarrow Y_j(\omega) = l_s^\omega \in D_j \\ 1 \Longleftrightarrow Y_j(\omega) = l_s^\omega \notin D_j \end{cases} \tag{4.3}$$

Además, es trivial que

$$[Y_j(\omega) \notin D_j] \equiv [Y_j(\omega) \in \mathcal{Y}_j - D_j] \tag{4.4}$$

■

Cuando la rama por la que se realiza un corte se representa por un evento $b \in B$, entonces la otra rama se representa por su complentario b^c.

Corolario 4.1 *Los pares de cortes b, $b^c \in B$ definen una partición en Ω.*

Demostración. Se deduce trivialmente de la proposición 4.1 que la partición definida en Ω es $Ext_\Omega(b)$, $Ext_\Omega(b^c)$. ■

4.1.2 Función de admisibilidad

La función de admisibilidad introducida en (3.17) de un corte posible $b \in B$ desde un nodo explorable $r \in X$, es en general una función que depende de la elección del criterio de parada para los estratos en los nodos hijos del nodo r cuando se realiza el corte. $Adm_\nu(\beta_r \wedge \mu_r, b)$ da el valor 1 ($verdad$) si el corte es admisible y 0 ($falso$) si el corte es no admisible.

- La admisibilidad es nula si y sólo si ocurre alguna de estas condiciones:

1. $\beta_r \wedge b \equiv \emptyset^{\mathcal{B}}$ o $\beta_r \wedge b^c \equiv \emptyset^{\mathcal{B}}$, entendida esta equivalencia como que uno de los eventos de B que representa el nodo, es el evento vacío (véase definición 1.24). Esto ocurre, en la presente formulación del algoritmo, sólo en el caso de que un predictor no binario forme parte de la descripción actual del nodo r y b haga referencia a categorías de ese predictor no incluidas en la descripción del corte β_r.

2. $\min\{Ext_\Omega(\beta_r \wedge b \wedge \mu_r), Ext_\Omega(\beta_r \wedge b^c \wedge \mu_r)\} < \nu$ con $\nu \geq 5$. Es decir, si el peso de uno de los nodos hijos es pequeño. El **peso del nodo** t_k, descrito en $\mathcal{B}\hat{\wedge}\mathcal{N}$ por $\beta_k \wedge \mu_k$ es $Ext_\Omega(\beta_k \wedge \mu_k)$.

3. $Stop_\tau(\beta_r \wedge b \wedge \mu_r, S_i) = 1, \forall i \in S^r$ (véase 4.1.6).

4. $Stop_\tau(\beta_r \wedge b^c \wedge \mu_r, S_i) = 1, \forall i \in S^r$ (véase 4.1.6).

   Es decir, si se verifica la condición de parada de todos los estratos en alguno de los dos nodos hijos del nodo r (condiciones 3. y 4.).

5. Si no hay predictores alternativos. Esta es evidente.

6. Si el corte hace referencia a un predictor que es consecuente de una regla de no aplicabilidad y cuyo antecedente es uno de los predictores que definen el nodo r.

4.1.3 Medidas de contenido de información

Medida de contenido de información

La medida de contenido de información introducida en (3.18) de un árbol $T = \{\beta_k \wedge \alpha_k \wedge \mu_k\}_{k=1,\dots,K}$ del conjunto $\mathcal{T} \subset \mathcal{P}(\mathcal{B}\hat{\wedge}\mathcal{A}\hat{\wedge}\mathcal{N})$ con respecto al conjunto de individuos Ω es (Bravo y García-Santesmases, 1997):

$$\begin{array}{rll} IC: & \mathcal{T} & \longrightarrow R \\ & T & \longmapsto IC\{T,\Omega\} := -\sum_{k=1}^{K} P(\beta_k \wedge \mu_k) Ent(Z|\beta_k \wedge \mu_k) \end{array} \tag{4.5}$$

donde:

- $P(.)$ son las probabilidades estimadas de los nodos del árbol y como es habitual en los árboles de Segmentación, estas probabilidades son las probabilidades empíricas, definidas como:

$$P(\beta_k \wedge \mu_k) = \frac{Card(Ext_\Omega(\beta_k \wedge \mu_k))}{Card(\Omega)} \tag{4.6}$$

  para $\beta_k \wedge \mu_k \in \mathcal{B}\hat{\wedge}\mathcal{N}$.

- $Ent(Z|\beta_k \wedge \mu_k)$ es una medida estimada de la entropía de Shannon (véase (2.8)) para la variable Z en el nodo t_k. En 4.1.4, se estiman las probabilidades correspondientes.

- Las probabilidades de las clases de Z en el nodo t_k son estimadas por las probabilidades empíricas de las clases en el subconjunto de Ω definido por la extensión de la aserción $\beta_k \wedge \mu_k$, es decir, en el subconjunto $Ext_\Omega(\beta_k \wedge \mu_k)$.

Así mismo, se definen las probabilidades:

$$P(S_i|\beta \wedge \mu) \ : \ = P([M = i]|\beta \wedge \mu) \tag{4.7}$$

$$P(\beta \wedge \mu \cap S_i) \ : \ = P(\beta \wedge \mu \cap [M = i]) \tag{4.8}$$

para $\beta \wedge \mu \in \mathcal{B}\hat{\wedge}\mathcal{N}$ e $i \in \{1, ..., m\}$, como las probabilidades empíricas correspondientes. La intersección entre aserciones está 1.5.1.

La definición de la medida de contenido de información, en (4.5), es una medida negativa de la entropía poderada media en los nodos del árbol.

A continuación se generaliza la definición de medida de contenido de información al conjunto $\mathcal{P}(\mathcal{B}\hat{\wedge}\mathcal{A}\hat{\wedge}\mathcal{N}) \times E$, ya que esta medida se utilizará también para árboles de $\mathcal{T}$ a los que se les quita un nodo explorable (véase (4.11)) y para un nodo del árbol con respecto a un estrato (véase (4.15)). La generalización de la medida de información al conjunto $\mathcal{P}(\mathcal{B}\hat{\wedge}\mathcal{A}\hat{\wedge}\mathcal{N}) \times E$ es:

$$\begin{array}{rll} IC: & \mathcal{P}(\mathcal{B}\hat{\wedge}\mathcal{A}\hat{\wedge}\mathcal{N}) \times E & \longrightarrow R \\ & (T = \{\beta_k \wedge \alpha_k \wedge \mu_k\}_{k=1,...,L}, S_i) & \longmapsto IC\{T, S_i\} \end{array} \tag{4.9}$$

con:

$$IC\{T, S_i\} := -\sum_{k=1}^{L} P(\beta_k \wedge \mu_k \cap [M = i]) Ent(Z|\beta_k \wedge \mu_k \cap [M = i]) \tag{4.10}$$

Cuando $i \in \{1, ..., m\}$ es un estrato del nodo t_k, $Ent(Z|\beta_k \wedge \mu_k \cap [M = i])$ es una medida estimada de la entropía de Shannon (véase (2.8)) para la variable Z en el nodo t_k para el estrato S_i. Si $\beta_k \wedge \mu_k \cap [M = i] \equiv \emptyset^{\mathcal{B}\hat{\wedge}\mathcal{N}}$, no se calcula la entropía condicionada correspondiente por ser $P(\emptyset^{\mathcal{B}\hat{\wedge}\mathcal{N}}) = 0$.

Medida de contenido de información extendida

La medida de **contenido de información extendida** introducida en (3.19) del árbol T expandido desde el nodo $r \in X$, con el corte $b \in B_{r,\nu}$ con respecto a E es:

$$\begin{array}{rlll} EIC: & \mathcal{T} \times \mathcal{B}\hat{\wedge}\mathcal{N} \times B_{r,\nu} \times \mathcal{P}(\Omega) & \longrightarrow & R \\ & (T, \beta_r \wedge \mu_r, b, E) & \longmapsto & EIC\{T, \beta_r \wedge \mu_r, b, E\} \end{array} \tag{4.11}$$

se denota $EIC\{T, \beta_r \wedge \mu_r, b, E\} = EIC\{T, r, b, E\}$, con:

$$\begin{aligned} EIC\{T, r, b, E\} &:= IC\{T(r), \Omega\} \\ -P(\beta_r \wedge b \wedge \mu_r) \sum_{i \in S^r} P(S_i|\beta_r \wedge b \wedge \mu_r) Ent(Z|\beta_r \wedge b \wedge [M = i]) \\ -P(\beta_r \wedge b^c \wedge \mu_r) \sum_{i \in S^r} P(S_i|\beta_r \wedge b^c \wedge \mu_r) Ent(Z|\beta_r \wedge b^c \wedge [M = i]) \end{aligned} \tag{4.12}$$

y siendo:

- $T(r) = T - \{\beta_r \wedge \alpha_r \wedge \mu_r\}$, el árbol que resulta de T eliminado el nodo r
- $\mu_r = [M \in S^r]$
- $P(S_i|\beta_r \wedge b \wedge \mu_r)$ y $P(S_i|\beta_r \wedge b^c \wedge \mu_r)$ estiman las probabilidades del estrato $S_i \in E$ (para $i \in S^r$) en los elementos $\beta_r \wedge b \wedge \mu_r$, $\beta_r \wedge b^c \wedge \mu_r \in \mathcal{B}\hat{\wedge}\mathcal{N}$, que representan dos posibles nodos explorables. Cada una de ellas es la proporción de individuos del estrato S_i en los nodos con respecto al total de individuos en los mismos, respectivamente.

  Para $i \in S^r$:

$$\begin{aligned} P(S_i|\beta_r \wedge b \wedge \mu_r) &= \frac{Card(Ext_\Omega(\beta_r \wedge b \wedge [M = i]))}{Card(Ext_\Omega(\beta_r \wedge b \wedge \mu_r))} \\ &= \frac{Card(Ext_{S_i}(\beta_r \wedge b))}{Card(Ext_\Omega(\beta_r \wedge b \wedge \mu_r))} \end{aligned} \tag{4.13}$$

  Para $i \notin S^r$:

$$P(S_i|\beta_r \wedge b \wedge \mu_r) = P([M = i]|\beta_r \wedge b \wedge \mu_r) = P(\emptyset^{\mathcal{B}\hat{\wedge}\mathcal{N}}) = 0 \tag{4.14}$$

  Y de igual modo, $P(S_i|\beta_r \wedge b^c \wedge \mu_r)$.

  Por simplicidad, los sumatorios en (4.11) se referencian a los indicadores de estratos en S^r.

- $Ent(Z|\beta_r \wedge b \wedge [M=i])$, $Ent(Z|\beta_r \wedge b^c \wedge [M=i])$ son las entropías estimadas de la variable clase en el estrato S_i en ambos nodos (véase (4.23) y (4.24) en 4.1.4). Las probabilidades estimadas de las clases en los nodos hijos para el estrato S_i son las probabilidades empíricas correspondientes en $Ext_{S_i}(\beta_r \wedge b)$ y $Ext_{S_i}(\beta_r \wedge b^c)$, respectivamente.

Nota 4.1 *Dado que el término* $\frac{1}{Card(\Omega)}$ *es común en todas las estimaciones de IC y de EIC puede prescindirse de él en las estimaciones correspondientes de ambas medidas, a los efectos de obtención del árbol.*

Proposición 4.2 *Se tiene que:*

$$EIC\{T,r,b,E\} = IC\{T(r),\Omega\} + \sum_{i \in S^r} IC\{\{\beta_r \wedge b \wedge \alpha_r^b \wedge \mu_r, \beta_r \wedge b^c \wedge \alpha_r^{b^c} \wedge \mu_r\}, S_i\} \tag{4.15}$$

con $\alpha_r^b = Calc_\alpha(\beta_r \wedge b \wedge \mu_r)$ *y* $\alpha_r^{b^c} = Calc_\alpha(\beta_r \wedge b^c \wedge \mu_r)$.

Demostración. Basta probar que para $i \in S^r$ que:

$$\begin{gathered} IC\{\{\beta_r \wedge b \wedge \alpha_r^b \wedge \mu_r, \beta_r \wedge b^c \wedge \alpha_r^{b^c} \wedge \mu_r\}, S_i\} = \\ -P(\beta_r \wedge b \wedge \mu_r)P(S_i|\beta_r \wedge b \wedge \mu_r)Ent(Z|\beta_r \wedge b \wedge [M=i]) = \\ -P(\beta_r \wedge b^c \wedge \mu_r)P(S_i|\beta_r \wedge b^c \wedge \mu_r)Ent(Z|\beta_r \wedge b^c \wedge [M=i]) \end{gathered} \tag{4.16}$$

Sea $i \in S^r$, se comprueba fácilmente que:

$$\begin{gathered} -P(\beta_r \wedge b \wedge \mu_r)P(S_i|\beta_r \wedge b \wedge \mu_r)Ent(Z|\beta_r \wedge b \wedge [M=i]) = \\ -P(\beta_r \wedge b \wedge \mu_r)P([M=i]|\beta_r \wedge b \wedge \mu_r)Ent(Z|\beta_r \wedge b \wedge [M=i]) = \\ -P(\beta_r \wedge b \wedge [M=i])Ent(Z|\beta_r \wedge b \wedge [M=i]) \end{gathered} \tag{4.17}$$

De igual modo se comprueba que:

$$-P(\beta_r \wedge b^c \wedge \mu_r)P(S_i|\beta_r \wedge b^c \wedge \mu_r)Ent(Z|\beta_r \wedge b^c \wedge [M=i] =$$
$$-P(\beta_r \wedge b^c \wedge \mu_r)P([M=i]|\beta_r \wedge b^c \wedge \mu_r)Ent(Z|\beta_r \wedge b^c \wedge [M=i]) =$$
$$-P(\beta_r \wedge b^c \wedge [M=i])Ent(Z|\beta_r \wedge b^c \wedge [M=i]) \tag{4.18}$$

De (4.17) y (4.18) se deduce fácilmente (4.16). ■

La segunda parte de $EIC\{T, r, b, E\}$ en (4.15) es una suma del contenido de información de los nodos hijos de r con respecto a cada estrato. Resulta ser el negativo de la suma de la entropía interna ponderada en los estratos de la variable clase en estos nodos.

Se deduce trivialmente de la proposición 4.2 el siguiente corolario por permanecer $IC(T(r), \Omega)$ constante en (4.15) en la maximización.

Corolario 4.2 *Maximizar $EIC\{T, r, b, E\}$ en el conjunto $B_{r,\nu}$ en una iteración del algoritmo (véase 3.3.3) es equivalente a maximizar:*

$$\sum_{i \in S^r} IC\{\{\beta_r \wedge b \wedge \alpha_r^b \wedge \mu_r, \beta_r \wedge b^c \wedge \alpha_r^{b^c} \wedge \mu_r\}, S_i\} \tag{4.19}$$

4.1.4 Descripción de la estimación de la variable clase

El conjunto $\mathcal{A}$ de elementos que describen la estimación de la variable clase en los nodos del árbol, se compone de eventos modales probabilistas para la variable clase Z (véase definición 1.16). Es decir[1],

$$\mathcal{A} = \{[Z \sim (c_1\, p_1, \ldots, c_s\, p_s)] | p_l \in [0,1], \sum_l p_l = 1\} \tag{4.20}$$

con $\sim$ la relación de dominio de (1.63).

[1] En esta notación de los elementos de $\mathcal{A}$ se identifica las categorías $1, ..., s$ con las clases $c_1, \ldots, c_s$.

Para un nodo decisional k, descrito en $\mathcal{B}\hat{\wedge}\mathcal{N}$ por $\beta_k \wedge \mu_k$, la descripción del nodo k en $\mathcal{A}$ es:

$$\alpha_k = Calc_\alpha(\beta_k \wedge \mu_k) = [Z \sim (c_1\, p_1^k, \ldots, c_s\, p_s^k)] \tag{4.21}$$

$$\text{con} \qquad p_l^k = P([Z = l]|\beta_k \wedge \mu_k) = \frac{Card(Ext_\Omega(\beta_k \wedge [Z = l] \wedge \mu_k))}{Card(Ext_\Omega(\beta_k \wedge \mu_k))} \tag{4.22}$$

para $l \in \{1, \ldots, s\}$. El uso del operador $Calc_\alpha(.)$ introducido en (3.20) es una cuestión notacional que facilita la referencia a estas estimaciones de la variable clase en otras partes del capítulo. Las probabilidades de las clases de Z en los nodos decisionales se estiman por las probabilidades empíricas condicionales de las clases a los nodos. La probabilidad para la clase $l \in \{1, ..., s\}$ en el nodo k se estima por la proporción de elementos de la extensión de la aserción $\beta_k \wedge \mu_k$ en Ω que son de la clase c_l.

Por extensión, para un elemento $\beta_k \wedge [M = i] \in \mathcal{B}\hat{\wedge}\mathcal{N}$, con $i \in \{1, \ldots, m\}$:

$$\alpha_k^i = Calc_\alpha(\beta_k \wedge [M = i]) = [Z \sim (c_1\, p_1^{ki}, \ldots, c_s\, p_s^{ki})] \tag{4.23}$$

es la descripción de la variable clase en un nodo del árbol y para un estrato S_i, necesaria en (4.10), (4.12) y (4.15), con:

$$p_l^{ki} = P([Z = l]|\beta_k \wedge [M = i])) = \frac{Card(Ext_\Omega(\beta_k \wedge [Z = l] \wedge [M = i]))}{Card(Ext_\Omega(\beta_k \wedge [M = i]))} \tag{4.24}$$

para $l \in \{1, \ldots, s\}$.

Se deduce fácilmente el siguiente corolario.

Corolario 4.3 *p_l^k en (4.22) es:*

$$p_l^k = \sum_{i \in S^k} P(S_i|\beta_k \wedge \mu_k) P([Z = l]|\beta_k \wedge [M = i])), l \in \{1, \ldots, s\} \tag{4.25}$$

o lo que es lo mismo,

$$p_l^k = \sum_{i \in S^k} P(S_i|\beta_k \wedge \mu_k) p_l^{ki}, \ \textit{para } l \in \{1, \ldots, s\} \tag{4.26}$$

con $\mu_k = [M \in S^k]$, $P(S_i|\beta_k \wedge \mu_k)$ *definido en (4.7) y expresado como (4.13) y (4.14), y* p_l^{ki} *en (4.24).*

4.1.5 Condición de nodo decisional

La condición de nodo decisional para un estrato desde un nodo está en relación con el contenido de información del nodo en el estrato. Para los estratos que verifican la condición de nodo decisional se crean uno o varios nodos decisionales (véase 3.3.4). La condición introducida en (3.21) para que un estrato $S_i \in E$ de un nodo r pertenezca a un nodo decisional se define de la siguiente forma:

$$Deccon_\Gamma(\beta_r \wedge \mu_r, S_i) = 1 \iff IC\{\beta_r \wedge \alpha_r \wedge \mu_r, S_i\} > \Gamma, i \in S^r \tag{4.27}$$

$$\text{con} \qquad \mu_r = [M \in S^r] \tag{4.28}$$

Otro criterio alternativo a (4.27) es:

$$Deccon_{\Gamma'}(\beta_r \wedge \mu_r, S_i) = 1 \iff Ent(Z|\beta_r \wedge \mu_r \cap S_i) < \Gamma', i \in S^r \tag{4.29}$$

Este criterio es similar al anterior pero sin tener en cuenta el peso del nodo y estrato.

Nota 4.2 *En 3.3.4 que describe el paso* 3 *del algoritmo, se asume que a partir de un nodo explorable, sólo se puede obtener un nodo decisional. Se expresa de esta manera, para una mayor claridad en la exposición, ya que no se modifica sustancialmente el algoritmo. En el caso de obtención de varios nodos decisionales desde un nodo, las modificaciones que sufre el algoritmo general son las relativas*

al valor de K o número de nodos decisionales que se incrementa, las descripciones en $\mathcal{A}$ y $\mathcal{N}$ de los nuevos nodos decisionales que se obtienen y la medida de contenido de información del árbol. Las descripciones en $\mathcal{A}$ y $\mathcal{N}$ especifican la estimación de la variable clase en los nodos decisionales y los estratos que pertenecen a dichos nodos y son similares a las especificadas en 3.3.4.

Desde un nodo k (como los de 3.3.4), hijo de un nodo r^+ (de 3.3.3), se pueden obtener varios nodos decisionales, para subconjuntos de estratos que cumplen la condición de nodo decisional, dependiendo de las probabilidades estimadas para las clases para cada uno de estos estratos. Sea S^k de (3.34), el conjunto de indicadores de estratos para los que se satisface la condición de nodo decisional desde k. Sea α_k^i, para cada uno de los estratos $i \in S^k$:

$$\alpha_k^i = Calc_\alpha(\beta_k \wedge [M = i]) = [Z \sim (c_1\, p_1^{ki}, \ldots, c_s\, p_s^{ki})] \tag{4.30}$$

la estimación de la variable clase en $\mathcal{A}$ que correspondería para el nodo k y el estrato S_i.

$$Sea \qquad l_i'^k = \arg \max_{l \in \{1,\ldots,s\}} \{p_l^{ki}\} \tag{4.31}$$

el índice del conjunto $\{1, \ldots, s\}$ donde p_l^{ki} es maxima, es decir, la clase $l_i'^k$ de Z es la clase de mayor probabilidad estimada en el nodo k y el estrato S_i (llegándose a una solución de compromiso en el caso de que coincidieran dos clases).

$$Sea \qquad S_l^k = \{i \in S^k | l_i'^k = l\} \tag{4.32}$$

para $l \in \{1, \ldots, s\}$, el conjunto de índices de estratos de S^k que tienen a l como la clase de Z de mayor probabilidad estimada. Cada conjunto S_l^k, $l \in \{1, \ldots, s\}$ no vacío define un nodo decisional (que se escinde del nodo k) para los estratos en S_l^k. La actualización del árbol con los nuevos nodos decisionales se realiza como en 3.3.4 para el nodo decisional construído para los estratos en S_l^k.

Se comprueba fácilmente que

$$S_l^k \subseteq S^k \text{ para } l \in \{1, \ldots, s\} \tag{4.33a}$$

$$\cup_{l=1}^{s} S_l^k = S^k \tag{4.33b}$$

4.1.6 Condición de parada

La condición de parada introducida en (3.22) desde un nodo k para un estrato $S_i \in E$ se basa en el peso del estrato en el nodo y se define de la siguiente forma:

$$Stop_\tau(\beta_k \wedge \mu_k, S_i) = 1 \iff Ext_\Omega(\beta_k \wedge [M = i]) < \tau, i \in S^k \tag{4.34}$$

siendo $\mu_k = [M \in S^k]$.

La expresión $Ext_\Omega(\beta_k \wedge [M = i])$ es el peso del nodo y estrato.

4.2 Criterios para datos modales probabilistas

En esta sección, se presentan la elección de los criterios expuestos en 3.2.4 para el método de Segmentación para datos estratificados cuando los datos de entrada son modales probabilistas (véase 3.1.1). Véase Bravo y García-Santesmases, 2000a.

4.2.1 Elementos posibles de partición

Este conjunto B se compone de cortes binarios que se representan por eventos probabilistas definidos en los predictores $Y_j : \Omega \longrightarrow \mathcal{M}^{\mathrm{Prob}}(\mathcal{Y}_j)$ (véase 3.1.1, punto 2.) con relaciones de tipo $\sim$, es decir,

$$B = \{b = [Y_j \sim D_j], b^c = [Y_j \sim \mathcal{Y}_j - D_j] | D_j \subset \mathcal{Y}_j = \{1, \ldots, l_j\}, j \in \{1, \ldots, p\}\} \tag{4.35}$$

La definición de los elementos $\beta_j = [Y_j \sim D_j] \in B$ es similar a (1.73). $\beta_j(\omega)$

representa la probabilidad de que $Y_j \in D_j (\in \mathcal{P}(\mathcal{Y}_j))$ dado ω, según la distribución de probabilidad $Y_j(\omega) = q_{\omega,j}$ definida sobre el conjunto $\mathcal{Y}_j$.

Proposición 4.3 *Dado $\omega \in \Omega$, $b \in B$ se cumple que:*

$$b(\omega) + b^c(\omega) = 1$$

Demostración. Sea $\omega \in \Omega$ descrito en la variable Y_j por la descripción probabilista $Y_j(\omega) = (1\, p_1^j, \ldots, l_j\, p_{l_j}^j)$, se tiene que:

$$b(\omega) = [Y_j(\omega) \sim D_j] = [(1\, p_1^j, \ldots, l_j\, p_{l_j}^j) \sim D_j] = \sum_{l \in D_j} p_l^j$$

Y, de igual modo:

$$b^c(\omega) = \sum_{l \in \mathcal{Y}_j - D_j} p_l^j$$

Por tanto

$$b(\omega) + b^c(\omega) = \sum_{l \in D_j} p_l^j + \sum_{l \in \mathcal{Y}_j - D_j} p_l^j = \sum_{l \in \mathcal{Y}_j} p_l^j = 1$$

■

Corolario 4.4 *El corte b y $b^c \in B$ definen una partición con incertidumbre en Ω.*

Demostración. Los niveles de relación $b(\omega)$ y $b^c(\omega)$ representan las probabilidades $\Pr ob(Y \in D_j|\omega)$ y $\Pr ob(Y \notin D_j|\omega)$ respectivamente, según ley de probabilidad $Y_j(\omega) = q_{\omega,j}$. También, $\sum_{\omega \in \Omega} (b(\omega) + b^c(\omega)) = Card(\Omega)$ ■

4.2.2 Función de admisibilidad

La función de admisibilidad se define de la misma forma que para el caso de las variables predictoras monoevaluadas (véase 4.1.2), salvo por la condición 2. que es transformada en:

- 2. $\min\{\sum_{\omega\in\Omega}\beta_r \wedge b \wedge \mu_r(\omega), \sum_{\omega\in\Omega}\beta_r \wedge b^c \wedge \mu_r(\omega)\} < \nu$ con $\nu \geq 5$. Es decir, si el peso de uno de los nodos hijos es pequeño. El **peso del nodo** t_k, descrito en $\mathcal{B}\hat{\wedge}\mathcal{N}$ por $\beta_k \wedge \mu_k$ es $\sum_{\omega\in\Omega}\beta_k \wedge \mu_k(\omega)$. Antecedentes a esta denominación se encuentran en Périnel y Lechevallier, 2000.

4.2.3 Medidas de contenido de información

Cada uno de los elementos de la muestra diseño lleva asociado una descripción probabilista en un conjunto de descripciones $\mathcal{M}^{\mathrm{Prob}}(\mathcal{Y})$, dada por un vector de variables modales probabilistas $Y = (Y_1, ..., Y_p)$. Sea $\omega \in \Omega$, se tiene que la descripción probabilista $Y_j(\omega) = q_{\omega,j}$ representa la distribución de probabilidad de una variable aleatoria en $\mathcal{Y}_j$.

Sea la aserción $\beta \wedge \mu (\in \mathcal{B}\hat{\wedge}\mathcal{N})$ que representa un nodo del árbol en $\mathcal{B}\hat{\wedge}\mathcal{N}$. Según se define $\mathcal{B}$ en (3.2) y según la nota 3.2 (página 145) de caracterización de los elementos de $\mathcal{B}$, éstos se representan por una conjunción de eventos definidos sobre los predictores, donde no hay dos que se refieran al mismo predictor. Sea, por tanto, $\beta \wedge \mu = [(f(Y), M) \sim \times \in (f(D), S^M)]$, con $Y = (Y_1, ..., Y_p)$ y $f(.)$ una aplicación de filtro en los índices j (o una permutación) que selecciona $J \subseteq \{1, \ldots, p\}$, $\sim$ relación producto definida unidimensionalmente como en (1.73) y con función de combinación de niveles de relación la función producto como en (1.93); y, $\in$ relación definida para M. Sea $f(D) = (\{D_j | j \in J\})$ con $D_j \subseteq \mathcal{Y}_j = \{1, \ldots, l_j\}$ y $S^M \subseteq \{1, \ldots, m\}$, entonces $\beta \wedge \mu$ es una aplicación:

$$\beta \wedge \mu = \wedge_{j\in J}[Y_j \sim D_j] \wedge [M \in S^M] : \Omega \to [0, 1] \tag{4.36}$$

tal que dado $\omega \in \Omega$ descrito en la variable Y_j, $j \in J$ por $Y_j(\omega) = (1\, q^j_{\omega,1}, \ldots, l_j\, q^j_{\omega,l_j})$, se tiene que $\beta \wedge \mu(\omega)$ es:

$$\beta \wedge \mu(\omega) = \prod_{j \in J} \sum_{l \in D_j} q^j_{\omega,l} \times [M(\omega) \in S^M] = \begin{cases} \prod_{j \in J} \sum_{l \in D_j} q^j_{\omega,l} & \text{Si } M(\omega) \in S^M \\ 0 & \text{Si } M(\omega) \notin S^M \end{cases} \tag{4.37}$$

$\beta(\omega)$ representa la probabilidad de que $Y_j \in D_j$, dado ω si se cumple la independencia estadística entre las variables Y_j, según ley de probabilidad $Y(\omega) = q_\omega = (q_{\omega,1}, ..., q_{\omega,p})$ definida en $\mathcal{Y}$. Antecedentes a esta relación producto en Segmentación se encuentran en Quinlan, 1990, Ciampi et al. (1993, 1994, 1996), Araya (1995) y Périnel (1996, 1999). En el caso de independencia estadística en los predictores, $\beta \wedge \mu(\omega)$ representa la probabilidad de que $Y_j \in D_j$ y sea de un conjunto de estratos de índice en S^M, según ley de probabilidad $Y(\omega) = q_\omega$ y la pertenencia a un estrato. La complejidad actual de los objetos simbólicos no contempla los objetos simbólicos modales probabilistas definidos para un vector de variables simultáneamente que tome en cuenta las distribuciones de probabilidad conjuntas.

Medida de contenido de información

La medida de contenido de información de un árbol $T = \{\beta_k \wedge \alpha_k \wedge \mu_k\}_{k=1,\ldots,K}$ del conjunto $\mathcal{T}$ con respecto al conjunto de individuos Ω es (Bravo, 2000b):

$$\begin{array}{lll} IC: & \mathcal{T}(\subset \mathcal{P}(\mathcal{B}\hat{\wedge}\mathcal{A}\hat{\wedge}\mathcal{N})) & \longrightarrow R \\ & T & \longmapsto IC\{T,\Omega\} := -\sum_{k=1}^{K} P(\beta_k \wedge \mu_k) Ent(Z|\beta_k \wedge \mu_k) \end{array} \tag{4.38}$$

donde:

- $P(.)$ son las probabilidades estimadas:

$$P(\beta_k \wedge \mu_k) = \sum_{\omega \in \Omega} \beta_k \wedge \mu_k(\omega) \frac{1}{Card(\Omega)} = \sum_{\omega \in \Omega} \beta_k(\omega) \mu_k(\omega) \frac{1}{Card(\Omega)} \quad (4.39)$$

  considerando que todos los individuos $\omega \in \Omega$ tiene el mismo peso. Antecedentes a esta estimación de las probabilidades de los nodos se encuentran en Périnel y Lechevallier, 2000.

- $Ent(Z|\beta_k \wedge \mu_k)$ es una medida estimada de la entropía de Shannon (véase Dubois et al. (1991), Quinlan (1990)) para la variable Z en el nodo t_k. En 4.2.4, se estiman las probabilidades correspondientes.

- Las probabilidades de las clases de Z en el nodo t_k son estimadas por las probabilidades condicionadas de las clases en el nodo (véase (4.51) en 4.2.4).

La igualdad (4.39) resulta ser:

$$P(\beta_k \wedge \mu_k) = \sum_{\omega | M(\omega) \in S^k} \Pr ob(f_k(Y) \in D_k|\omega) \frac{1}{Card(\Omega)} \quad (4.40)$$

con f_k una aplicación de filtro en los índices j (o una permutación) que selecciona $J_k \subseteq \{1, \ldots, p\}$, $D_k = (\{D_{kj}|j \in J_k\})$ con $D_{kj} \subseteq \mathcal{Y}_{kj} = \{1, \ldots, l_{kj}\}$ la descripción de la aserción β_k, $\mu_k = [M \in S^k]$, $S^k \subseteq \{1, ..., m\}$ y $P'(f_k(Y) \in D_k|\omega)$ denota la probabilidad de $f_k(Y) \in D_k$ según la ley de probabilidad $Y(\omega) = q_\omega = (q_{\omega,1}, ..., q_{\omega,p})$.

Así mismo, se estiman las probabilidades:

$$P(S_i|\beta \wedge \mu) \quad : \quad = P([M = i]|\beta \wedge \mu) \quad (4.41)$$

$$P(\beta \wedge \mu \cap S_i) \quad : \quad = P(\beta \wedge \mu \cap [M = i]) \quad (4.42)$$

para $\beta \wedge \mu \in \mathcal{B}\hat{\wedge}\mathcal{N}$, $i \in \{1, ..., m\}$.

La medida de contenido de información en (4.38), se utilizará también para árboles de $\mathcal{T}$ a los que se les quita un nodo explorable (véase (4.43)).

Se generaliza aquí también la definición de la medida de contenido de información al conjunto $\mathcal{P}(\mathcal{B}\hat{\wedge}\mathcal{A}\hat{\wedge}\mathcal{N}) \times E$ de forma análoga a como se hizo en (4.9) y (4.10), sustituidas las $P(.)$ por las probabilidades estimadas según (4.39) y $Ent(Z|.)$ las medidas estimadas de la entropía de Shannon para la variable Z en el nodo t_k y el estrato S_i.

Medida de contenido de información extendida

La medida de **contenido de información extendida** del árbol T expandido desde el nodo $r \in X$, con el corte $b \in B_{r,\nu}$ con respecto a E (un conjunto de clases del conjunto Ω) es:

$$\begin{array}{rlll} EIC: & \mathcal{T} \times \mathcal{B}\hat{\wedge}\mathcal{N} \times B_{r,\nu} \times \mathcal{P}(\Omega) & \longrightarrow & R \\ & (T, \beta_r \wedge \mu_r, b, E) & \longmapsto & EIC\{T, r, b, E\} \end{array} \tag{4.43}$$

con:

$$\begin{aligned} EIC\{T, r, b, E\} &:= IC\{T(r), \Omega\} \\ -P(\beta_r \wedge b \wedge \mu_r) &\sum_{i \in S^r} P(S_i|\beta_r \wedge b \wedge \mu_r) Ent(Z|\beta_r \wedge b \wedge [M = i]) \\ -P(\beta_r \wedge b^c \wedge \mu_r) &\sum_{i \in S^r} P(S_i|\beta_r \wedge b^c \wedge \mu_r) Ent(Z|\beta_r \wedge b^c \wedge [M = i]) \end{aligned} \tag{4.44}$$

y siendo:

- $T(r) = T - \{\beta_r \wedge \alpha_r \wedge \mu_r\}$, el árbol que resulta de T eliminado el nodo r
- $\mu_r = [M \in S^r]$
- $P(S_i|\beta_r \wedge b \wedge \mu_r)$ y $P(S_i|\beta_r \wedge b^c \wedge \mu_r)$ estiman las probabilidades del estrato $S_i \in E$ (para $i \in S^r$) en los elementos $\beta_r \wedge b \wedge \mu_r$, $\beta_r \wedge b^c \wedge \mu_r \in \mathcal{B}\hat{\wedge}\mathcal{N}$, que

representan dos posibles nodos explorables:

$$P(S_i|\beta_r \wedge b \wedge \mu_r) = \frac{\sum_{\omega\in\Omega} \beta_r \wedge b \wedge [M=i](\omega)}{\sum_{\omega\in\Omega} \beta_r \wedge b \wedge \mu_r(\omega)} = \begin{cases} \frac{\sum_{\omega\in S_i} \beta_r \wedge b(\omega)}{\sum_{\omega\in\{S_{i_0}, i_0\in S^r\}} \beta_r \wedge b(\omega)} & \text{Si } i \in S^r \\ 0 & \text{Si } i \notin S^r \end{cases} \quad (4.45)$$

Y de forma similar para $P(S_i|\beta_r \wedge b^c \wedge \mu_r)$.

- $Ent(Z|\beta_r \wedge b \wedge [M=i])$, $Ent(Z|\beta_r \wedge b^c \wedge [M=i])$, $i \in S^r$, son las entropías de la variable clase en ambos nodos y el estrato S_i. Las probabilidades de las clases para un estrato se estiman en (4.53).

De forma equivalente a la proposición 4.2, se deduce la siguiente.

Proposición 4.4 *Se tiene que:*

$$EIC\{T, r, b, E\} = IC\{T(r), \Omega\} + \sum_{i\in S^r} IC\{\{\beta_r \wedge b \wedge \alpha_r^b \wedge \mu_r, \beta_r \wedge b^c \wedge \alpha_r^{b^c} \wedge \mu_r\}, S_i\} \quad (4.46)$$

con $\mu_r = [M \in S^r]$

$$e \quad IC\{\{\beta_r \wedge b \wedge \alpha_r^b \wedge \mu_r, \beta_r \wedge b^c \wedge \alpha_r^{b^c} \wedge \mu_r\}, S_i\} = -P(\beta_r \wedge b \wedge [M=i]) Ent(Z|\beta_r \wedge b \wedge [M=i]) - P(\beta_r \wedge b^c \wedge [M=i]) Ent(Z|\beta_r \wedge b^c \wedge [M=i]) \quad (4.47)$$

para $i \in S^r$

La segunda parte de $EIC\{T, r, b, E\}$ en (4.46) es una suma del contenido de información de los nodos hijos de r con respecto a cada estrato. Resulta ser el negativo de la suma de la entropía interna ponderada en los estratos de la variable clase en estos nodos.

Se deduce fácilmente de la proposición 4.4, el siguiente corolario.

Corolario 4.5 *Maximizar* $EIC\{T, r, b, E\}$ *en el conjunto* $B_{r,\nu}$ *en una iteración del algoritmo es equivalente a maximizar*

$$\sum_{i \in S^r} IC\{\{\beta_r \wedge b \wedge \alpha_r^b \wedge \mu_r, \beta_r \wedge b^c \wedge \alpha_r^{b^c} \wedge \mu_r\}, S_i\} \tag{4.48}$$

4.2.4 Descripción de la estimación de la variable clase

El conjunto $\mathcal{A}$ de elememos que describen la estimación de la variable clase en los nodos del árbol, se compone de eventos modales probabilistas para la variable clase Z (véase definición 1.16). Es decir[2],

$$\mathcal{A} = \{[Z \sim c_1\, p_1, \ldots, c_s\, p_s],\ p_l \in [0,1],\ \sum_l p_l = 1\} \tag{4.49}$$

con $\sim$ la relación de dominio de (1.63).

Para un nodo decisional k, descrito en $\mathcal{B}\hat{\wedge}\mathcal{N}$ por $\beta_k \wedge \mu_k$, la descripción del nodo k en $\mathcal{A}$ es:

$$\alpha_k = Calc_\alpha(\beta_k \wedge \mu_k) = [Z \sim c_1\, p_1^k, \ldots, c_s\, p_s^k] \tag{4.50}$$

$$\text{con} \qquad p_l^k = P([Z = l] | \beta_k \wedge \mu_k) = \frac{\sum_{\omega \in \Omega} \beta_k \wedge [Z = l] \wedge \mu_k(\omega)}{\sum_{\omega \in \Omega} \beta_k \wedge \mu_k(\omega)} \tag{4.51}$$

para $l \in \{1, \ldots, s\}$.

El uso del operador $Calc_\alpha(.)$ es una cuestión notacional que facilita la referencia a estas estimaciones en otras partes del capítulo.

[2]En esta notación de los elementos de $\mathcal{A}$ se identifica las categorías $1, ..., s$ con las clases $c_1, \ldots, c_s$.

Por extensión, para un elemento $\beta_k \wedge [M = i] \in \mathcal{B}\hat{\wedge}\mathcal{N}$, con $i \in \{1, \ldots, m\}$:

$$\alpha_k^i = Calc_\alpha(\beta_k \wedge [M = i]) = [Z \sim c_1\, p_1^{ki}, \ldots, c_s\, p_s^{ki}] \tag{4.52}$$

es la descripción de la variable clase en un nodo del árbol y para un estrato S_i, con:

$$p_l^{ki} = P([Z = l]|\beta_k \wedge [M = i]) = \frac{\sum_{\omega\in\Omega} \beta_k \wedge [Z = l] \wedge [M = i](\omega)}{\sum_{\omega\in\Omega} \beta_k \wedge [M = i](\omega)} \tag{4.53}$$

para $l \in \{1, \ldots, s\}$.

Estas estimaciones son necesarias en (4.44) y (4.46)

Se comprueba fácilmente la siguiente proposición.

Proposición 4.5 *p_l^k en (4.51) es:*

$$p_l^k = \sum_{i\in S^k} P(S_i|\beta_k \wedge \mu_k) P([Z = l]|\beta_k \wedge [M = i])) \tag{4.54}$$

para $l \in \{1, \ldots, s\}$, o lo que es lo mismo,

$$p_l^k = \sum_{i\in S^k} P(S_i|\beta_k \wedge \mu_k)\ p_l^{ki} \tag{4.55}$$

para $l \in \{1, \ldots, s\}$, con $\mu_k = [M \in S^k]$, $P(S_i|\beta_k \wedge \mu_k)$ como en (4.45) y p_l^{ki} de (4.53).

4.2.5 Condición de nodo decisional

Se aplican los mismos criterios que en 4.1.5. También se aplica la nota 4.2 (página 169).

4.2.6 Condición de parada

La condición de parada introducida en (3.22) desde un nodo k para un estrato $S_i \in E$ se basa en el peso del estrato en el nodo y se define de la siguiente forma:

$$Stop_\tau(\beta_k \wedge \mu_k, S_i) = 1 \iff \sum_{\omega\in\Omega} \beta_k \wedge [M = i](\omega) < \tau, i \in S^k \tag{4.56}$$

siendo $\mu_k = [M \in S^k]$.

La expresión $\sum_{\omega\in\Omega} \beta_k \wedge [M = i](\omega), i \in S^k$ es el peso del nodo k y el estrato S_i.

4.3 Caracterización del árbol

4.3.1 Datos monoevaluados y probabilistas

A continuación se muestran algunos resultados que caracterizan el árbol $T = \{\beta_k \wedge \alpha_k \wedge \mu_k\}_{k=1,...,K}$ para datos estratificados construido según la elección de criterios de 4.1 y 4.2. En las proposiciones siguientes se consideran tanto elementos de T como elementos de N. El conjunto N es el conjunto de todos los nodos que forman parte de algún árbol generado en la secuencia del algoritmo (véase 3.3.1).

Proposición 4.6 *$t_k(\mathcal{N})$ es un evento booleano.*

Demostración. Por la construcción algorítmica, se tiene que:

$$t_k(\mathcal{N}) = [M \in S^k] \tag{4.57}$$

con $S^k \subseteq \mathcal{M} = \{1, ..., m\}$.

Dado $\omega \in \Omega$, se tiene que $M(\omega) = i^\omega \in \mathcal{M}$. Y, por tanto, se tiene que:

$$t_k(\mathcal{N})(\omega) = [M \in S^k](\omega) = [M(\omega) \in S^k] = [i^\omega \in S^k] = \begin{cases} 0 \Longleftrightarrow i^\omega \notin S^k \\ 1 \Longleftrightarrow i^\omega \in S^k \end{cases} \tag{4.58}$$

■

Proposición 4.7 *Se tiene que*

$$\mu_r(\omega) = b \wedge \mu_r(\omega) + b^c \wedge \mu_r(\omega), \forall \omega \in \Omega \tag{4.59a}$$

$$\beta_r \wedge \mu_r(\omega) = \beta_r \wedge b \wedge \mu_r(\omega) + \beta_r \wedge b^c \wedge \mu_r(\omega), \forall \omega \in \Omega \tag{4.59b}$$

Demostración. *Dado $\omega \in \Omega$, se tiene por las proposiciones 4.1 y 4.3 que $b(\omega) + b^c(\omega) = 1, \forall \omega \in \Omega$. Por tanto,*

$$\begin{aligned} b \wedge \mu_r(\omega) + b^c \wedge \mu_r(\omega) &= b(\omega) \times \mu_r(\omega) + b^c(\omega) \times \mu_r(\omega) = \\ (b(\omega) + b^c(\omega)) \times \mu_r(\omega) &= \mu_r(\omega) \end{aligned} \tag{4.60}$$

Se deduce para datos monoevaluados de una parte y de otra, para datos modales probabilistas tales que el corte b haga referencia a un predictor que no entra en la definición de β_r:

$$\begin{aligned} \beta_r \wedge b \wedge \mu_r(\omega) + \beta_r \wedge b^c \wedge \mu_r(\omega) &= \beta_r(\omega) \times b(\omega) \times \mu_r(\omega) + \\ \beta_r(\omega) \times b^c(\omega) \times \mu_r(\omega) &= \beta_r(\omega) \times (b(\omega) + b^c(\omega)) \times \mu_r(\omega) = \\ \beta_r(\omega) \times \mu_r(\omega) &= \beta_r \wedge \mu_r(\omega) \end{aligned} \tag{4.61}$$

En caso contrario, sea $b = [Y_j \sim D_j]$ y $\beta_r = \beta_{r-1} \wedge [Y_j \sim D'_j]$ con $D_j, D'_j \in$

$\mathcal{P}(\mathcal{Y}_j)$, *se tiene:*

$$\beta_r \wedge b \wedge \mu_r(\omega) + \beta_r \wedge b^c \wedge \mu_r(\omega) = \beta_{r-1} \wedge [Y_j \sim D_j^{'} \cap D_j] \wedge \mu_r(\omega) +$$
$$\beta_{r-1} \wedge [Y_j \sim D_j^{'} \cap D_j^c] \wedge \mu_r(\omega) = \beta_{r-1} \wedge [Y_j \sim D_j^{'}] \wedge \mu_r(\omega) = \beta_r \wedge \mu_r(\omega) \quad (4.62)$$

■

Proposición 4.8 *En una iteración del algoritmo, si un nodo t_k del árbol se describe en $\mathcal{N}$ por $t_k(\mathcal{N}) = [M \in S^k]$ entonces cualquier nodo t descendiente del nodo t_k obtenido en la iteración cumple que $t(\mathcal{N}) = [M \in S]$ con $S \subseteq S^k$.*

Demostración. Supongamos, sin pérdida de generalidad que el nodo k es el nodo K. Si el nodo t_K es no explorable, no es necesario probar nada ya que no tendrá hijos.

Sea entonces K un nodo explorable y sean K_1 y K_2, los dos nuevos nodos explorables hijos de K. Sean S^{K_1}, $S^{K_2}(\subseteq S^K)$ (véase (3.34)) y R^{K_1}, $R^{K_2}(\subseteq S^K)$ (véase (3.35)) los subconjuntos de indicadores de estratos del nodo K (los nodos K_1 y K_2, respectivamente), que cumplen las condiciones de nodos decisional y terminal desde los nodos K_1 y K_2, respectivamente.

Para $k \in \{K_1,K_2\}$, sea $nuevo(S^k)$ el subconjunto de S^K que resulta de eliminar los indicadores de estratos que se dividen del nodo k para formar un nodo decisional o un nodo terminal (véase 3.3.4 y 3.3.5). Se tiene que:

$$nuevo(S^k) = S^K - (S^k \cup R^k), \text{ para } k \in \{K_1, K_2\} \quad (4.63)$$

y que S^K es unión de tres conjuntos disjuntos:

$$\text{de una parte} \quad nuevo(S^{K_1}), S^{K_1}, R^{K_1} \subseteq S^K \quad (4.64)$$
$$\text{y de otra} \quad nuevo(S^{K_2}), S^{K_2}, R^{K_2} \subseteq S^K \quad (4.65)$$

$$\text{Es decir, } S^K = nuevo(S^k) \cup S^k \cup R^k, \text{ para } k \in \{K_1, K_2\} \tag{4.66}$$

Asimismo, los descendientes t de t_K en esta iteración[3] se describen en $\mathcal{N}$ como:

$$[M \in nuevo(S^k)], [M \in S^k], [M \in R^k], \text{ con } k \in \{K_1, K_2\} \tag{4.67}$$

Es decir, las descripciones en $\mathcal{N}$ de los descendientes t de t_K son de la forma:

$$t(\mathcal{N}) = [M \in S] \text{ con } S \subseteq S^K \tag{4.68a}$$

por (4.64), (4.65) y (4.67). ■

Corolario 4.6 *Los descendientes de un nodo t_k que es explorado, con $t_k(\mathcal{B}\hat{\wedge}\mathcal{N}) = \beta_k \wedge \mu_k$ y $\mu_k = [M \in S^k]$, $S^k \subseteq \{1, ..., m\}$, en una iteración del algoritmo son de la forma, en $\mathcal{B}\hat{\wedge}\mathcal{N}$:*

$$\beta_k \wedge b \wedge \mu_u, u = 1, ..., u_k \tag{4.69}$$

con b el corte óptimo y μ_u descrito como (4.67), cumpliendo las condiciones (4.64) a (4.66). Es decir, siendo:

$$\mu_u = [M \in S^u], \text{con } \cup_{u=1,...,u_k} S^u = S^k \text{ y } S^u \text{ disjuntos dos a dos.} \tag{4.70}$$

O de forma similar a (4.69) para el corte b^c, cumpliéndose (4.70).

Demostración. De una parte, la componente en $\mathcal{B}$ de las ramas hijas y sus descendientes en esta iteración del algoritmo son respectivamente $\beta_k \wedge b$ y $\beta_k \wedge b^c$, $b \in \mathcal{B}$, según 3.3.3.

De otra parte, se deduce de forma trivial de la proposición 4.8 y de su demostración, que la componente en $\mathcal{N}$ de los descendientes en una iteración del

[3] Salvo construcción de varios nodos decisionales en la rama. Véase nota 4.2.

algoritmo del nodo t_k son como (4.70).

Con lo cual se verifica (4.69) y de forma similar para el corte b^c. ■

De forma recurrente se demuestra que si en una iteración del algoritmo, un nodo t_k del árbol se describe en $\mathcal{N}$ por $t_k(\mathcal{N}) = [M \in S^k]$ entonces cualquier descendiente t del nodo t_k cumple que $t(\mathcal{N}) = [M \in S]$ con $S \subseteq S^k$.

Corolario 4.7 *Los descendientes por ramas de un nodo explorable t_k en una iteración del algoritmo representan una partición con respecto al conjunto de los estratos contenidos en t_k.*

Demostración. Sea $t_k(\mathcal{N}) = [M \in S^k]$ la descripción en $\mathcal{N}$ del nodo t_k, $\omega \in \Omega$ con $[M(\omega) \in S^k] = 1$ y b el corte óptimo en esta iteración del algoritmo.

Para una de las ramas, la del corte b, se tiene de la caracterización de los descendientes de r en la iteración, especificadas en (4.69) y (4.70), que dado $\omega \in \{\omega \in \Omega \mid [M(\omega) \in S^k] = 1\}$, entonces existe un único $u_0 \in \{1, ..., u_k\}$ tal que $[M(\omega) \in S^{u_0}] = 1$, es decir, tal que

$$\beta_k \wedge b \wedge \mu_{u_0}(\omega) = \beta_k \wedge b(\omega) \tag{4.71a}$$

$$\beta_k \wedge b \wedge \mu_{u'}(\omega) = 0, \qquad \text{para } u' \in \{1, ..., u_k\} - u_0 \tag{4.71b}$$

Y de igual forma se deduce para los descendientes de la otra rama, la de b^c.

■

4.3.2 Datos monoevaluados

Sea $T = \{\beta_k \wedge \alpha_k \wedge \mu_k\}_{k=1,...,K}$ un árbol obtenido por el proceso de segmentación para datos estratificados para datos de entrada monoevaluados. A continuación se muestran algunos resultados relativos a este árbol.

Proposición 4.9 *Dado un nodo t_k del árbol T, $t_k(\mathcal{B}\hat{\wedge}\mathcal{N})$ es una aserción booleana.*

Demostración. Se tiene que:

$$t_k(\mathcal{B}\hat{\wedge}\mathcal{N}) = t_k(\mathcal{B}) \wedge t_k(\mathcal{N}) \tag{4.72}$$

Por la proposición 4.6, $t_k(\mathcal{N})$ es un evento booleano.

La aserción $t_k(\mathcal{B})$ es una conjunción lógica de eventos booleanos, según la proposición 4.1.

$\wedge$ es el operador conjuntivo estándar.

Por tanto, $t_k(\mathcal{B}\hat{\wedge}\mathcal{N})$ es una aserción booleana. ■

Según la definición 3.1, un individuo $\omega \in \Omega$ se relaciona con un nodo t_k del árbol, $k = 1, ..., K$, descrito en $\mathcal{B}\hat{\wedge}\mathcal{N}$ por $\beta_k \wedge \mu_k$ si $\beta_k \wedge \mu_k(\omega) = 1$, o lo que es lo mismo si $\omega \in Ext_\Omega\{\beta_k \wedge \mu_k\}$ y, ω no se relaciona con el nodo t_k si $\beta_k \wedge \mu_k(\omega) = 0$, o lo que es lo mismo si $\omega \notin Ext_\Omega\{\beta_k \wedge \mu_k\}$.

Proposición 4.10 *El corte $b \wedge \mu_r$ y $b^c \wedge \mu_r$, $b \in B$ define una partición con respecto a los estratos contenidos en μ_r.*

Demostración. Se deduce de la proposición 4.9 y del corolario 4.1 que los elementos de esa partición son $Ext_\Omega(b \wedge \mu_r)$, $Ext_\Omega(b^c \wedge \mu_r)$.

$$\sum_{\omega\in\Omega}(b \wedge \mu_r(\omega) + b^c \wedge \mu_r(\omega)) = \sum_{\omega\in\Omega}\mu_r(\omega) = \sum_{i\in S^r} Card(S_i)$$

con $\mu_r = [M \in S^r]$. ■

Proposición 4.11 *El corte $\beta_r \wedge b$ y $\beta_r \wedge b^c$, $\beta_r \in \mathcal{B}$, $b \in B$ define una partición con respecto a β_r.*

Demostración.

$$\beta_r \wedge b(\omega) = \begin{cases} \beta_r(\omega) & \text{si } b(\omega) = 1 \\ 0 & \text{si } b(\omega) = 0 \end{cases} \tag{4.73a}$$

$$\beta_r \wedge b^c(\omega) = \begin{cases} 0 & \text{si } b^c(\omega) = 0 \Longleftrightarrow b(\omega) = 1 \\ \beta_r(\omega) & \text{si } b^c(\omega) = 1 \Longleftrightarrow b(\omega) = 0 \end{cases} \tag{4.73b}$$

Por tanto,

$$Ext_\Omega(\beta_r) = Ext_\Omega(\beta_r \wedge b) \cup Ext_\Omega(\beta_r \wedge b^c) \tag{4.74}$$

■

Proposición 4.12 *El conjunto de aserciones* $\{\beta_k \wedge \mu_k, k = 1, ..., K\}$ *define una partición del conjunto* Ω.

Demostración.

- *En la primera iteración del algoritmo se cumple que todos los descendientes del primer nodo definen una partición en* Ω.

Sea b_0 el corte óptimo en la primera iteración. Por el corolario 4.1, b_0 y b_0^c definen una partición en Ω. Y por el corolario 4.7, los descendientes en esta iteración de cada una de las ramas definidas por b_0 y b_0^c definen una partición con respecto a los estratos contenidos en el primer nodo que son todos.

- *Supuesto que se verifica en la iteración* L *que el conjunto de aserciones* $\{\beta_k \wedge \mu_k, k = 1, ..., K_L\}$ *define una partición del conjunto* Ω*, veamos que se verifica en la iteración* $L + 1$.

Sea t_r el nodo por el que se realiza la partición en esta iteración del algoritmo con $t_r(\mathcal{B}\hat{\wedge}\mathcal{N}) = \beta_r \wedge \mu_r$ y sea b y b^c el corte óptimo. Basta probar que todos

los descendientes de t_r en esta iteración del algoritmo definen una partición con respecto a $\beta_r \wedge \mu_r$.

Por la proposición 4.11, $\beta_r \wedge b$ y $\beta_r \wedge b^c$ definen una partición con respecto a β_r. Y por el corolario 4.7, los descendientes en esta iteración de cada una de las ramas de t_r definidas por $\beta_r \wedge b$ y $\beta_r \wedge b^c$ definen una partición con respecto a los estratos contenidos en el nodo t_r. ■

Se deduce fácilmente de la proposición 4.12 el siguiente corolario.

Corolario 4.8 *Dado $\omega \in \Omega$ existe un único nodo $k_0 \in \{1, ..., K\}$ que se relaciona con ω, es decir, tal que $\beta_{k_0} \wedge \mu_{k_0}(\omega) = 1$ y para todo $k \neq k_0$ se tiene que $\beta_{k_0} \wedge \mu_{k_0}(\omega) = 0$.*

Proposición 4.13 *El contenido de información del árbol con respecto a Ω no disminuye en cada iteración del algoritmo.*

Demostración. Al ser la medida IC aditiva en el conjunto de nodos del árbol (véase (4.5)), de una iteración a otra del algoritmo la medida de contenido de información del árbol con respecto a Ω, difiere en la parte correspondiente al nodo r que se explora en la iteración (obtenido en (3.29) en 3.3.3).

Es decir, se debe demostrar que:

$$IC\{\beta_r \wedge \alpha_r \wedge \mu_r, \Omega\} \leq \sum_{u=1,\ldots,u_{r_1}} IC\{\beta_r \wedge b \wedge \alpha_r^u \wedge \mu_u, \Omega\} + \sum_{u=1,\ldots,u_{r_2}} IC\{\beta_r \wedge b^c \wedge \alpha_r^{'u} \wedge \mu'_u, \Omega\} \tag{4.75}$$

con $b \in B_{r,\nu}$ el corte optimo obtenido y $\{\{\beta_r \wedge b \wedge \alpha_r^u \wedge \mu_u\}_{u=1,\ldots,u_{r_1}}, \{\beta_r \wedge b^c \wedge \alpha_r^{'u} \wedge \mu'_u\}_{u=1,\ldots,u_{r_2}}\}$ los nodos descendientes de r en la iteración (véase corolario 4.6). Es decir que la medida de contenido de información del nodo r con respecto a Ω es menor o igual que la suma de la medida de contenido de información de los descendientes de r en la iteración, con respecto a Ω.

Esta proposición se demuestra a partir del teorema conocido de no aumento de la entropía de una variable condicionada por otra con respecto a la entropía de la variable[4], obtenido a partir del lema de Gibbs[5] (véase Gil, 1981).

Se demuestra en dos pasos.

En 3.3.3, el nodo r se divide en dos nodos r_1 y r_2 por los valores de un predictor formando una partición de los individuos del nodo r (proposición 4.11). Por el teorema citado, la entropía de Z condicionada por el predictor (y el nodo r) no aumenta con respecto a la entropía de Z condicionada al nodo r. Y, por tanto, también si los términos implicados en la desigualdad se multiplican por $P(\beta_r \wedge \mu_r)$. Es decir,

$$P(\beta_r \wedge \mu_r)Ent(Z|\beta_r \wedge \mu_r) \geq P(\beta_r \wedge b \wedge \mu_r)Ent(Z|\beta_r \wedge b \wedge \mu_r) + P(\beta_r \wedge b^c \wedge \mu_r)Ent(Z|\beta_r \wedge b^c \wedge \mu_r) \quad (4.76)$$

[4]El teorema deduce que para dos variables Y, Z se verifica

$$Ent(Z) \geq \sum_{j=1,...,l_j} P(Y=j)Ent(Z|Y=j)$$

para una variable Y con valores en $\{1,...,l_j\}$. Por tanto, si ambas variables están condicionadas simultánemente por otras (denotadas las condiciones por r), se tiene:

$$Ent(Z|r) \geq \sum_{j=1,...,l_j} P(Y=j|r)Ent(Z|Y=j,r)$$

Y, de igual modo si se multiplican todos los términos por $P(r)$, se sigue:

$$P(r)Ent(Z|r) \geq \sum_{j=1,...,l_j} P(Y=j,r)Ent(Z|Y=j,r)$$

[5]El lema de Gibbs dice que para dos colecciones $(p_1,...,p_s)$ y $(q_1,...,q_s)$ de números no negativos tales que $\sum_{i=1,...,s} p_i = \sum_{i=1,...,s} q_i$, entonces se verifica que $\sum_{i=1,...,s} p_i \log(p_i) \leq \sum_{i=1,...,s} q_i \log(p_i)$

y por tanto,

$$IC\{\beta_r \wedge \alpha_r \wedge \mu_r, \Omega\} \leq IC\{\beta_r \wedge b \wedge \alpha_r^b \wedge \mu_r, \Omega\} + IC\{\beta_r \wedge b^c \wedge \alpha_r^{b^c} \wedge \mu_r, \Omega\} \tag{4.77}$$

En 3.3.4 y 3.3.5, para cada uno de los nodos r_1 y r_2, se obtienen nodos decisionales y terminales que representan dos particiones de los nodos en el conjunto de estratos del nodo r (corolarios 4.6 y 4.7).

Los valores $P(\beta_r \wedge b \wedge \mu_u)$, con μ_u, $u = 1, ..., u_{r_1}$ de (4.70), son las probabilidades empíricas de los nodos descendientes de r_1 en una iteración, y $Ent(Z|\beta_r \wedge b \wedge \mu_u)$ la entropía de Z condicionada por estos nodos. Y, de igual modo para $P(\beta_r \wedge b^c \wedge \mu_u^{'})$ y $Ent(Z|\beta_r \wedge b^c \wedge \mu_u^{'})$, $u = 1, ..., u_{r_2}$.

Por el teorema citado, en cada una de las ramas r_1 y r_2, la entropía de Z condicionada por la variable M condicionada al nodo padre r y transformada en grupos de categorías, no aumenta.Y, por tanto, también si los términos implicados en la desigualdad se multiplican por $P(\beta_r \wedge b \wedge \mu_r)$ o por $P(\beta_r \wedge b^c \wedge \mu_r)$. Por tanto, para r_1 se tiene:

$$P(\beta_r \wedge b \wedge \mu_r) Ent(Z|\beta_r \wedge b \wedge \mu_r) \geq \sum_{u=1,...,u_{r_1}} P(\beta_r \wedge b \wedge \mu_u) Ent(Z|\beta_r \wedge b \wedge \mu_u) \tag{4.78}$$

y por tanto,

$$IC\{\beta_r \wedge b \wedge \alpha_r^b \wedge \mu_r, \Omega\} \leq \sum_{u=1,...,u_{r_1}} IC\{\beta_r \wedge b \wedge \alpha_r^u \wedge \mu_u, \Omega\} \tag{4.79}$$

y de igual modo para r_2. Por tanto se deduce (4.75).

Además, deducido del mismo teorema, esta entropía sólo sería la misma, si la variable Z, fuera independiente del predictor y las transformaciones de la variable M. ■

4.3.3 Datos modales probabilistas

Sea $T = \{\beta_k \wedge \alpha_k \wedge \mu_k\}_{k=1,\dots,K}$ un árbol obtenido por el proceso de segmentación para datos estratificados para datos de entrada modales probabilistas. A continuación se muestran algunos resultados relativos a este árbol.

De (4.36) y (4.37) se sigue la siguiente proposición.

Proposición 4.14 *Dado un nodo t_k del árbol T, $t_k(\mathcal{B}\hat{\wedge}\mathcal{N})$ es una aserción probabilista. Además, dado $\omega \in \Omega$ y $t_k(\mathcal{B}\hat{\wedge}\mathcal{N}) = \wedge_{j\in J}[Y_j \sim D_j]\wedge[M \in S^k]$, $\beta_k \wedge \mu_k(\omega)$ es la probabilidad de que $Y_j \in D_j$, $j \in J \subseteq \{1, ..., p\}$ dado ω y sea de un conjunto de estratos de índice en S^k.*

Proposición 4.15 *El corte $b \wedge \mu_r$ y $b^c \wedge \mu_r$, $b \in B$, $\mu_r = [M \in S^r]$ define una partición con incertidumbre con respecto a los estratos contenidos en μ_r.*

Demostración. Se demuestra fácilmente a partir del corolario 4.4 y por (4.59a).

Se tiene que,

$$\sum_{\omega\in\Omega}(b \wedge \mu_r(\omega) + b^c \wedge \mu_r(\omega)) = \sum_{\omega\in\Omega}\mu_r(\omega) = \sum_{i\in S^r} Card(S_i) \tag{4.80}$$

con $\mu_r = [M \in S^r]$. También, se tiene por (4.59b) que

$$\sum_{\omega\in\Omega}(\beta_r \wedge b \wedge \mu_r(\omega) + \beta_r \wedge b^c \wedge \mu_r(\omega)) = \sum_{\omega\in\Omega}\beta_r \wedge \mu_r(\omega) = \sum_{i\in S^r}\beta_r(\omega) \tag{4.81}$$

para $\beta_r \in \mathcal{B}$. ■

Proposición 4.16 *Dado $\omega \in \Omega$ se tiene que*

$$\sum_{k=1}^{K}\beta_k \wedge \mu_k(\omega) = 1 \tag{4.82}$$

y más concretamente si $\omega \in S_i (\Longleftrightarrow M(\omega) = i)$, *entonces:*

$$\sum_{k \in \{1,...,K\}/i \in S^k} \beta_k \wedge \mu_k(\omega) = 1 \tag{4.83}$$

con $\mu_k = [M \in S^k]$, *para* $k \in \{1, ..., K\}$.

Demostración. En la iteración 0, el primer nodo t_1 cumple $\forall \omega \in \Omega$:

$$\begin{aligned} t_1(\mathcal{B}\hat{\wedge}\mathcal{N})(\omega) = t^{\mathcal{B}\hat{\wedge}\mathcal{N}}(\omega) = [Y_1(\omega) \sim \mathcal{Y}_1] \wedge ... \wedge [Y_p(\omega) \sim \mathcal{Y}_p] \\ \wedge [M(\omega) \in \{1, ..., m\}] = 1 \end{aligned} \tag{4.84}$$

Supongamos que se cumple la condición para la iteración L, es decir:

$$\sum_{k=1,...K_L} \beta_k \wedge \mu_k(\omega) = 1, \forall \omega \in \Omega \tag{4.85}$$

Veamos que también se cumple la condición para la iteración $L + 1$.

Sea el nodo r por el que se realiza el corte óptimo con b y $b^c \in B_{r,\nu}$. Y sea $\omega \in \Omega$, basta probar que

$$\beta_r \wedge \mu_r(\omega) = \sum_{u=1,...,u_r} \beta_u \wedge \mu_u(\omega) \tag{4.86}$$

con $\beta_u \wedge \mu_u$, la descripción en $\mathcal{B}\hat{\wedge}\mathcal{N}$ del descendiente $u-$ésimo del nodo r, en esta iteración del algoritmo, según se deduce del corolario 4.6, con $u \in \{1, ..., u_r\}$[6], $\beta_u = \beta_r \wedge b$ o $\beta_u = \beta_r \wedge b^c$.

Por la proposición 4.7, se tiene que b y b^c cumplen que

$$\beta_r \wedge \mu_r(\omega) = \beta_r \wedge b \wedge \mu_r(\omega) + \beta_r \wedge b^c \wedge \mu_r(\omega), \forall \omega \in \Omega \tag{4.87}$$

[6]Haciendo uso de la notación que considera conjuntamente este índice de 1 en adelante uniendo los hijos en esta iteración de ambas ramas. (Descendientes rama izquierda: $\mu_u, u = 1, ..., u_{r_1}$; descendientes rama derecha: $\mu_u, u = 1, ..., u_{r_2}$; total descendientes: $\mu_u, u = 1, ..., u_r$, con $r = r_1 + r_2$)

Por el corolario 4.6 se tiene que el nodo descrito en $\mathcal{B}\hat{\wedge}\mathcal{N}$ por $\beta_r \wedge b \wedge \mu_r$ se divide en los hijos $\{\beta_r \wedge b \wedge \mu_u, u = 1, ..., u_{r_1}\}$, con $\mu_u = [M \in S^u]$, con $\cup_{u=1,...,u_{r_1}} S^u = S^r$ y los conjuntos S^u disjuntos dos a dos. Se tiene por el corolario 4.7:

$$\beta_r \wedge b \wedge \mu_r(\omega) = \sum_{u=1,..,u_{r_1}} \beta_r \wedge b \wedge \mu_u(\omega) \tag{4.88}$$

Y de igual modo

$$\beta_r \wedge b^c \wedge \mu_r(\omega) = \sum_{u=1,..,u_{r_2}} \beta_r \wedge b^c \wedge \mu_u(\omega) \tag{4.89}$$

con lo que queda demostrado (4.86).

Además, sea $\omega \in \Omega$, con $M(\omega) = i \in \{1, ..., m\}$, se comprueba facilmente que:

$$\sum_{k=1,...,K_{L+1}} \beta_k \wedge \mu_k(\omega) = \sum_{k/i \in S^k} \beta_k \wedge \mu_k(\omega) = 1 \tag{4.90}$$

$$\text{ya que} \qquad \mu_k(\omega) = 0 \Longleftrightarrow M(\omega) = i \notin S^k \tag{4.91}$$

siendo $\mu_k = [M \in S^k]$ para $k = 1, ..., K$. ■

Se deducen de las proposiciones 4.14 y 4.16, la siguiente proposición y corolario.

Proposición 4.17 *El conjunto de aserciones* $\{\beta_k \wedge \mu_k, k = 1, ..., K\}$ *define una partición con incertidumbre en el conjunto* Ω.

Corolario 4.9 *El subconjunto de nodos de* T *que tienen un nivel de relación no nula con un individuo* $\omega \in \Omega$ *tal que* $M(\omega) = i$, *está contenido en el subconjunto* $\{k/i \in S^k\}$, *siendo* $\mu_k = [M \in S^k]$ *para* $k = 1, ..., K$.

Proposición 4.18 *El contenido de información del árbol con respecto a* Ω *no disminuye en cada iteración del algoritmo.*

Demostración. La demostración se asemeja a la demostración de la proposición 4.13, considerando la aditividad de la medida IC en el conjunto de nodos del árbol (véase (4.38)) y, centrándose en la no disminución de la medida de contenido de información del árbol con respecto a Ω correspondiente al nodo r que se explora en la iteración (véase (3.29) en 3.3.3) utilizando el teorema conocido de no aumento de la entropía de una variable condicionada por otra con respecto a la entropía de la variable (véase Gil, 1981). Se debe demostrar la expresión (4.75).

En 3.3.3, el nodo r se divide en dos nodos r_1 y r_2 por los valores de un predictor. Los valores $P(\beta_r \wedge b \wedge \mu_r)$ y $P(\beta_r \wedge b^c \wedge \mu_r)$ estiman las probabilidades de los nodos r_1 y r_2 (véase (4.39)) y $Ent(Z|\beta_r \wedge b \wedge \mu_r)$ y $Ent(Z|\beta_r \wedge b^c \wedge \mu_r)$ la entropía de Z condicionada por estos nodos. Por el teorema citado, se puede deducir (4.76) y (4.77).

En 3.3.4 y 3.3.5, para cada uno de los nodos r_1 y r_2, se obtienen nodos decisionales y terminales que representan dos particiones de los nodos en el conjunto de estratos del nodo r (corolarios 4.6 y 4.7).

Los valores $P(\beta_r \wedge b \wedge \mu_u)$, con μ_u, $u = 1, ..., u_{r_1}$ de (4.70), estiman las probabilidades de los nodos descendientes de r_1 (véase (4.39)) y $Ent(Z|\beta_r \wedge b \wedge \mu_u)$, $u = 1, ..., u_{r_1}$ la entropía de Z condicionada por estos nodos.

Por el teorema citado, se cumplen las expresiones (4.78) y (4.79). Y, de igual modo para r_2.

Por tanto, se deduce la expresión (4.75). ■

4.4 Descripción simbólica de los estratos

Los estratos pueden describirse por un conjunto de objetos simbólicos con unos pesos respectivos. Los objetos simbólicos representan las reglas de predicción y los pesos miden la importancia relativa de estas reglas en el estrato.

4.4.1 Datos monoevaluados

Para un estrato S_i, los pesos w_k^i en (3.11), $k \in \{1, \ldots, K\}$, se definen como las probabilidades condicionadas empíricas de los nodos al estrato S_i:

$$w_k^i = P(\beta_k \wedge \mu_k | S_i), k \in \{1, \ldots, K\} \tag{4.92}$$

los pesos se calculan por la proporción de individuos de los nodos y estrato con respecto al total de individuos del estrato.

Sean $k \in \{1, \ldots, K\}$ los nodos del árbol con $t_k(\mathcal{N}) = [M \in S^k]$. Si en la descripción de k forma parte el estrato S_i, es decir si $i \in S^k$, entonces:

$$w_k^i = \frac{Card(Ext_{S_i}(\beta_k))}{Card(S_i)} \tag{4.93}$$

mientras que para los nodos $k \in \{1, \ldots, K\}$ tales que el estrato i no forma parte de su definición, es decir si $i \notin S^k$, entonces:

$$w_k^i = 0 \tag{4.94}$$

Ejemplo 4.1 ***Descripción de un municipio del ejemplo 3.3***

Para los datos de entrada del ejemplo 3.3, la descripción del municipio S_{19} *es:*

$$\begin{aligned} S_{19} : \{&0.6[espacio{=}\,(+)] \wedge [optimismo{=}\,(+)] \wedge [Z \sim (-)\,0.08, (+)\,0.92], \\ &0.05[espacio = (-)] \wedge [optimismo{=}\,(-)] \wedge [Z \sim (-)\,0.91, (+)\,0.09], \\ 0.25&[espacio = (+)] \wedge [optimismo = (-)] \wedge [habitab = (+)] \wedge [Z \sim (-)\,0.28, (+)\,0.72], \\ 0.04&[espacio = (+)] \wedge [optimismo = (-)] \wedge [habitab = (-)] \wedge [integracion = (+)] \\ &\wedge [conformismo = (-)] \wedge [Z \sim (-)\,0.4, (+)\,0.6]\} \end{aligned} \tag{4.95}$$

Es decir, el municipio S_{19} *se describe por estas cuatro reglas que tienen pesos*

respectivos 0.6, 0.05, 0.25 y 0.04. Los pesos se calculan por la proporción de individuos de estos nodos con respecto al total de individuos del municipio S_{19}. *En 4.9.3 se puede ver una salida más completa de este ejemplo y en la figura 4.6 que se presenta allí, se recuadran doblemente los nodos que describen el municipio* S_{19}.

4.4.2 Datos modales probabilistas

Para un estrato S_i, los pesos w_k^i en (3.11), $k \in \{1, \ldots, K\}$, se definen las estimaciones de las probabilidades condicionadas de los nodos al estrato. Es decir,:

$$w_k^i = P(\beta_k \wedge \mu_k | S_i), k \in \{1, \ldots, K\} \tag{4.96}$$

Sean $k \in \{1, \ldots, K\}$ los nodos del árbol con $t_k(\mathcal{N}) = [M \in S^k]$. Si en la descripción de k forma parte el estrato S_i, es decir si $i \in S^k$, entonces:

$$w_k^i = \frac{\sum_{\omega \in S_i} \beta_k(\omega)}{Card(S_i)} \tag{4.97}$$

mientras que para los nodos $k \in \{1, \ldots, K\}$ tales que el estrato i no forma parte de su definición, es decir si $i \notin S^k$, entonces:

$$w_k^i = 0 \tag{4.98}$$

4.4.3 Interpretación de los estratos

Se introducen aquí dos términos que recuerdan la contribución relativa y absoluta de los factores obtenidos en un Análisis de Componentes Principales o Análisis de Correspondencias (Diday et al., 1982) y que, en este caso contribuyen a la interpretación de los estratos y los nodos.

Contribución relativa de un nodo a un estrato

Este término contribuye a la interpretación del estrato. Hace referencia a los pesos w_k^i de (3.11). Sea el estrato S_i, las contribuciones relativas de los nodos $\{t_k\}_{k=1,...K}$ del árbol, con $t_k(\mathcal{B}\hat{\wedge}\mathcal{N}) = \beta_k \wedge \mu_k$, al estrato S_i se definen como:

$$\begin{aligned} w_k^i &= P(\beta_k \wedge \mu_k | S_i) := P(\beta_k \wedge \mu_k | [M=i]) \\ &= \begin{cases} 0 & \text{Si } \mu_k \cap [M=i] = \emptyset^{\mathcal{N}} \\ \frac{P(\beta_k \wedge [M=i])}{P([M=i])} & \text{En otro caso} \end{cases} \quad \text{Para } k = 1, ..., K \end{aligned} \tag{4.99}$$

El término w_k^i mide la importancia relativa del nodo k en el estrato S_i. Valores altos de w_k^i, $k = 1, ...K$ indican las reglas que son de importancia para el estrato S_i. Las expresiones (4.92) y (4.96) muestran las contribuciones relativas para datos de entrada monoevaluados y modales probabilistas, respectivamente.

Contribución absoluta de un estrato en un nodo.

Este término contribuye a identificar los estratos que caracterizan un nodo decisional. Sea t_k un nodo del árbol descrito en $\mathcal{B}\hat{\wedge}\mathcal{N}$ por $\beta_k \wedge \mu_k$, y sean los estratos $\{S_1, ..., S_m\}$, las contribuciones absolutas de los estratos al nodo t_k se definen como:

$$\begin{aligned} wa_i^k &= P(S_i | \beta_k \wedge \mu_k) := P([M=i] | \beta_k \wedge \mu_k) \\ &= \begin{cases} 0 & \text{Si } \mu_k \cap [M=i] = \emptyset^{\mathcal{N}} \\ \frac{P(\beta_k \wedge [M=i])}{P(\beta_k \wedge \mu_k)} & \text{En otro caso} \end{cases} \quad \text{Para } i \in \{1, ..., m\} \end{aligned} \tag{4.100}$$

El término wa_i^k mide la importancia del estrato S_i en el nodo k. Valores altos de wa_i^k, $i = 1, ..., m$ indican los estratos que caracterizan un nodo decisional k.

Sea $\mu_k = [M \in S^k]$, para $i \in S^k$,

$$wa_i^k = \frac{Card(Ext_{S_i}(\beta_k))}{Card(Ext_\Omega(\beta_k \wedge \mu_k))}, \quad \text{para datos monoevaluados} \quad (4.101)$$

$$wa_i^k = \frac{\sum_{\omega \in S_i} \beta_k(\omega)}{\sum_{\omega \in \Omega} \beta_k \wedge \mu_k(\omega)}, \quad \text{para datos modales probabilistas} \quad (4.102)$$

4.4.4 Ventajas del método

En esta sección se describen algunas ventajas del método con respecto a los árboles de Segmentación tradicionales, con independencia de una de sus ventajas fundamentales que es la incorporación de datos simbólicos en el mismo. De una parte, se compara el método propuesto con la aplicación de un método de Segmentación que considera la variable estrato como un predictor más. Y de otra, se compara con la aplicación de un método de Segmentación para cada estrato independientemente. Esta comparación se ilustra con datos monoevaluados, si bien las conclusiones favorables del método son igualmente válidas para datos modales probabilistas.

En primer lugar, hay que destacar que los objetivos del método son diferentes y en segundo lugar que la representación del árbol aporta más información global que las otras dos alternativas.

Aunque de las reglas de predicción obtenidas por el método propuesto, pudiera pensarse que la variable estrato juega el papel de un predictor más, sin embargo, la aproximación del método es muy diferente de la aproximación que considera la variable estrato como un predictor, ya que el objetivo que se persigue es la *generalización o caracterización del estrato por objetos simbólicos* que representan las reglas de predicción de la variable clase por los predictores (véase 1.4.5, 3.1.2 y 3.2.3). Por tanto es una variable de interés prioritario que requiere su incorporación a las reglas de predicción en todo momento. Además, el método propuesto favorece la incorporación en las reglas de predictores que son buenos en los estratos individualmente.

Por otra parte, si se obtiene un árbol diferente para cada estrato, el objetivo de la *clasificación de los estratos por reglas de predicción comunes* para subconjuntos de estratos, tampoco se alcanzaría. De este modo, se obtiene el mejor árbol para cada estrato pero no una clasificación de los estratos. El método propuesto, además, tiene la ventaja que la interpretación de los estratos que se obtiene da una idea conjunta del estrato en el entorno de todos los estratos, y no aisladamente.

M	Y1	Y2	Z
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	2	1
1	2	1	2
1	2	1	2
1	2	1	2
1	2	2	2
1	2	2	2
1	2	2	2
1	2	2	2
1	2	2	2
1	2	2	2
1	2	2	2
2	1	2	2
2	1	2	2
2	1	2	2
2	1	2	2
2	1	2	2
2	1	2	2
2	1	2	2
2	1	1	2
2	1	1	2
2	1	1	2
2	2	2	1
2	2	1	1
2	2	1	1
2	2	1	1
2	2	1	1
2	2	1	1
2	2	1	1
2	2	1	1
2	2	1	1
2	2	1	1

M	Y1	Y2	Z
3	1	1	2
3	1	1	2
3	1	2	2
3	1	2	2
3	1	2	2
3	1	2	2
3	1	2	2
3	1	2	2
3	1	2	2
3	1	1	1
3	1	1	1
3	2	1	1
3	2	1	1
3	2	1	1
3	2	1	1
3	2	2	1
3	2	2	1
3	2	2	1
3	2	2	1
3	2	2	2
3	2	2	2
4	1	1	1
4	1	1	1
4	1	1	1
4	1	1	1
4	1	1	1
4	1	1	1
4	1	2	1
4	1	2	1
4	1	1	2
4	1	1	2
4	2	2	2
4	2	2	2
4	2	2	2
4	2	2	2
4	2	2	2
4	2	2	2
4	2	2	2
4	2	2	1
4	2	2	1

Tabla 4.1: Datos de entrada de 4.4.4

El siguiente ejemplo muestra cómo se obtienen reglas de predicción diferentes con el método propuesto y con el tradicional. Sea un conjunto Ω de 80 individuos descritos por datos monoevaluados (véase tabla 4.1) en la variable estrato M ($\mathcal{M} = \{1, ..., 4\}$), los predictores Y_1, Y_2 ($\mathcal{Y}_j = \{1, 2\}, j = 1, 2$) y la variable clase Z ($\mathcal{Z} = \{1, 2\}$), ($m = 4$, $n = 80$). Se construyen para este conjunto de datos el árbol estratificado y una aplicación clásica de los árboles de Segmentación.

El criterio de optimización elegido es la minimización de la entropía ponderada

de la variable clase Z en los nodos del árbol. Las predicciones en los nodos decisionales se estiman por distribuciones empíricas sobre el dominio $\mathcal{Z}$. En el nodo inicial, se parte de una entropía máxima para Z de 0.69.

La figura 4.1 representa el árbol de decisión considerando la información de los estratos y la figura 4.2 el árbol de Segmentación obtenido por un algoritmo de Segmentación tradicional. Los criterios de parada que se adoptan en esta segunda aplicación son los mismos que al considerar la información de los estratos, en términos generales, no proseguir el proceso si el tamaño del nodo es pequeño o la estimación de la variable clase en el nodo cumple el criterio de nodo decisional globalmente. En ambas figuras, los nodos terminales o decisionales son rectangulares, n indica el número de individuos y p la probabilidad estimada para la primera clase, es decir, para $Z = 1$. Esta probabilidad se estima por la proporción de individuos $\omega \in \Omega$ que son del nodo con $Z(\omega) = 1$.

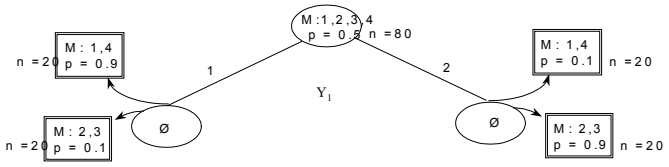


Figura 4.1: Árbol para datos estratificados

El árbol para datos estratificados se muestra en la figura 4.1, donde M identifica los indicadores de estratos de un nodo. La entropía final ponderada del árbol es 0.325. Los nodos decisionales obtenidos son:

$[Y_1 = 1] \wedge [Z \sim (1(\,0.9), 2\,(0.1)] \wedge [M \in \{1, 4\}]$

$[Y_1 = 1] \wedge [Z \sim (1(0.1), 2(0.9)] \wedge [M \in \{2, 3\}]$

$[Y_1 = 2] \wedge [Z \sim (1(0.1), 2(0.9)] \wedge [M \in \{1, 4\}]$

$[Y_1 = 2] \wedge [Z \sim (1(0.9), 2(0.9)] \wedge [M \in \{2, 3\}]$

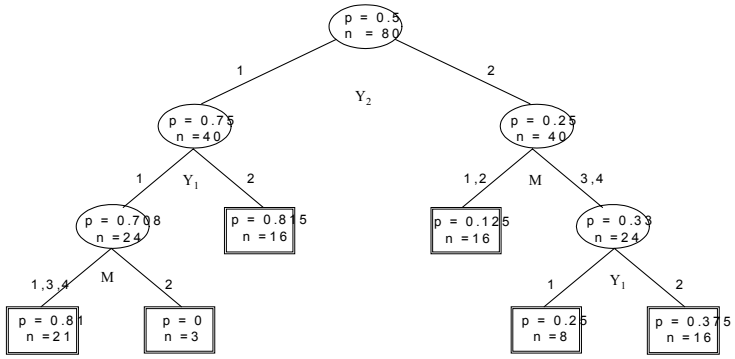


Figura 4.2: Árbol de decisión tradicional

La aplicación del algoritmo de Segmentación tradicional al considerar M otro predictor, obtiene el árbol de la figura 4.2. La entropía final ponderada del árbol es ahora 0.48 y las reglas de predicción obtenidas son diferentes:

$[Y_2 = 1] \wedge [Y_1 = 1] \wedge [Z \sim (1(0.81), 2(0.19)] \wedge [M \in \{1, 3, 4\}]$

$[Y_2 = 1] \wedge [Y_1 = 1] \wedge [Z = 2] \wedge [M = 2]$

$[Y_2 = 1] \wedge [Y_1 = 2] \wedge [Z \sim (1(0.81), 2(0.19)]$

$[Y_2 = 2] \wedge [Z \sim (1(0.12), 2(0.88)] \wedge [M \in \{1, 2\}]$

$[Y_2 = 2] \wedge [Y_1 = 1] \wedge [Z \sim (1(0.25), 2(0.75)] \wedge [M \in \{3, 4\}]$

$[Y_2 = 2] \wedge [Y_1 = 2] \wedge [Z \sim (1(0.375), 2(0.625)] \wedge [M \in \{3, 4\}]$

El método propuesto para datos estratificados considera la *información de los estratos en todos los pasos* del algoritmo, proporciona un *árbol más simple* (más corto) con reglas más simples y obtiene una *entropía de la variable clase menor*. Y, lo más importante, proporciona conjuntos de estratos con reglas de predicción comunes y una descripción de cada estrato o grupos de individuos por objetos

simbólicos. La descripción de los estratos en este ejemplo, según el árbol de la figura 4.1 es:

$$S_1 = S_4 = \{0.5\,[Y_1 = 1] \wedge [Z \sim 1\,(0.9), 2\,(0.1)]\ ,\ 0.5\,[Y_1 = 2] \wedge [Z \sim 1\,(0.1), 2\,(0.9)]\}$$

$$S_2 = S_3 = \{0.5\,[Y_1 = 1] \wedge [Z \sim 1\,(0.1), 2\,(0.9)]\ ,\ 0.5\,[Y_1 = 2] \wedge [Z \sim 1\,(0.9), 2\,(0.1)]\}$$

Si se modifican algunos umbrales en el método para datos estratificados, entonces la entropía ponderada obtenida es de 0.23 para la variable clase Z; mientras que en la aplicación clásica el mejor árbol o de menor entropía, proporciona un valor de 0.39.

Otra diferencia que puede apreciarse en las dos aplicaciones es que si bien el predictor Y_2 globalmente es más discriminante (véase figura 4.2), sin embargo cuando se incorpora la información de los estratos, es el predictor Y_1 el más discriminante.

En resumen, la incorporación de la información de los estratos en todos los pasos del algoritmo proporciona que éstos puedan describirse por reglas de predicción en todos los pasos del algoritmo. El árbol obtenido con el método propuesto para datos estratificados considera la información de los estratos presentando parsimonia en la representación al contener en un único árbol reglas de predicción comunes a varios estratos y favorece los buenos predictores en los estratos redundando, en esta aplicación, en la obtención de un árbol más corto que con el método tradicional.

Se demuestra fácilmente la siguiente proposición.

Proposición 4.19 *Parsimonia en la representación. En una iteración del algoritmo, sea un nodo explorable $r \in X$, con $\mu_r = [M \in S^r]$, si todos los estratos $i \in S^r$ cumplen que*

$$\beta'_r = \arg\max_{b \in B_{r,\nu}} \{IC\{\{\beta_r \wedge b \wedge \alpha_r^b \wedge \mu_r, \beta_r \wedge b^c \wedge \alpha_r^{b^c} \wedge \mu_r\}, S_i\} \tag{4.103}$$

con $\alpha_r^b = Calc_\alpha(\beta_r \wedge b \wedge \mu_r)$ y $\alpha_r^{b^c} = Calc_\alpha(\beta_r \wedge b^c \wedge \mu_r)$ para $b \in B_{r,\nu}$, entonces

se tiene que:

$$\beta'_r = \arg \max_{b \in B_{r,\nu}} \{EIC\{T, r, b, E\}\} \tag{4.104}$$

Esta proposición demuestra que si el corte que mejor predice las clases es el mismo para todos los estratos en un nodo, entonces es el mejor para todos globalmente. Una vez realizado el corte por el predictor, los estratos que verifican la condición de nodo decisional se agrupan para formar una regla de predicción común, mientras que el resto de los estratos prosiguen el proceso recursivo. Y, así, sucesivamente.

Quiere esto decir que la obtención de un único árbol de Segmentación para todos los estratos según el método propuesto, en el caso que los cortes óptimos sean los mismos para todos los estratos, obtiene los mismos cortes óptimos, evitando la creación de tantos árboles como estratos. Y, que si las reglas de predicción son las mismas para todos los estratos, el árbol que se obtiene para todos los estratos es el mismo que el árbol que se obtiene por el método que incorpora la información de los estratos.

Además, el método tiene las ventajas de distinguir estratos con reglas antagónicas, es decir, los mismos valores de los predictores pueden prededir distinta clase (véase figura 4.1 y más adelante en la aplicación de 4.9.1); de caracterizar unos estratos antes que otros (véase aplicación 4.9.3 más adelante), es decir, que salen del proceso recursivo con anterioridad; de distinguir estratos que predicen la misma clase con reglas comunes salvo los valores de uno de los predictores y, en definitiva, de clasificar estratos por reglas de predicción comunes y de distinguir estratos según reglas de predicción distintas.

4.5 Predicción

Se ha visto hasta el momento que el árbol de Segmentación para datos estratificados proporciona una descripción de las clases de Z según los valores de los predictores y según el estrato de procedencia de los individuos. Asimismo, se describen por medio de objetos simbólicos cada uno de los estratos, mediante las reglas de predición de clases de Z que se verifican en ellos y los pesos que dichas reglas tienen en estos estratos.

Además del *propósito descriptivo* de esta técnica, que es, sin duda fundamental y en muchas ocasiones el único propósito, la obtención de un árbol de Segmentación para datos estratificados a partir de una *muestra diseño*, proporciona un mecanismo de predicción de nuevos individuos que provengan de los mismos estratos que la muestra diseño. Esta sección describe diversas formas de asignar estas predicciones.

En la fase de creación del árbol se considera que todos los individuos tienen valor conocido en los predictores, variable clase y variable estrato. En la fase de predicción, la variable clase es desconocida y se desea estimar mediante el árbol obtenido. En esta fase, además, se considera la predicción para observaciones incompletas en las que alguna de las variables predictoras poseen valores desconocidos o no observados.

Una *regla de predicción* R asocia a una descripción de un individuo $\omega \in \Omega_2$ en los espacios de descripciones $\mathcal{Y}_1 \times ... \times \mathcal{Y}_p \times \mathcal{M}$ (datos monoevaluados) o $\mathcal{M}(\mathcal{Y}_1) \times ... \times \mathcal{M}(\mathcal{Y}_p) \times \mathcal{M}$ (datos modales probabilistas), una descripción en el espacio de descripciones $\mathcal{Z}$ o $\mathcal{M}(\mathcal{Z})$ o el evento correspondiente en $\mathcal{A}$ (véase (3.8) y (3.9)).

4.5.1 Nivel de relación con el árbol

Sea un árbol T definido por $T = \{\beta_k \wedge \alpha_k \wedge \mu_k\}_{k=1,...,K}$ (véase 3.5) y sea $\Omega_2 = \{\omega_1, ..., \omega_r\}$ un conjunto de individuos, la *muestra de predicción*. Sean

los individuos $\omega \in \Omega_2$ descritos por los predictores $Y_1, \ldots, Y_p$ (monoevaluados o modales probabilistas) y la variable estrato definida para $\omega \in \Omega_2$ como $M(\omega) = i$, si pertenece al estrato $i-$ésimo con $i \in \{1, ..., m\}$. Sea $\Omega^* = \Omega \cup \Omega_2$, el conjunto total de individuos, el de la muestra diseño y el de la muestra de predicción.

Se extiende la definición de los objetos simbólicos de los conjuntos B, $\mathcal{B}$, $\mathcal{A}$ y $\mathcal{N}$, que definen los nodos del árbol, al conjunto Ω^*, de forma trivial. Dado que pueden existir valores no observados en los predictores de la muestra de predicción, a los efectos de los árboles de Segmentación, se establece que el nivel de relación de una descripción con un valor desconocido o no observado es nulo, es decir,

$$[desconocido \mathcal{R} d] = [d \mathcal{R} desconocido] = 0 \tag{4.105}$$

cualquiera que sea la relación $\mathcal{R}$.

Se extiende la definición 3.1 de nivel de relación de un individuo con un nodo del árbol al conjunto Ω^*. Y, asimismo, se define el nivel de relación de un individuo con un árbol como una función de los niveles de relación del individuo con los nodos del árbol.

Definición 4.1 ***Nivel de relación de un individuo con el árbol.*** *El nivel de relación del individuo $\omega \in \Omega^*$ con el árbol T es una función de los niveles de relación del individuo con los nodos del árbol, es decir, sea la función:*

$$\begin{array}{rlll} F: & \Omega^* & \longrightarrow \mathcal{L}(\subseteq [0,1]) \\ & \omega & \longmapsto F(\omega) = H\{\beta_k \wedge \mu_k(\omega)\}_{k=1,\ldots,K} \end{array} \tag{4.106}$$

el valor $F(\omega)$ se define como una función $H(.)$ aplicada a los niveles de relación de individuo ω con los nodos del árbol. $F(\omega)$ es el nivel de relación del individuo ω con el árbol T.

En la aproximación para datos monoevaluados y modales probabilistas, se

considera que la función $H(.)$ es la suma. Es decir,

$$F(\omega) = \sum_{k=1,...K} \beta_k \wedge \mu_k(\omega) \tag{4.107}$$

Se cumplen también en Ω^* las proposiciones 4.6 y los corolarios 4.7 y 4.9 y si las observaciones son completas la proposición 4.7.

Datos monoevaluados

Se cumplen en Ω^* la proposición 4.9, y si las observaciones de Ω_2 son completas, las proposiciones 4.1, 4.10, 4.11 y 4.12 y los corolarios 4.1 y 4.8.

Corolario 4.10 *El nivel de relación entre un individuo $\omega \in \Omega_2$ y un árbol T vale 0 o 1, es decir, $F(\omega) \in \{0, 1\}$. Y más concretamente,*

- *Si $\omega \in \Omega_2$ es una observación completa en $Y_1 \times ... \times Y_p \times M$, entonces el nivel de relación del árbol T con ω es 1.*
- *El nivel de relación del árbol T con ω es 0, sólo en el caso que ω tenga valor desconocido en algún predictor de la regla de predicción que le corresponde.*

Demostración. Es evidente a partir del corolario 4.8 y debido a que el nivel de relación entre un valor desconocido o no observado y una descripción es nulo. ■

Datos modales probabilistas

Se cumplen en Ω^*, si las observaciones de Ω_2 son completas, las proposiciones 4.3, 4.14, 4.15, 4.16 y 4.17 y el corolario 4.4. Se deducen fácilmente los siguientes corolarios.

Corolario 4.11 *El nivel de relación de un individuo ω y un árbol T es 1 para una observación completa.*

Corolario 4.12 *Si el nivel de relación de un individuo ω y un árbol T es menor que* 1*, entonces la observación es incompleta.*

4.5.2 Reglas de predicción para datos monoevaluados

Sea el árbol $T = \{\beta_k \wedge \alpha_k \wedge \mu_k\}_{k=1,\dots,K}$ de 4.5.1 con $\alpha_k = [Z \sim q_k]$. Cada una de las aserciones $\beta_k \wedge \alpha_k \wedge \mu_k$ proporciona una regla de predicción básica para los elementos $\omega \in \Omega_2$ tales que $\beta_k \wedge \mu_k(\omega) = 1$. Esta afirmación puede precisarse algo más. Sea $\omega \in \Omega_2$, entonces pueden darse dos situaciones alternativas:

- En el caso de que el árbol T no se relacione con ω, entonces no se proporciona predicción. Esta circunstancia sólo puede darse en el caso de que alguno de los predictores no sea observado para ω y que la regla de aplicación para ese individuo contenga ese predictor. Una variante a la no predicción puede ser una generalización de la aproximación de Quinlan, 1990, cuando existe falta de observación. Propone que cuando en una regla de predicción no se puede asignar ω a ninguno de los nodos hijos, se calculan con la muestra diseño las probabilidades condicionadas de las ramas con respecto al padre y se asigna ω a ambos nodos hijos con las probabilidades condicionadas estimadas con la muestra diseño. Estas probabilidades son trasmitidas a los descendientes. En este caso la asignación de las estimaciones se realiza como en 4.5.3.

- Si el árbol T se relaciona con ω, sea k_0 el nodo del árbol T que se relaciona con ω, entonces la predicción de Z para ω viene dada por el evento $\alpha_{k_0} = [Z \sim q_{k_0}]$. Se pueden proporcionar dos posibles predicciones:

  - La distribución de probabilidad definida por q_{k_0}
  - La clase de mayor probabilidad definida según la distribución q_{k_0}

  Una variante a estas predicciones puede ser la asignación aleatoria de una clase de Z según ley de probabilidad q_{k_0}.

4.5.3 Reglas de predicción para datos modales probabilistas

Sea el árbol $T = \{\beta_k \wedge \alpha_k \wedge \mu_k\}_{k=1,\dots,K}$ de 4.5.1 con $\alpha_k = [Z \sim q_k]$. Cada una de las aserciones $\beta_k \wedge \alpha_k \wedge \mu_k$ proporciona una regla de predicción con incertidumbre básica para los elementos $\omega \in \Omega_2$. A partir de estas reglas de predicción básicas se pueden elegir diversas alternativas para proporcionar una predicción a los inidividuos.

Aplicaciones auxiliares a la predicción

A continuación, se proporcionan aplicaciones auxiliares a la predicción que obtienen para los elementos de Ω_2 descripciones en $\mathcal{M}(\mathcal{Z})$. Estas descripciones son las descripciones simbólicas de eventos de $\mathcal{A}$. A partir de estos eventos se proporcionan las predicciones de los elementos de Ω_2. Estas aplicaciones auxiliares son:

- La aplicación que asocia a los elementos $\omega \in \Omega_2$, una distribución de probabilidad que pondera por los niveles de relación del individuo ω con los nodos t_k del árbol las distribuciones de probabilidad estimadas en los nodos para la variable clase Z, es decir:

$$\begin{aligned} R_1^{aux}: \ & \Omega_2 \longrightarrow \mathcal{M}(\mathcal{Z}) \\ & \omega \longmapsto R_1^{aux}(\omega) = q_\omega^1 := \sum_k \beta_k \wedge \mu_k(\omega) q_k = \sum_{k/i^\omega \in S^k} \beta_k(\omega) q_k \end{aligned} \tag{4.108}$$

  siendo $M(\omega) = i^\omega \in \{1, ..., m\}$, $\mu_k = [M \in S^k]$, para $k = 1, ..., K$. $R_1^{aux}(\omega) = q_\omega$ es una distribución de probabilidad que se identifica con $q_\omega^1 \equiv (c_1\ p_{\omega 1}^1, ..., c_s\ p_{\omega s}^1)$. Es una mixtura de las distribuciones de probabilidad definidas en los nodos del árbol. Los valores de la ponderación son los niveles de relación del individuo con los nodos, o las probabilidades de los

nodos dado el individuo. Antecedentes de combinación de distribuciones de probabilidad ponderadas con pesos de suma 1, pueden verse en Dubois y Prade, 1989 y aplicación de las mismas en árboles de Segmentación con datos simbólicos en Périnel, 1996.

- La aplicación que asocia a los elementos $\omega \in \Omega_2$, la distribución de probabilidad estimada para el nodo del árbol que tiene un nivel de relación mayor con el elemento ω, es decir,

$$\begin{array}{rccl} R_2^{aux}: & \Omega_2 & \longrightarrow & \mathcal{M}(\mathcal{Z}) \\ & \omega & \longmapsto & R_2^{aux}(\omega) = q_{k_\omega} \end{array} \tag{4.109}$$

$$\text{con } k_\omega = \arg\max_k \beta_k \wedge \mu_k(\omega) = \arg \max_{k/i^\omega \in S^k} \beta_k(\omega) \tag{4.110}$$

siendo $M(\omega) = i^\omega \in \{1, ..., m\}$, $\mu_k = [M \in S^k]$, para $k = 1, ..., K$. Es decir, se asigna la estimación del nodo al cual tiene mayor probabilidad de pertenecer el elemento ω.

A partir de estas aplicaciones auxiliares (4.108) y (4.109) que proporcionan una descripción en $\mathcal{M}(\mathcal{Z})$, se obtienen los correspondientes elementos de $\mathcal{A}$ asignando dichas descripciones a los eventos correspondientes:

$$\begin{array}{rccl} R^{aux}: & \Omega_2 & \longrightarrow & \mathcal{A} \\ & \omega & \longmapsto & R^{aux}(\omega) = \begin{cases} [Z \sim R_1^{aux}(\omega)] & \text{Si el criterio es } R_1^{aux} \\ [Z \sim R_2^{aux}(\omega)] & \text{Si el criterio es } R_2^{aux} \end{cases} \end{array} \tag{4.111}$$

Se identifica

$$R^{aux}(\omega) = [Z \sim q_\omega], \text{ con } q_\omega \equiv (c_1\ p_{\omega 1}, ..., c_s\ p_{\omega s}) \tag{4.112}$$

Predicción

Finalmente, se pueden aplicar distintas alternativas a las predicciones finales:

- Predecir la clase $l \in \mathcal{Z}$ de mayor probabilidad definida según la distribución q_ω de (4.112), es decir:

$$\begin{array}{rccl} R_1: & \Omega_2 & \longrightarrow & \mathcal{A} \\ & \omega & \longmapsto & R_1(\omega) = [Z = l] \end{array} \tag{4.113}$$

$$\text{con } l = \arg \max_{j \in \{1,\ldots,s\}} \{p_{\omega j}\} \tag{4.114}$$

- La distribución de probabilidad definida por q_ω, es decir:

$$\begin{array}{rccl} R_2: & \Omega_2 & \longrightarrow & \mathcal{A} \\ & \omega & \longmapsto & R_2(\omega) = [Z \sim q_\omega] \end{array} \tag{4.115}$$

Una variante a estas predicciones puede ser la asignación aleatoria de una clase de Z según ley de probabilidad q_ω.

Observaciones incompletas

En el caso de observaciones incompletas debido a los predictores, pueden aplicarse como criterios de aplicaciones auxiliares tanto (4.108) como (4.109). Sin embargo, hay que proporcionar con la predicción el nivel de relación del individuo con el árbol que viene dado en (4.106) o (4.107).

Tanto en el caso de observaciones completas como incompletas, puede ser de utilidad para la interpretación indicar la secuencia ordenada de los niveles de relación de los nodos con el individuo, así como las estimaciones de la variable clase para esos nodos.

Alternativamente, se puede adaptar el criterio de Quinlan (Quinlan, 1990) para falta de observación, estimando con la muestra diseño las probabilidades

de las ramas como los cocientes de los pesos de ellas con respecto al padre y asignando los individuos a ambos nodos hijos con estas probabilidades.

4.6 Calidad del árbol

Una medida de calidad del árbol es una medida de la tasa de predicciones correctas que es 1 menos la estimación de la tasa de error cometida en las predicciones.

4.6.1 Antecedentes

Tradicionalmente para árboles de Segmentación para datos monoevaluados, la estimación de la tasa de error para un nodo t_k depende de la predicción en el nodo. Estas estimaciones de las tasas de error pueden ser estimadas con la muestra diseño, aunque en general representan una subestimación del error.

Sea un árbol T definido por $T = \{t_k = \beta_k \wedge \alpha_k \wedge \mu_k\}_{k=1,\dots,K}$ (véase 3.5) y sea $\alpha_k = [Z \sim q_k]$ con $q_k \equiv (c_1\ p_1^k, ..., c_s\ p_s^k)$, la descripción en $\mathcal{A}$ del nodo t_k. A continuación se especifican las tasas de error, según sea la predicción en los nodos:

1. Si se selecciona como predicción la clase más frecuente, es decir, la clase l tal que $p_l^k = \max_{i=1,\dots,s} p_i^k$, entonces la estimación de la proporción de predicciones correctas en el nodo es p_l^k y la estimación de la tasa de error es $1 - p_l^k$.

2. Si se selecciona la clase l que proporcione la mínima pérdida esperada, es decir, si se supone una función de pérdida que asocia a la clase $i-$ésima la pérdida L_i^k, entonces se selecciona en el nodo la clase l que haga mínima la pérdida esperada, es decir, $p_l^k L_l^k = \min_{i=1,\dots,s} p_i^k L_i^k$ y ésta es la estimación de la tasa de error.

2. Si se asigna probabilísticamente en el nodo una clase según la ley de probabilidad q_k, entonces la proporción de predicciones correctas es $\sum_{i=1,\ldots,s} (p_i^k)^2$ y la estimación de la tasa de error es $1 - \sum_{i=1,\ldots,s} (p_i^k)^2$.

En estos casos, el estimador de la tasa de error en el árbol completo es:

$$TE(T) = \sum_{k=1,\ldots,K} TE(t_k) = \sum_{k=1,\ldots,K} te(t_k)p(t_k) \tag{4.116}$$

con:

- $te(t_k)$ la estimación de la tasa de error en el nodo t_k, según los puntos 1. a 3. anteriores.
- $p(t_k)$ el peso relativo del nodo t_k.

No siempre una medida de calidad se mide por la estimación de la tasa de error o eficiencia del árbol, sino que forman parte de ellas la simplicidad y la asociación. Breiman propone la *tasa de error compleja* que toma en cuenta tanto el número de errores como la complejidad del árbol (véase Breiman et al., 1984).

Las estimaciones de las tasas de error en la predicción se pueden realizar con la muestra diseño con la que se construye el árbol, como en los puntos 1. a 3. anteriores, pero esto lleva a tasas de error optimistas. Por esta razón se proponen en la literatura criterios de parada en la construcción del árbol que no lleven a una sobreestimación de la variable clase Z.

Además, se proponen varios métodos alternativos de estimación de las tasas de error que no lleven a estimaciones tan optimistas y sean tasas de error más fiables. En estos casos se asumen predicciones a una clase. Entre ellos se encuentran los que utilizan una *muestra prueba* independiente para estimar la tasa de error y los que utilizan la muestra completa de forma más eficaz para la estimación de la tasa de error.

La solución más simple es usar una muestra prueba independiente, dividiendo la muestra original en una *muestra diseño* para construir el árbol y una *muestra prueba* que no forma parte de la construcción del árbol para evaluar la calidad del árbol.

Otras técnicas más sofisticadas utilizan la muestra completa de forma más eficiente. Es el caso de los estimadores obtenidos por *validación cruzada* o técnicas de *bootstrap* (Efron y Tibshinari, 1993). Éstas últimas no dan buenos resultados en los árboles de Segmentación (véase Breiman et al., 1984). La técnica de validación cruzada consiste en realizar una partición de la muestra original en L subconjuntos, que constituirán sendas muestras prueba. Se construyen L árboles con el total de la muestra salvo el subconjunto i-ésimo seleccionado que se utiliza como muestra prueba. De esta forma se obtienen L estimaciones de la tasa de error. Como resultado de la tasa de error se proporciona la media o media ponderada de todas estas estimaciones. Esta técnica permite utilizar conjuntos prueba de tamaño reducido, incluso de tamaño 1.

En todos estos casos alternativos que se utilizan muestras prueba, la estimación de la tasa de error se mide por el cociente:

$$TE(T) = \frac{n_t}{n_2} \tag{4.117}$$

con n_2 el número de elementos de la muestra prueba y n_t el número de elementos de la muestra prueba que son mal clasificados.

Otras técnicas, las técnicas de *poda*, en lugar de proponer criterios de parada en la construcción del árbol, construyen el árbol completo y crean una sucesión de subárboles del mismo, según las estimaciones de las tasas de error y seleccionan áquel que optimice alguno de los criterios de calidad.

Antecedentes de la poda de árboles se encuentran en Breiman et al., 1984 (véase Guo y Gelfand, 1992). La técnica consiste en seleccionar un árbol podado del árbol original, minimizando el estimador de la *tasa de error compleja* sobre

una familia paramétrica de subárboles $\{S\}$ podados del árbol original. Las estimaciones de las tasas de error se pueden realizar con la muestra prueba o con el estimador de la validación cruzada. Se define la tasa de error compleja del subárbol podado S como:

$$TE_\tau(S) = TE(S) + \tau|S| \tag{4.118}$$

con $\tau \geqslant 0$ y $|S|$ el número de nodos terminales de S.

La familia de subárboles podados se obtiene como $T(\tau)$, $\tau \geqslant 0$ minimizando:

$$TE_\tau(T(\tau)) = \min_{S \leq T} TE_\tau(S) \tag{4.119}$$

fijado τ. $\{S \ / \ S \leq T\}$ indica el conjunto de subárboles de T. Se tiene que $T(0) = T$ y $T(\tau) = nodo - raíz$ para un τ suficientemente grande. Véase en Breiman et al., 1984 el algoritmo para generar la sucesión finita $T(\tau)$, $\tau \geqslant 0$.

De la familia $\{T(\tau), \tau \geq 0\}$ se selecciona el subárbol $T(\tau^*)$ que minimiza un estimador honesto de la probabilidad de error:

$$\widehat{TE}(T(\tau^*)) = \min_{\tau} \widehat{TE}(T(\tau)) \tag{4.120}$$

Quinlan (Quinlan, 1990) propone una poda pesimista que utiliza únicamente la muestra diseño. Ciampi (Ciampi, 1992) propone además de la poda del árbol una etapa de unión posterior de nodos. Para una extensión de esta sección puede consultarse Safavian y Landgrebe, 1991.

4.6.2 Aproximación para el árbol de datos estratificados

En esta sección se presenta los criterios de parada que se adoptan en el método presentado en esta Memoria, así como las modificaciones que sufre el algoritmo (véase 3.3) cuando se alcanzan los criterios de parada adoptados.

Criterios de parada

Tradicionalmente, los criterios de parada se relacionan:

- con el tamaño de los nodos, impidiendo que el método de Segmentación prosiga por ellos. Shlien, 1992 incluso propone que este número sea 20. Otros autores no toman en consideración este punto (Breiman et al., 1984, Quinlan, 1986).

- con la calidad de predicción estimada mediante la muestra diseño en los nodos, impidiendo que el método de Segmentación prosiga por ellos si la calidad de predicción es la máxima o próxima a la máxima.

- con la simplicidad del árbol, impidiendo árboles excesivamente largos.

En el método presentado para el árbol de datos estratificados, se han propuesto diversos criterios de parada para no realizar una sobrestimación de la variable clase Z. De una parte se toma en consideración:

- que el peso de un nodo no sea de tamaño pequeño. Esto se asegura exigiendo mediante la función de admisibilidad de los cortes que los pesos de los nodos hijos posibles no sean de tamaño pequeño. Véase la condición 2. de 4.1.2 y 4.2.2.

- que el peso de los estratos en un nodo no sea muy pequeño. Esto se asegura mediante la función de admisibilidad de los cortes, y mediante la condición de parada de los estratos. Véase las condiciones 3. y 4. de 4.1.2 y 4.2.2 y véase 4.1.6 y 4.2.6.

- el proceso de Segmentación termina en un nodo explorable si todos los estratos del mismo entran a formar parte de uno o varios nodos decisionales, al tener una medida de contenido de información del nodo y estrato para la variable clase *alta*. Véase 3.3.4, 4.1.5 y 4.2.5. Esto asegura no proseguir

el proceso recursivo en estratos que predicen una clase con poca o nula incertidumbre.

Además, se proponen los siguientes *criterios de parada* para un nodo explorable no expresados anteriormente:

- Permitir un nivel de profundidad máximo en el árbol ($l\max$), con lo que además se asegura una mayor interpretabilidad de las reglas que se obtienen.
- No permitir que el incremento relativo de la medida de contenido de información del nuevo árbol una vez explorado el nodo, con respecto al anterior árbol, sea menor que un umbral ρ determinado.

Para los nodos explorables que alcanzan los criterios de parada anteriores, se propone dividirlos en dos nodos terminales. Si de estos nodos explorables se han obtenido previamente nodos terminales por haberse satisfecho la condición de parada para algunos estratos, entonces antes de dividirlos en dos, se incorporan estos nodos terminales al nodo explorable del que se han escindido.

Por tanto, se debe establecer la condición que determina cuáles son los estratos que conforman cada uno de los nodos terminales en los que se escinde el nodo. Sea

$$\begin{array}{rll} Terminal_\sigma: & \mathcal{B}\hat{\wedge}\mathcal{N}\times E & \longrightarrow \{0,1\} \\ & (\beta_r\wedge\mu_r, S_i) & \longmapsto Terminal_\sigma(\beta_r\wedge\mu_r, S_i) \end{array} \tag{4.121}$$

la función de verdad que aporta la condición que debe verificar un estrato i de un nodo explorable r para formar uno de los nuevos nodos. El conjunto de indicadores de estratos que forman parte del otro nodo es:

$$\{i\in S^r / Terminal_\sigma(\beta_r\wedge\mu_r, S_i)=0\} \tag{4.122}$$

siendo $\mu_r=[M\in S^r]$.

Si el número de clases es $s = 2$, como la mayoría de los algoritmos de Segmentación desarrollados en la literatura, al dividir un nodo en dos, el criterio que se adopta es asignar a cada uno de ellos los estratos que estiman una probabilidad de predicción mayor a cada una de las clases de la variable Z. Es decir, se adopta el criterio:

$$Terminal_{\sigma}(\beta_r \wedge \mu_r, S_i) = 1, \text{ si } p_1^{ri} < 0.5$$

para $i \in S^r$ y siendo

$$\mu_r = [M \in S^r] \text{ y } \alpha_r^i = Calc_{\alpha}(\beta_r \wedge [M = i]) = [Z \sim (c_1\, p_1^{ri}, c_2\, (1 - p_1^{ri})]$$

Nivel de profundidad máximo

El algoritmo general presentado en 3.3 se modifica para incorporar la condición de *máximo nivel de profundidad* de un nodo modificando el paso 1 del mismo de *evaluación de la condición de admisibilidad y actualización del conjunto de nodos explorables,* detallado en 3.3.2. Se presenta a continuación, la nueva redacción del paso 1 del algoritmo que incorpora, como verificación inicial, la condición de máximo nivel de profundidad de los nodos explorables y en el caso que se satisfaga, entonces divide el nodo explorable en dos nodos terminales. En cursiva se incluyen las sentencias que no cambian con respecto a 3.3.2. La nueva redacción es:

Para cada nodo explorable $r \in X$:

Si el nivel de profundidad del nodo es igual a l max, entonces se aplica el proceso *Terminal-Divide* al nodo r.

Proceso Terminal-Divide:

Si el nodo r tiene como hijo de la iteración anterior un nodo terminal al haberse

satisfecho la condición de parada para algunos estratos (en 3.3.5), éstos se incorporan al nodo r antes de aplicar el proceso de división. Esto implica[7] la actualización del número de nodos (al eliminar uno), del árbol T y del conjunto N que elimina el nodo terminal y actualiza el nodo explorable r, la actualización de r y del conjunto S^r. Posteriormente se aplica el proceso de división siguiente:

Sea:

$$D^{r_1} = \{i \in S^r | Terminal_\sigma(\beta_r \wedge \mu_r, S_i) = 1\} \tag{4.123}$$

el conjunto de indicadores de estratos del nodo r que se escinden de él para formar uno de los dos nodos terminales. Sea:

$$D^{r_2} = S^r - D^{r_1} \tag{4.124}$$

el resto de indicadores de estratos del nodo r que componen el otro nodo terminal.

El nodo r deja de ser explorable:

[D1] $X \leftarrow X - \{r\}$

Si alguno de los conjuntos D^{r_1} o D^{r2} es vacío, entonces el nodo r pasa a ser terminal. Si ambos son no vacíos, entonces el nodo r se divide en dos nodos terminales. A continuación se describen los nuevos nodos:

[D2] Si $D^{r_1} \neq \emptyset$ y $D^{r_2} \neq \emptyset$, entonces $K \leftarrow K + 1$

[D3] Si $D^{r_1} \neq \emptyset$, entonces $\mu_{D^{r_1}} = [M \in D^{r_1}]$, $\alpha_{D^{r_1}} = Calc_\alpha(\beta_r \wedge \mu_{D^{r_1}})$

y $T \leftarrow T \cup \{\beta_r \wedge \alpha_{D^{r_1}} \wedge \mu_{D^{r_1}}\}$, $N \leftarrow N \cup \{\beta_r \wedge \alpha_{D^{r_1}} \wedge \mu_{D^{r_1}}\}$

[D4] Si $D^{r_2} \neq \emptyset$, entonces $\mu_{D^{r_2}} = [M \in D^{r_2}]$, $\alpha_{D^{r_2}} = Calc_\alpha(\beta_r \wedge \mu_{D^{r_2}})$

y $T \leftarrow T \cup \{\beta_r \wedge \alpha_{D^{r_2}} \wedge \mu_{D^{r_2}}\}$, $N \leftarrow N \cup \{\beta_r \wedge \alpha_{D^{r_2}} \wedge \mu_{D^{r_2}}\}$

[7] Se omite deliberadamente la formalización completa de esta actualización, debido a las modificaciones formales que implicaría en el proceso general del algoritmo presentado en 3.3.

Se añade la nueva descripción del árbol a la sucesión de árboles obtenidos $\mathcal{T}$ y se calcula el nuevo valor de medida de contenido de información del árbol con respecto a Ω:

[D5] $\mathcal{T} \leftarrow \mathcal{T} \cup T$

[D5] Se calcula $IC\{T, \Omega\}$

Fin del proceso Terminal-Divide

Si no, entonces *se comprueba la condición de admisibilidad para los elementos de B. Si no existen cortes admisibles desde el nodo r, éste deja de ser explorable. Es decir, $X \leftarrow X - \{r\}$.*

Fin

Y finalmente, en ambos casos, *si el conjunto de nodos explorables queda vacío entonces se termina el proceso recursivo.*

- *Si $X = \emptyset$ entonces* ***Salida del algoritmo*** *al ser el conjunto de nodos explorables el conjunto vacío, $\emptyset$.*

- *Si $X \neq \emptyset$, entonces $K \leftarrow K + 1$. Es decir, se incrementa en uno el número de nodos del árbol, ya que se realiza la división en dos de uno de los nodos explorables en 3.3.3.*

Incremento de contenido de información mínimo

El algoritmo general presentado en 3.3 se modifica para incorporar la condición de *mínimo incremento relativo de contenido información* modificando el final del mismo detallado en *Exploración de otro nodo o final del algoritmo* en 3.3.5. En la modificación del algoritmo se hace uso de los símbolos T_{ant}, $\mathcal{N}_{ant}$, K_{ant} y $\mathcal{T}_{ant}$ para indicar, respectivamente, el árbol obtenido en la iteración anterior, el conjunto de nodos tal y como estaba al final de la iteración anterior, el número

de nodos del árbol T_{ant} y la secuencia de árboles obtenidos hasta la iteración anterior. En la especificación completa del algoritmo, estos conjuntos y número son necesarios para hacer referencia al árbol obtenido en la iteración anterior a la actual, ya que en el caso que se alcance la condición de mínimo incremento de contenido de información, se parte del árbol de la iteración anterior para dividir el nodo estudiado en dos. Si de este nodo tuviera un hijo nodo terminal de la iteración anterior por satisfacerse la condición de parada de algunos estratos en 3.3.5, entonces, previamente a su división se incorporarían a este nodo los estratos del nodo terminal.

El final del algoritmo queda entonces como sigue (en cursiva se incluyen las sentencias que no cambian con respecto a 3.3.5):

Exploración de otro nodo o final del algoritmo.

Se calcula el nuevo valor de medida de contenido de información del árbol T con respecto a Ω:

Se calcula $IC\{T, \Omega\}$

Si es la iteración inicial, entonces $\mathcal{T} \leftarrow \mathcal{T} \cup T$

Si no es la iteración inicial, entonces sea

$$\Delta IC = \frac{|IC\{T, \Omega\} - IC\{T_{ant}, \Omega\}|}{|IC\{T_{ant}, \Omega\}|} \tag{4.125}$$

el incremento relativo de la medida de contenido de información de esta iteración del algoritmo con respecto a la anterior.

Si se cumple la condición de mínimo incremento relativo de contenido de información, entonces se vuelve al árbol de la iteración anterior y el nodo r^+ obtenido en 3.3.3 se divide en dos. En caso contrario, *se añade la nueva descripción del árbol a la sucesión de árboles obtenidos $\mathcal{T}$:*

Si $\Delta IC < \rho$, entonces

Se vuelve al árbol de la iteración anterior, es decir,

[D0] $T \leftarrow T_{ant}$, $\mathcal{N} \leftarrow \mathcal{N}_{ant}$, $K \leftarrow K_{ant}$, $\mathcal{T} \leftarrow \mathcal{T}_{ant}$

Al nodo r^+ obtenido en 3.3.3, se le aplica el proceso Terminal-Divide[8] especificado en la página 216, excluida la sentencia [D1].

Si no, entonces $\mathcal{T} \leftarrow \mathcal{T} \cup T$.

Fin (de comprobación de la condición de ΔIC)

Fin (de la condición de iteración no inicial)

Y finalmente, en todas las iteraciones, se almacena la información relativa al árbol en esta iteración del algoritmo y se comprueba si el conjunto de nodos explorables es no vacío para continuar con otra iteración del algoritmo en *3.3.2.*

[D6] $T_{ant} \leftarrow T$, $\mathcal{N}_{ant} \leftarrow \mathcal{N}$, $K_{ant} \leftarrow K$, $\mathcal{T}_{ant} \leftarrow \mathcal{T}$

- *Si* $X \neq \emptyset$ *entonces ir a 3.3.2*
- *si no* ***Salida del algoritmo.***

Nota 4.3 *Como mejora del algoritmo presentado en 3.3, el proceso Terminal-Divide se aplica también a los nodos explorables que en 3.3.2 dejan de serlo por no tener cortes admisibles. En este caso, se aplica el proceso Terminal-Divide de página 216 excluida la sentencia [D1]. Esta mejora se aplica independientemente de la condición de nivel de profundidad máximo.*

Nota 4.4 *En el caso que se incorporen simultáneamente al algoritmo, la condición de mínimo incremento relativo de contenido de información y el proceso Terminal-Divide, entonces se debe añadir al proceso Terminal-Divide de la página 216, la sentencia [D6] de almacenamiento de la información relativa al árbol en la iteración actual del algoritmo, es decir:*

[8] Se excluye la sentencia [D1] ya que en 3.3.2 se hace r^+ no explorable y la información relativa a nodos explorables no se actualiza en [D0].

[D6] $T_{ant} \leftarrow T$, $\mathcal{N}_{ant} \leftarrow \mathcal{N}$, $K_{ant} \leftarrow K$, $\mathcal{T}_{ant} \leftarrow \mathcal{T}$

antes de Fin del proceso Terminal-Divide. La vuelta a un árbol de una iteración anterior, sólo se realiza después de una iteración completa del método.

En las nuevas aproximaciones del método, se verifican los resultados presentados en 4.3. En particular, las proposiciones 4.13 y 4.18 de no disminución de la medida de contenido de información de una iteración a la siguiente. Además, la condición de incremento de contenido de información mínimo a un umbral ρ, asegura al menos este incremento relativo de una iteración a otra.

Estimación de la tasa de error

La estimación de la tasa de error se puede proporcionar con una muestra prueba o por validación cruzada. En los casos en los que el criterio de predicción proporciona una clase, la estimación de la tasa de error de cada muestra prueba se mide por el cociente (4.117), es decir, $TE(T) = \frac{n_t}{n_2}$ con n_2 el número de elementos de la muestra prueba y n_t el número de elementos de la muestra prueba que son mal clasificados. Las muestras prueba referidas pueden ser una muestra prueba propiamente dicha o cada una de las muestras prueba utilizadas en el procedimiento de la validación cruzada.

Del mismo modo, se define la tasa de error para un estrato $i \in \{1, ..., m\}$, relativa a una muestra prueba como el cociente:

$$TE_i(T) = \frac{n_{ti}}{n_{2i}} \tag{4.126}$$

con n_{2i} el número de elementos ω de la muestra prueba tales que $M(\omega) = i$ y n_{ti} el número de estos elementos que son mal clasificados.

Estas estimaciones pueden realizarse por validación cruzada que necesitan menor tamaño de muestra prueba. En el caso de la estimación de tasas de error para el árbol para datos estratificados, los L subconjuntos que constituyen

sendas muestras prueba en la validación cruzada deben contener elementos de todos los estratos cuando sea posible. Es recomendable que el tamaño de los subconjuntos sea mayor o igual que el número de estratos, m. Se recomienda que el tamaño de cada muestra prueba en la estimación por validación cruzada sea aproximadamente igual a $\frac{n_2}{L}$, y que el número de elementos del estrato $i-$ésimo en la muestra prueba sea aproximadamente igual a $\frac{Card(S_i)}{L}$.

En el caso de utilización de una única muestra prueba para estimar las tasas de error, se recomienda que el número de elementos de los estratos sea proporcional a los elementos totales del conjunto de individuos originales.

4.7 Extensiones del método

Las extensiones del método permiten que la función $Ent(.)$ (en (4.5) y (4.38)) que es una medida de incertidumbre en las clases en los nodos del árbol pueda ser sustituida por otras medidas de incertidumbre (véase 2.1.7 y 2.2.3). Otras extensiones del método permiten la incorporación de un peso en los individuos de partida y de probabilidades 'a priori' de las clases.

4.7.1 Peso en los individuos

La incorporación de pesos en los individuos es posible en la formulación del método propuesto. Los datos recogidos en las Oficinas de Estadística en muchas ocasiones contienen ponderaciones de las observaciones que recogen. En general, estas observaciones con sus pesos respectivos son representativas de la población a la que pertenecen[9].

Sean $Q(\omega)$ y $q(\omega)$ para $\omega \in \Omega$, los pesos absoluto y relativo, respectivamente,

[9]En el caso de que los objetos simbólicos sean obtenidos a partir de grupos de individuos descritos por datos monoevaluados ponderados, los pesos de los individuos originales pueden repercutirse en los datos simbólicos obtenidos (véase Stéphan et al., 2000). El método propuesto permite que, además, estos datos simbólicos puedan a su vez ser ponderados.

de los elementos de Ω, es decir,

$$q(\omega) = \frac{Q(\omega)}{\sum\limits_{\omega\in\Omega} Q(\omega)} \tag{4.127}$$

En este caso, los cardinales de un estrato $i \in \{1, ..., m\}$ y del conjunto Ω se asumen:

$$Card(S_i) = \sum_{\omega\in\Omega} [M = i](\omega)Q(\omega) \tag{4.128a}$$

$$Card(\Omega) = \sum_{\omega\in\Omega} Q(\omega) \tag{4.128b}$$

En conjuntos de datos representativos de la población total, $q(\omega)$ estima la probabilidad del subconjunto de la población que el elemento $\omega \in \Omega$ representa.

Datos monoevaluados

La estimación de $P(\beta \wedge \mu)$ en (4.5), (4.10) y (4.11) es:

$$P(\beta \wedge \mu) = \sum_{\omega\in\Omega} \beta \wedge \mu(\omega)q(\omega) \tag{4.129}$$

para un elemento $\beta\wedge\mu \in \mathcal{B}\hat{\wedge}\mathcal{N}$. En realidad (4.6) es un caso particular de (4.129) que sustituye $q(\omega)$ por $\frac{1}{Card(\Omega)}$.

La estimación de $P(S_i|\beta_r \wedge b \wedge \mu_r)$ de (4.11) es:

$$P(S_i|\beta_r \wedge b \wedge \mu_r) = \begin{cases} \frac{\sum\limits_{\omega\in\Omega} \beta_r\wedge b\wedge[M=i](\omega)q(\omega)}{\sum\limits_{\omega\in\Omega} \beta_r\wedge b\wedge\mu_r(\omega)q(\omega)} & , \text{ Si } i \in S^r \\ 0 & , \text{ Si } i \notin S^r \end{cases} \tag{4.130}$$

y de forma similar para $P(S_i|\beta_r \wedge b^c \wedge \mu_r)$, siendo $\mu_r = [M \in S^r]$.

La estimación de las probabilidades de las clases p_l^k de (4.22) en un nodo k

son:

$$p_l^k = P([Z=l]|\beta_k \wedge \mu_k) = \frac{\sum\limits_{\omega\in\Omega} \beta_k \wedge [Z=l] \wedge \mu_k(\omega) q(\omega)}{\sum\limits_{\omega\in\Omega} \beta_k \wedge \mu_k(\omega) q(\omega)}, l \in \{1,\ldots,s\} \quad (4.131)$$

Se tiene el corolario 4.3.

Nota 4.5 *Dado que el término* $q(\omega) = \frac{Q(\omega)}{\sum\limits_{\omega\in\Omega} Q(\omega)}$ *es común en todas las estimaciones de IC y de EIC, éste puede sutituirse por el término* $Q(\omega)$ *en las estimaciones correspondientes de ambas medidas, ya que los elementos que se obtienen de la optimización son los mismos.*

La condición de admisibilidad (2) de 4.1.2 queda modificada en:

- 2. $\min\{\sum\limits_{\omega\in\Omega} \beta_r \wedge b \wedge \mu_r(\omega)Q(\omega), \sum\limits_{\omega\in\Omega} \beta_r \wedge b^c \wedge \mu_r(\omega)Q(\omega)\} < \nu$ con $\nu \geq 5$. Es decir, si el peso de uno de los nodos hijos es pequeño. El **peso del nodo** t_k, descrito en $\mathcal{B}\hat{\wedge}\mathcal{N}$ por $\beta_k \wedge \mu_k$ es

$$\sum_{\omega\in\Omega} \beta_k \wedge \mu_k(\omega)Q(\omega) \quad (4.132)$$

La condición de parada desde un nodo k para un estrato $S_i \in E$ de 4.1.6 queda modificada de la siguiente forma. La función $Stop_\tau(.)$ de (3.22) se caracteriza por:

$$Stop_\tau(\beta_k \wedge \mu_k, S_i) = 1 \iff \sum_{\omega\in S_i} \beta_k(\omega)Q(\omega) < \tau, \text{para } i \in S^k \quad (4.133)$$

siendo $\mu_k = [M \in S^k]$.

La expresión $\sum\limits_{\omega\in S_i} \beta_k(\omega)Q(\omega)$ es el peso del nodo y estrato.

La importancia relativa w_k^i de un nodo k en la descripción de un estrato S_i,

con $t_k(\mathcal{N}) = [M \in S^k]$ de (4.93) es, ahora, para $i \in S^k$:

$$w_k^i = \frac{\sum\limits_{\omega \in S_i} \beta_k(\omega) Q(\omega)}{Card(S_i)} = \frac{\sum\limits_{\omega \in S_i} \beta_k(\omega) q(\omega)}{\sum\limits_{\omega \in S_i} q(\omega)} \tag{4.134}$$

La contribución absoluta wa_i^k de un estrato S_i al nodo k, con $t_k(\mathcal{N}) = [M \in S^k]$ de (4.100) es ahora para $i \in S^k$:

$$wa_i^k = \frac{\sum\limits_{\omega \in S_i} \beta_k(\omega) Q(\omega)}{\sum\limits_{\omega \in \Omega} \beta_k \wedge \mu_k(\omega) Q(\omega)} = \frac{\sum\limits_{\omega \in S_i} \beta_k(\omega) q(\omega)}{\sum\limits_{\omega \in \Omega} \beta_k \wedge \mu_k(\omega) q(\omega)} \tag{4.135}$$

Para $i \notin S^k$, ambas son nulas.

Se tienen las proposiciones y corolarios de 4.1, 4.3.1 y 4.3.2. Se obtienen particiones de elementos de Ω cuyos pesos para los nodos representados por $\beta_k \wedge \mu_k$ en $\mathcal{B} \times \mathcal{N}$ se obtienen como en (4.132). La predicción se realiza como en 4.5. En cuanto a las tasas de error de un estrato $i \in \{1, ..., m\}$ en una muestra prueba, éstas tienen la expresión de (4.126) con n_{2i} la suma de los pesos absolutos de los elementos ω de la muestra tales que $M(\omega) = i$ y n_{ti} la suma de los pesos absolutos de estos elementos que son mal clasificados. La tasa de error global tiene las mismas transformaciones.

Datos modales probabilistas

La estimación de $P(\beta \wedge \mu)$ en (4.38), (4.47) y (4.43) es**:**

$$\begin{aligned} P(\beta \wedge \mu) &= \sum_{\omega \in \Omega} \beta \wedge \mu(\omega) q(\omega) = \sum_{\omega \in \Omega} \beta(\omega) \mu(\omega) q(\omega) \\ &= \sum_{\omega | M(\omega) \in S^M} \Pr ob(Y \in D | \omega) q(\omega) \end{aligned} \tag{4.136}$$

para un elemento $\beta \wedge \mu = [Y \in D] \wedge [M \in S^M] \in \mathcal{B} \hat{\wedge} \mathcal{N}$. En realidad, (4.39) es un caso particular de (4.136) que sustituye $q(\omega)$ por $\frac{1}{Card(\Omega)}$.

La expresión de $P(S_i|\beta_r \wedge b \wedge \mu_r)$ y de $P(S_i|\beta_r \wedge b \wedge \mu_r)$ en (4.43) es similar a (4.130). Es de aplicación la nota 4.5.

La estimación de las probabilidades de las clases p_l^k de (4.51) en un nodo k tienen la expresión (4.131).

La condición de admisibilidad 2. de 4.2.2 y la condición de parada de los estratos de 4.2.6 queda modificada de igual modo que en la subsección anterior.

La importancia relativa w_k^i de un nodo k en la descripción de un estrato S_i, con $t_k(\mathcal{N}) = [M \in S^k]$ de (4.97) tiene ahora para $i \in S^k$ la expresión (4.134) y la contribución absoluta wa_i^k de un estrato S_i al nodo k, para $i \in S^k$ la expresión (4.135). Para $i \notin S^k$, ambas son nulas.

Se tienen las proposiciones y corolarios de 4.2, 4.3.1 y 4.3.3. Se obtienen particiones con incertidumbre de elementos de Ω cuyos pesos para los nodos, representados por $\beta_k \wedge \mu_k$ en $\mathcal{B} \times \mathcal{N}$ se obtienen como en (4.132). La predicción se realiza como en 4.5. En cuanto a las tasas de error tienen las mismas transformaciones que en la subsección anterior.

4.7.2 Probabilidades 'a priori' de las clases

Probabilidades comunes a los estratos

En caso de muestreo en cada una de las clases o en cualquier otro tipo de muestreo si se proporcionan las probabilidades *a priori* $(\pi_1, ..., \pi_s)$ de las clases, entonces, la estimación de las probabilidades de las clases en un nodo k se determinan de la siguiente forma:

- En el caso de datos monoevaluados, p_l^k de (4.22), como:

$$p_l^k = \frac{Card(Ext_\Omega(\beta_k \wedge [Z = l] \wedge \mu_k))\pi_l}{\sum_{i=1,..s} Card(Ext_\Omega(\beta_k \wedge [Z = i] \wedge \mu_k))\pi_i}, l = 1, ..., s \qquad (4.137)$$

- En el caso de datos modales probabilistas, p_l^k de (4.51), como:

$$p_l^k = \frac{\sum_{\omega\in\Omega} \beta_k \wedge [Z=l] \wedge \mu_k(\omega)\pi_l}{\sum\limits_{\omega\in\Omega}\sum\limits_{i=1,..s} \beta_k \wedge [Z=i] \wedge \mu_k(\omega)\pi_i}, l = 1, ..., s \tag{4.138}$$

Probabilidades diferentes en cada estrato

En caso de muestreo en cada una de las clases o en cualquier otro tipo de muestreo, si se proporcionan las probabilidades *a priori* $(\pi_1^e, ..., \pi_s^e)$ de las clases en los estratos $e \in \{1, ..., m\}$, entonces la estimación de las probabilidades de las clases en un nodo k se determinan:

- En el caso de datos monoevaluados, p_l^{ke} de (4.24) como

$$p_l^{ke} = \frac{Card(Ext_{S_e}(\beta_k \wedge [Z=l]))\pi_l^e}{\sum\limits_{i=1,..s} Card(Ext_{S_e}(\beta_k \wedge [Z=i]))\pi_i^e}, l = 1, ..., s \tag{4.139}$$

  Y p_l^k de (4.22), como:

$$p_l^k = \sum_{e\in S^k} \frac{Card(Ext_{S_e}(\beta_k))}{Card(Ext_\Omega(\beta_k \wedge \mu_k))} p_l^{ke}, l = 1, ..., s \tag{4.140}$$

  con $\mu_k = [M \in S^k]$ y p_l^{ke} de (4.139).

- En el caso de datos modales probabilistas, p_l^{ke} de (4.53) como

$$p_l^{ke} = \frac{\sum_{\omega\in S_e} \beta_k \wedge [Z=l](\omega)\pi_l^e}{\sum\limits_{\omega\in S_e}\sum\limits_{i=1,...,s} \beta_k \wedge [Z=i](\omega)\pi_i^e}, l = 1, ..., s \tag{4.141}$$

  Y p_l^k de (4.51), como:

$$p_l^k = \sum_{e\in S^k} \frac{\sum\limits_{\omega\in\Omega} \beta_k \wedge [M=e](\omega)}{\sum\limits_{\omega\in\Omega} \beta_k \wedge \mu_k(\omega)} p_l^{ke}, l = 1, ..., s \tag{4.142}$$

con $\mu_k = [M \in S^k]$ y p_l^{ke} de (4.141).

4.7.3 Peso en los individuos y probabilidades 'a priori' de las clases

Las medidas de IC e EIC se expresan como en 4.7.1. Se tienen las consideraciones de 4.7.1.

Probabilidades comunes a los estratos

Sean $(\pi_1, ..., \pi_s)$ las probabilidades *a priori* de las clases. La estimación de las probabilidades de las clases p_l^k de (4.22) y (4.51) en un nodo k se determinan por la expresión:

$$p_l^k = \frac{\sum\limits_{\omega\in\Omega} \beta_k \wedge [Z = l] \wedge \mu_k(\omega) q(\omega) \pi_l}{\sum\limits_{\omega\in\Omega} \sum\limits_{i=1,...,s} \beta_k \wedge [Z = i] \wedge \mu_k(\omega) q(\omega) \pi_i}, l = 1, ..., s \tag{4.143}$$

Probabilidades diferentes en cada estrato

Sean $(\pi_1^e, ..., \pi_s^e)$ las probabilidades *a priori* de las clases en los estratos $e \in \{1, ..., m\}$. La estimación de las probabilidades de las clases de (4.24) y (4.53) en un nodo k para el estrato e se determinan por:

$$p_l^{ke} = \frac{\sum_{\omega\in\Omega} \beta_k \wedge [Z = l] \wedge [M = e](\omega) q(\omega) \pi_l^e}{\sum\limits_{\omega\in\Omega} \sum\limits_{i=1,...,s} \beta_k \wedge [Z = i] \wedge [M = e](\omega) q(\omega) \pi_i^e}, l = 1, ..., s \tag{4.144}$$

Y las probabilidades de las clases p_l^k de (4.22) y (4.51), tienen la expresión:

$$p_l^k = \sum_{e\in S^k} \frac{\sum\limits_{\omega\in\Omega} \beta_k \wedge [M = e](\omega) q(\omega)}{\sum\limits_{\omega\in\Omega} \beta_k \wedge \mu_k(\omega) q(\omega)} p_l^{ke}, l = 1, ..., s \tag{4.145}$$

con $\mu_k = [M \in S^k]$ y p_l^{ke} expresado como (4.144).

4.8 Extension del método a otros datos simbólicos

En el capítulo 3, se presenta el algoritmo de Segmentación para datos estratificados de forma genérica. La formalización presentada es trasladable a otros datos simbólicos y a otras formas de representación de la incertidumbre. En esta sección se sugieren desde un punto de vista global algunas generalizaciones de los criterios expuestos en 4.1 y 4.2 para datos monoevaluados y modales probabilistas, respectivamente, a otros tipos de datos simbólicos para los predictores y también extensión a clases difusas. También se sugieren algunas generalizaciones de objetos simbólicos que representen los nodos del árbol para los nuevos datos simbólicos. En el capítulo 1 se han definido los objetos y datos simbólicos que se especifican en esta sección. En 2.2.3, se presentan algunos antecendentes de tratamiento de la incertidumbre en Segmentación y que son referenciados aquí.

El árbol de Segmentación para datos estratificados se puede representar genéricamente como en (3.5) y (3.6). Los datos de entrada del método generalizado pueden representar otros tipos de incertidumbre (véase 1.3.4 y 1.4.4) u otros tipos de datos simbólicos, como por ejemplo, los datos multievaluados para variables categóricas (véase 1.3.2) o los datos de intervalo para las variables continuas (véase 1.4.4). Del mismo modo que los datos simbólicos de entrada pueden ser de diferente tipo, las aserciones que representan los nodos pueden ser de distinto tipo a las presentadas en 4.1 y 4.2 y/o ser definidas por otras relaciones de dominio y/o otras funciones de combinación de niveles de relación diferentes a las de 4.1 y 4.2. Éste puede ser el caso de las aserciones posibilistas o difusas, de creencia, con datos de intervalo, otros tipos de aserciones probabilistas, etc...

En la nueva representación del árbol, los conjuntos B (de (3.13)) y $\mathcal{A}$ (de (3.3)) pueden ser definidos por otros tipos de variables y datos simbólicos. Los conjuntos $\mathcal{B}$ y $\mathcal{N}$, siguen las definiciones de (3.2) y (3.4), respectivamente. Las

funciones de combinación de niveles de relación en $\mathcal{B}$ pueden cambiar así mismo. Deben estar definidas las funciones de combinación de niveles de relación entre elementos de los conjuntos $\mathcal{A}$, $\mathcal{B}$ y $\mathcal{N}$ (véase definición 1.26 y Diday, 1991). Se asume en lo sucesivo que la función de combinación de niveles de relación de los elementos de $\mathcal{N}$ con elementos de $\mathcal{B}$ es el producto.

Si varios eventos de B hacen referencia al mismo predictor, para que estén definidos los elementos de $\mathcal{B}$, es necesaria la definición de la conjunción de eventos relativos al mismo predictor (véase definición 1.27)[10] y en particular, la intersección de los elementos del conjunto de descripciones de los eventos de B. También es necesario el evento y la aserción vacías en los predictores. Antecedentes a las definiciones necesarias pueden encontrarse en Diday, 1991.

En aras de una generalización de criterios, la función IC de (3.18) se define para $T = \{\beta_k \wedge \alpha_k \wedge \mu_k\}_{k=1,\dots,K}$, como:

$$\begin{array}{lll} IC: & \mathcal{T}(\subset \mathcal{P}(\mathcal{B}\hat{\wedge}\mathcal{A}\hat{\wedge}\mathcal{N})) & \longrightarrow R \\ & T & \longmapsto IC\{T,\Omega\} := -\sum_{k=1}^{K} P(\beta_k \wedge \mu_k) Ent(Z|\beta_k \wedge \mu_k) \end{array} \tag{4.146}$$

La función $Ent(Z|\beta_k \wedge \mu_k)$ es una función que mide la incertidumbre o variabilidad de la estimación de la variable clase en el nodo k. Puede ser una medida de entropía o incertidumbre, medida de entropía o incertidumbre difusa o una medida de entropía o incertidumbre aplicada a distribuciones de posibilidad.

[10] En ocasiones, puede ser conveniente redefinir la definición 1.27 al caso general propuesto por Diday (Diday, 1993). O, más concretamente que la conjunción de eventos sobre la misma variable sea el evento asociado a la unión de las descripciones de los eventos. Éste puede ser el caso asociado a las variables difusas o posibilistas de la tabla 4.2, cuando la función $g(.)$ definida en $\mathcal{B}$ es una $T-$conorma, señaladas en la tabla con [1]. La unión e intersección de conjuntos difusos (tercera fila) se definen por funciones de pertenencia asociadas a una $T-$norma y $T-$conorma, respectivamente.

$P(\beta_k \wedge \mu_k)$ puede ser:

$$P(\beta_k \wedge \mu_k) = \sum_{\omega \in \Omega} \beta_k \wedge \mu_k(\omega) \tag{4.147}$$

que representa la suma de los niveles de relación de los elementos $\omega \in \Omega$ a la representación de los nodos en $\mathcal{B}\hat{\wedge}\mathcal{N}$. Es decir, el *peso* del nodo t_k, entendido por peso la suma de los niveles de relación de la descripción del nodo con los individuos. Puede ser el caso de una *cardinalidad difusa*. En este caso, si β_k es una aserción difusa, entonces $\beta_k \wedge \mu_k(\omega)$ es el grado de pertenencia de un individuo ω a un nodo difuso t_k del árbol. Antecedentes de este tratamiento de la incertidumbre, aunque sin la incorporación de la información de los estratos, pueden encontrarse en Séchet, 1995 y Verde, 1995 (véase 2.2.3). Se pueden generar particiones difusas en el sentido de Ruspini o no (véase 2.2.2).

O, bien $P(\beta_k \wedge \mu_k)$ puede ser:

$$P(\beta_k \wedge \mu_k) = \max_{\omega \in \Omega} \beta_k \wedge \mu_k(\omega) \tag{4.148}$$

si los datos de entrada en los predictores se expresan por distribuciones de posibilidad[11] sobre los conjuntos $\mathcal{Y}_j$. El valor $\beta_k \wedge \mu_k(\omega)$ es la posibilidad del nodo k

[11] Los predictores Y_j son variables modales posibilistas (definición 1.7):

$$\begin{array}{rl} Y_j: & \Omega \longrightarrow \mathcal{M}^{pos}(\mathcal{Y}_j) \\ & \omega \longmapsto (1\ q_\omega(1), ..., l_j\ q_\omega(l_j)) \end{array}$$

con $\mathcal{Y}_j = \{1, ..., l_j\}$.

Los elementos de B son de la forma $[Y_j \sim z]$ o de la forma $[Y_j \sim D_j]$, con $z \in \mathcal{Y}_j, D_j \in \mathcal{P}(\mathcal{Y}_j)$. Estos objetos simbólicos son de la forma:

$$[Y_j \sim z](\omega) = q_\omega(z)$$
$$[Y_j \sim D_j\}](\omega) = \max_{z \in D_j} q_\omega(z)$$

(véase (1.79) y (1.80)).

La función $g(.)$ de combinación de niveles de relación de los elementos de $\mathcal{B}$ puede ser el mínimo u otra $T-$norma. Es posible que en algunas aplicaciones pueda tener sentido que sea el máximo o una $T-$conorma. En este caso, las propiedades de la función $g(.)$ determinadas en (1.51) y (1.52), deben modificarse.

para el individuo ω y el valor $P(\beta_k \wedge \mu_k)$ representa la *posibilidad* del nodo. El operador máximo en (4.148) puede sustituirse por otra $T-$conorma.

O, bien $P(\beta_k \wedge \mu_k)$ puede ser en lugar de (4.147):

$$P(\beta_k \wedge \mu_k) = \sum_{\omega \in \Omega} \beta_k \wedge \mu_k(\omega) q(\omega) \tag{4.149}$$

con $q(.)$ una función de ponderación relativa de los elementos $\omega \in \Omega$ (véase 4.7). En estos casos, es necesario considerar si para la semántica de nodos y datos tiene sentido que $P(.)$ aplique la ponderación de los individuos a los niveles de relación de los individuos con los nodos o debiera aplicarse de forma diferente.

Las extensiones más directas del método presentado en esta Memoria son las que consideran los predictores como variables multievaluadas o como variables de intervalo. Se puede distinguir:

- Si los individuos ω están descritos por datos multievaluados, el conjunto B se define de forma análoga a (4.35) definiendo la relación $\sim$ entre la descripción multievaluada de un individuo en un predictor y la descripción multievaluada de un elemento $b \in B$ como la proporción de los datos multievaluados que se encuentran en la definición de b (véase (1.68)).

  Esta descripción por datos multievaluados de los individuos es equivalente a considerar una descripción por datos modales probabilistas considerando una distribución uniforme de probabilidad sobre el conjunto de datos que define el dato multievaluado del individuo. Con lo que este caso, es un caso particular del método para datos modales probabilistas (véase 4.2). Entonces B es (4.35).

- Si los individuos ω están descritos por datos de intervalo, el conjunto B se define con elementos del tipo $[Y_j < c_j]$ y $[Y_j \geq c_j]$, o sus equivalentes respectivos $[Y_j \sim (-\infty, c_j)]$ y $[Y_j \sim [c_j, \infty)]$, definiendo la relación entre la descripción de intervalo de un individuo en un predictor y la descripción de

intervalo de un elemento $b \in B$ como la proporción de los datos de intervalo del individuo que se encuentran en la definición de b (visto como una extensión a datos de intervalo de (1.68)). En este caso, se encuentran definidos el evento vacío y la conjunción de eventos sobre el mismo predictor al estar definidas el conjunto vacío y la intersección en el conjunto de intervalos de la recta real.

En ambos casos, la función de combinación de niveles de relación (en $\mathcal{B}$) es el producto y se comparten los criterios especificados en 4.2. Todas las demás consideraciones del método aportadas en el capítulo anterior y éste son aplicables. En particular, las deducidas para datos de entrada modales probabilistas.

Otras extensiones toman en cuenta datos de entradas posibilistas o difusos. En estos casos, los elementos de B se refieren a variables posibilistas o difusas. Por tanto, las relaciones difusas y las funciones de niveles de relación toman en cuenta estas semánticas.

En el conjunto de extensiones, teniendo en cuenta las diversas semánticas, los valores $\{\beta_k \wedge \mu_k(\omega), k = 1, ..., K\}$ pueden representar valores binarios (pertenencia o no pertenencia), una distribución de probabilidad sobre los nodos del árbol, unos grados de pertenencia a nodos difusos o una distribución de posibilidad sobre los nodos.

También, extendiendo las aplicaciones de Dubois et al., 1991 y Maher y St. Claire, 1993 (véase 2.2.3), cuando las descripciones de los individuos son conjuntos difusos en lugar de grados de pertenencia a conjuntos difusos, los niveles de relación en las ramas se establecen a partir de unos valores de necesidad o posibilidad aplicados a pares de funciones de pertenencia de conjuntos difusos[12]. En este caso, los valores $\beta_k \wedge \mu_k(\omega)$ son función de estos valores de necesidad (véase (2.40)) o soporte necesario (véase (2.42)) y de la posibilidad (véase (2.41))

[12] En este caso, los elementos de B son del tipo $[Y_j \sim d]$ con $Y_j(\omega)$ y d, dos conjuntos difusos definidos sobre el mismo espacio de referencia.

o, generalizando la relación producto a *intervalos,* el intervalo con cota inferior el soporte necesario y cota superior el soporte posible o posibilidad asociadas a los nodos (véase(2.43)). También se puede aplicar una $T-$conorma como función $g(.)$ de combinación de niveles de relación aplicados a las posibilidades de las ramas, según se introduce en (1.97) o una $T-$norma aplicada a los valores de necesidad de las ramas (véase 1.98) ((Diday, (1991, 1995a)). En el primer caso, deben redefinirse las propiedades (1.51) y (1.52).

En la tabla 4.2, se proponen para predictores posibilistas o difusos, los elementos de B[13], las relaciones de dominio definidas en B, las funciones $g(.)$ de combinación de niveles de relación para elementos de $\mathcal{B}$, la función que se aplica a los niveles de relación de los individuos con el nodo para obtener el valor $P(.)$ correspondiente en el sumatorio de IC y la aplicación $h(.)$ de predicción. Entre paréntesis, antecedentes anteriores de las medidas.

La elección de los elementos de $\mathcal{A}$, como se introdujo en 2.2.3, puede ser diversa. Este es el caso de eventos booleanos, probabilistas o posibilistas. Los eventos booleanos predicen una única clase. Los eventos probabilistas tienen como descripciones distribuciones de probabilidad. La obtención de estas probabilidades en un nodo puede ser frecuentista, de máxima verosimilitud (Périnel, 1996), estimadas (Quinlan, 1990) o probabilidades difusas. Aplicaciones de esta última aproximación, aunque sin considerar la presencia de estratos pueden verse en Séchet, 1995, Verde 1995 y Araya[14], 1995. Los eventos posibilistas, en este caso tienen como descripciones distribuciones de posibilidad, grados de pertenencia a conjuntos difusos o conjuntos difusos. Aplicaciones de distribuciones de posibilidad, aunque sin considerar estratos pueden verse en Rives, 1990, Yuan y Shaw, 1995. Y, de conjuntos difusos en Dubois et al., 1991. En el caso de datos de

[13]Los elementos de B, en los dos primeros casos, tienen una expresión como en (4.35), si bien con distintos tipos de variables. En el tercer caso, la complementariedad del evento difuso se describe por el conjunto difuso complementario (véase definición 1.11).

[14]Séchet parte de predictores difusos, Verde de predictores y clases difusas y Araya de predictores probabilistas y clases difusas.

Datos-Predictor	**Eventos** B	**Relación en** B	$g(.)$ **en** $\mathcal{B}$	$P(.)$ **en** IC	$h(.)$ **predicción**
Grados de pertenencia	$[Y_j \sim z_j]$ ó $[Y_j \sim D_j]$ $z_j \in \mathcal{Y}_j$, $D_j \in \mathcal{P}(\mathcal{Y}_j)$	Grado de pertenencia Idem que (1.80)	Mínimo o $T-$norma (Séchet, Verde) T-conorma[1] ((1.99), Diday) */gr.pert/*	Suma en ω	{gr.pert, predicción}$_{nodos}$ Media gr.pert*predicción Predicción nodo de max gr. pert.
Distribución de posibilidad	$[Y_j \sim z_j]$ ó $[Y_j \sim D_j]$ $z_j \in \mathcal{Y}_j$, $D_j \in \mathcal{P}(\mathcal{Y}_j)$	Posibilidad (1.80)	$T-$conorma[1] ((1.99), Diday) */posibilidad/* ($T-$norma)	Máximo en ω */posibilidad nodo/*	{posibilidad, predicción}$_{nodos}$ Predicción nodo de max. posibilidad
Conjunto difuso	$[Y_j \sim d]$ $Y_j(\omega), d$ conjuntos difusos	Función de N y P (Dubois et al.) $[S_n, S_p]$ (Maher y St.Clair)	Máximo[1] de P ((1.97), Diday) Otras ($\wedge^*[S_n, S_p]$ (2.43)[2], Dubois et al.)	(4.146) no Otras (Dubois et al.)	En función de S_n y S_p ($\vee^*[S'_n, S'_p]$ (2.44)[2]) Otras (Dubois et al.)

[1] Necesario redefinir las propiedades (1.51) y (1.52)

[2] Necesario extender la aplicación de combinación de niveles de relación a intervalos

Tabla 4.2: Extensión del método para datos posibilistas o difusos

entrada multievaluados o de intervalo y variable clase monoevaluada, el conjunto $\mathcal{A}$ sigue la definición (3.3). En la tabla 4.3 se proponen estimaciones de la variable clase Z según los tipos de datos de entrada para predictores y clases. Entre paréntesis aparecen referencias anteriores de las mismas.

Clases Predictores	**Monoevaluados**	**Grados de pertenencia**	**Conjuntos difusos**
Monoevaluados	Probabilidades	Distribución de posibilidad[1] ((2.37) Rives, Yuan y Shaw) Probabilidades difusas	
De intervalo	Probabilidades	Idem que arriba	
Multievaluadas	Probabilidades	Idem que arriba	
Probabilistas	Probabilidades (Araya, Périnel, Bravo & García-Santesmases)	Probabilidades difusas (Araya) Distribución de posibilidad[3]	
Grados de pertenencia	Probabilidades difusas (Verde)	Probabilidades difusas (Séchet)	
Distribución de posibilidad	Distribución de posibilidad[2]	Distribución de posibilidad[4]	
Conjuntos difusos			Conjunto difuso (Dubois et al.)

Tabla 4.3: Extensiones del método. Estimaciones de la variable clase Z, según tipos de predictores y clases.

En esta tabla, las distribuciones de posibilidad propuestas para un nodo k del árbol se expresan de forma común. En un nodo k del árbol, la posibilidad de las clases se define por:

$$Pos(c_i) = \max_{\omega \in \Omega} \beta_k \wedge [Z \sim c_i] \wedge \mu_k(\omega) \quad \text{para} \quad i = 1, ..., s \tag{4.150}$$

La función de combinación de niveles de relación para los elementos de $\mathcal{B}\hat{\wedge}\mathcal{N}$ y $\mathcal{A}$ es el producto, salvo en [3], que es la $T-$norma mínimo. A continuación se detallan estas posibilidades según los tipos de datos de entrada de los predictores y las clases. En [1], para predictores monoevaluados y clases difusas, (4.150) define

la distribución de posibilidad marginal de las clases en el nodo k introducida en (2.37),

$$Pos(c_i) = \max_{\omega/\beta_k \wedge \mu_k(\omega)=1} [Z \sim c_i](\omega), \text{ para } i = 1, ..., s \tag{4.151}$$

con $[Z \sim c_i](\omega) = q_{c_i}(\omega)$ grado de pertenencia de la clase c_i para ω. En [2], para predictores posibilistas y clases monoevaluadas, la distribución (4.150) es:

$$Pos(c_i) = \max_{\omega \in c_i} \beta_k \wedge \mu_k(\omega), \text{ para } i = 1, ..., s \tag{4.152}$$

con $\beta_k \wedge \mu_k(\omega) = q_\omega(k)$ la posibilidad del nodo k para ω (véase tabla 4.2). En [3], para predictores con descripciones grados de pertenencia para categorías difusas y clases difusas, la distribución (4.150) se introduce en (2.41) y es:

$$Pos(c_i) = \max_{\omega \in \Omega} \min\{\beta_k \wedge \mu_k(\omega), [Z \sim c_i](\omega)\} = \max_{\omega \in \Omega} \min\{q_k(\omega), q_{c_i}(\omega)\} \tag{4.153}$$

para $i = 1, ..., s$, con $\beta_k \wedge \mu_k(\omega) = q_k(\omega)$ el grado de pertenencia del nodo k para ω (véase tabla 4.2) y $[Z \sim c_i](\omega) = q_{c_i}(\omega)$ grado de pertenencia de la clase c_i para ω. En [4], para predictores posibilistas y clases difusas, la distribución (4.150) es:

$$Pos(c_i) = \max_{\omega \in \Omega} \beta_k \wedge \mu_k(\omega) q_{c_i}(\omega), \text{ para } i = 1, ..., s \tag{4.154}$$

con $\beta_k \wedge \mu_k(\omega) = q_\omega(k)$ la posibilidad del nodo k para ω (véase tabla 4.2) y $[Z \sim c_i](\omega) = q_{c_i}(\omega)$ grado de pertenencia de la clase c_i para ω.

La función EIC de (3.19) se puede generalizar del mismo modo que la función IC. En las generalizaciones de IC y EIC, cuando se considera $P(.)$ como en (4.147) o (4.149) se puede definir para el estrato $i-$ésimo y el nodo k:

$$P([M = i]|\beta_k \wedge \mu_k) := \frac{P(\beta_k \wedge \mu_k \cap [M = i])}{P(\beta_k \wedge \mu_k)} \tag{4.155}$$

La intersección de eventos de elementos de $\mathcal{N}$ se encuentra definida con anterioridad. Se comprueba que $\sum_{i=1,..,m} P(\beta_k \wedge \mu_k \cap [M=i]) = P(\beta_k \wedge \mu_k)$ por tanto la proporción (4.155) es un valor de 0 a 1 que tiene una interpretación similar a la contribución absoluta de un estrato en un nodo (4.100).

En las generalizaciones de IC y EIC, cuando se considera $P(.)$ como en (4.148) se defin para el nodo k:

$$\begin{aligned} P([M &= i]|\beta_k \wedge \mu_k) := P(\beta_k \wedge \mu_k \cap [M=i]) = \max_{\omega \in \Omega} \beta_k \wedge \mu_k \cap [M=i](\omega) \\ &= \max_{\omega \in S_i} \beta_k \wedge \mu_k(\omega) \qquad \text{para } \ i = 1, ..., m \end{aligned} \tag{4.156}$$

que representan las posibilidades de los estratos condicionadas por el nodo k.

Las contribuciones absolutas de un estrato a un nodo, introducidas en 4.4.3, se reemplazan ahora por el cociente entre los pesos de los estratos en el nodo y el peso del nodo como en (4.155) o por los valores de la distribución de posibilidad definida sobre los estratos en un nodo como en (véase(4.156)).

La función de admisibilidad definida como en 4.2.2 transforma la condición 1 de 4.1.2, en que la admisibilidad para un corte $b \in B$ es nula desde un nodo explorable $r \in X$ si alguno de los eventos de $\beta_r \wedge b$ o $\beta_r \wedge b^c$ son el evento vacío. Los criterios de condición de nodo decisional y condición de parada de los estratos (véase 3.2.4) pueden formalizarse como en 4.2, salvo cuando se considera $P(.)$ como en (4.148) en IC y EIC que se modifican:

- La función de admisibilidad cambia además la condición 2. de 4.2.2 que es transformada en que la admisibilidad para un corte $b \in B$ es nula desde un nodo explorable $r \in X$ si:

$$\min\{\max_{\omega \in \Omega} \beta_r \wedge b \wedge \mu_r(\omega), \max_{\omega \in \Omega} \beta_r \wedge b^c \wedge \mu_r(\omega)\} < \nu \tag{4.157}$$

  con $\nu \in [0,1]$. Es decir, si la posibilidad de uno de los nodos hijos es pequeña.

- La condición de parada desde un nodo k para un estrato $S_i \in E$ se basa en la posibilidad del estrato en el nodo y se define de la siguiente forma:

$$Stop_\tau(\beta_k \wedge \mu_k, S_i) = 1 \iff \max_{\omega \in \Omega} \beta_k \wedge [M = i](\omega) < \tau, i \in S^k \quad (4.158)$$

con $\tau \in [0, 1]$, siendo $\mu_k = [M \in S^k]$.

El método presentado en 3.3 no cumple necesariamente (3.32a) y (3.32b), por la utilización de $T-$conormas como funciones de combinación de niveles de relación. Este hecho que no afecta el desarrollo del método, ya que se parte de una descripción en $\mathcal{B}$ del primer corte, expresadas en (3.31a) y (3.31b).

En el caso posibilista o difuso, la aplicación $h(.)$ de predicción introducida en (3.9) en términos generales se aplica a un conjunto de grados de pertenencia (definiendo una partición en el sentido de Ruspini o no) o a una distribución de posibilidad, relativas al conjunto de nodos, con las estimaciones de la variable clase en ellos. Dado un individuo, se obtiene como predicción una secuencia de grados de pertenencia o valores de posibilidad y estimaciones de la variable Z, relativos a los nodos del árbol. Se puede dar esta secuencia como predicción u optar por una operación de los grados de pertenencia (media, por ejemplo) combinados con las estimaciones o, tomar como predicción la estimación de la variable clase correspondiente al nodo con el máximo u otra $T-$conorma del grado de pertenencia o valor de posibilidad. En los casos que dado un individuo, su predicción es un evento probabilista o posibilista, también se le puede asociar la clase de mayor probabilidad o de mayor posibilidad; Y, en el caso posibilista, la clase de mayor grado de distinguibilidad (véase Rives, 1990 y 2.2.3). En la tabla 4.2 aparecen sintetizadas estas propuestas.

La descripción de los estratos (véase 3.2.3) se define también como un conjunto de objetos simbólicos o reglas de predicción ponderados. Pueden definirse las contribuciones relativas de los nodos a un estrato como en (4.99) que sobre todo tienen sentido si la suma de niveles de relación de un individuo con los nodos del

árbol es 1 como en el caso de las particiones difusas en el sentido de Ruspini.

Cuando las particiones son difusas pero no en el sentido de Ruspini, la suma de los w_k^i de (4.99) para $k = 1, ..., K$ no es 1. Estos valores de pueden interpretar también como grados de pertenencia del estrato a los nodos del árbol.

La descripción de los estratos puede ser una distribución de posibilidad sobre los nodos del árbol. Cuando $P(.)$ se define como en (4.148), entonces se definen las posibilidades de los nodos para el estrato $i-$ésimo como:

$$pos_k^i = \max_{\omega \in \Omega} \beta_k \wedge \mu_k \cap [M = i](\omega), \ k = 1, ..., K \tag{4.159}$$

donde el máximo puede ser sustituida por otra $T-$conorma como en (4.148) y (4.156). La interpretación de estas posibilidades es diferente al de las contribuciones relativas de los nodos al estrato. En este caso, el estrato se interpreta como una distribución de posibilidad sobre los nodos del árbol. Tampoco se cumple como en (3.11) que $\sum_{k=1,...,K} pos_k^i = 1$.

Las propuestas generales aportadas en esta sección se pueden extender a otras generalizaciones que incluyan sus propias relaciones de dominio entre las descripciones de los nodos y las descripciones de los datos. También se puede generalizar a datos simbólicos de creencia en los predictores e, incluso, la matriz de entrada de datos puede mezclar distintos tipos de datos simbólicos para los predictores. Basta definir los elementos de los conjuntos $\mathcal{A}$, B y $\mathcal{B}$ con sus relaciones de dominio correspondientes y las funciones de combinación de niveles de relación.

Por último destacar que si bien no todas las extensiones posibles del método deben tener una medida de contenido de información expresada como en (4.146), es de destacar que el marco formal de representación del árbol mediante objetos simbólicos aporta una formalización que encuadra muy diversos tipos de variables y datos y de ahí la importancia de esta formalización.

4.9 Aplicaciones

4.9.1 Datos SES, 1995

El algoritmo de Segmentación para datos estratificados ha sido aplicado al conjunto de datos *"T25IT Italy: Monthly earnings por local unit size, NACE (economic activity) and ISCO (profession)"*[15]. Este conjunto de datos contiene datos estadísticos acerca de la estructura y distribución de salarios del año 1995 en Italia. Se ha obtenido en la dirección de Internet *http://europa.eu.int/eu/comm/eurostat/research/conferences/ntts-98.* La utilización de estos datos fue sugerida por los organizadores del congreso NTTS-98 (New Techniques and Technologies for Statistics, 1998) y resultados de esta aplicación pueden consultarse en Bravo y García-Santesmases, 2000b. Según se detalla en las especificaciones el propósito de estos datos estadísticos es determinar los salarios recibidos por los empleados de ciertos trabajos.

Descripción de los datos

Los datos representados en este conjunto representan un conjunto de 5.511.473 empleados divididos en 2772 segmentos. Estos segmentos se describen por la combinación de las variables: *sexo*, tamaño de la empresa en número de empleados ($lu_service$) (7 categories), profesión ($ISCO$) (una taxonomía de 9 categorías) y actividad económica ($NACE$) (una taxonomía de 22 categorías). $ISCO$ y $NACE$ son clasificaciones oficiales de la Unión Europea de las variables profesión y sector económico, respectivamente.

Las variables recogidas en cada segmento son: el número de empleados con las características definidas por el segmento (n_univer), número medio de horas trabajadas a la semana (m_hour), media del salario bruto mensual (m_m_earn), coeficiente de variación del salario bruto mensual (cvm_earn) y media mensual

[15] SES: Statistics on the Structure and Distribution of Earnings.

del valor de los bonos periódicos (m_m_bon). Se calcula además la variable media del salario bruto por hora (m_h_earn) de la siguiente forma:

$$m_h_earn = \frac{m_m_earn}{(4 + \frac{2}{5}) \times m_hour}$$

A partir de las distribuciones de probabilidad empíricas de las variables m_hour, m_m_earn, m_m_bon y m_h_earn, que tienen en cuenta el número de empleados en cada segmento, se obtuvieron los percentiles 25, 50 y 75. Para cada una de las variables m_hour, m_m_earn, m_m_bon y m_h_earn se obtienen tres variables binarias[16] cuyo valor es 1 si el valor de la variable original en el segmento es menor o igual que el percentil respectivo y vale 0 en caso contrario. A partir de m_hour, se construyen las variables $h25$, $h50$ y $h75$; a partir de m_m_earn, las variables $sal25$, $sal50$ y $sal75$; a partir de m_m_bon, las variables $b25$, $b50$ y $b75$ y a partir de m_h_earn, las variables $salh25$, $salh50$ y $salh75$. A partir de la variable cvm_earn se construye la variable binaria cvm considerada la media de la variable original como umbral de corte.

Para aplicar el algoritmo de Segmentación para datos estratificados según los pesos de los segmentos, se construye una matriz de 13.083 unidades de datos, de tal forma que se replica un segmento por su peso relativo en el conjunto. El peso unidad considerado es de 400. Los segmentos con valor de $n_univer < 400$ están presentes una vez en la matriz de datos. Cuatro segmentos que tienen un valor de $n_univer > 70000$ se replican sólamente 175 veces. Futuras versiones del software desarrollado (véase capítulo 5) consideraran el peso de las observaciones, sin necesidad de realizar replicado de las mismas.

[16] Al estar implementado el método para predictores binarios, la creación de estas variables binarias basadas en los percentiles 25, 50 y 75 permite simular cortes del árbol en estos percentiles de los predictores originariamente continuos: valores medios de salarios y horas trabajadas.

La matriz de datos de entrada

El conjunto Ω contiene 13083 ($n = 13083$) unidades de datos descritos en la epígrafe anterior, el conjunto $E \subset \mathcal{P}(\Omega)$ contiene subconjuntos de unidades de datos que pertenecen al mismo sector $NACE$. El número de estratos es 5 al reducir la taxonomía original a 5 categorías ($m = 5$). La variable clase Z es la variable binaria $administrativo$[17] ($s = 2$), y los predictores son variables relacionadas con el sexo, salarios y horas trabajadas por los empleados: $sexo$, $h25$, $h50$, $h75$, $sal25$, $sal50$, $sal75$, $b25$, $b50$, $b75$, $salh25$, $salh50$, $salh75$, cvm y los cortes binarios posibles de la variable tamaño de la empresa, $lu_service$. Los sectores $NACE$ considerados son: $minería$; $manufactura$; $electricidad$, gas y agua; $construcción$; y, $servicios$. Un ejemplo de una unidad de datos $\omega \in \Omega$ se describe por:

$$\begin{aligned}\omega : (&sexo(\omega) = f, administrativo(\omega) = sí, sal25(\omega) = no, sal50(\omega) = sí,\\ &sal75(\omega) = sí, \dots, NACE(\omega) = servicios, lu_service \geq 1000)\end{aligned} \tag{4.160}$$

que representa un segmento de 400 empleados (salvo excepciones) mujeres cuya profesión es $administrativo$, y cuyo salario bruto medio mensual se encuentra entre el percentil 25 y la mediana, que trabajan en el sector servicios y en una empresa que tiene más de 1000 empleados. Los objetivos que se persiguen en este ejemplo son:

- Explicar los empleados $administrativo$ por los predictores, afectados por el sector $NACE$ en el que trabajan;
- Encontrar conjuntos de sectores $NACE$ para los que esta explicación es la misma, es decir, para los que la caracterización de los empleados $administrativo$ por los predictores es la misma.

[17] Las cuatro grandes categorías de profesión son: profesionales liberales, técnicos, administrivos y empleados manuales.

- Describir o caracterizar un sector $NACE$ en función de la explicación obtenida para los empleados *administrativo*.

El ejemplo se puede ver como una descripción de la explicación de los empleados administrativos, según el sector para el que trabajan. Y, también se pueden ver los resultados como una herramienta de imputación de datos.

Resultados

Las figuras 4.3 y 4.4 muestran los árboles de decisión construidos para 5 y 3 niveles, respectivamente. Los nodos decisionales son los cuadrados, que difieren de color según su procedencia y de intensidad de color según la clase que explican. Los nodos de color verde proceden de comprobar la condición de nodo decisional (véase 3.3.4) y los de color anaranjado de un proceso terminal-divide (véase 4.6.2). Los nodos de color claro representan reglas de predicción para empleados *administrativo* y los nodos de color oscuro representan reglas de predicción para empleados $no - administrativo$. Cuanto mayor intensidad tenga la claridad o la oscuridad mayor es la probabilidad de la clase que explican. La posición relativa de estos nodos respecto al nodo explorable del que proceden explican también su procedencia y la clase que explican: arriba se situan los procedentes de la condición de nodo decisional, de color verde y abajo los procedentes de un proceso terminal-divide, de color naranja. Los nodos que se sitúan a la izquierda del nodo explorable del que proceden son los de intensidad clara que explican la clase *administrativo* y los que se sitúan a la derecha son los de intensidad oscura que explican la clase $no - administrativo$. Los nodos explorables en algún paso del algoritmo, se presentan de color azul con diversas intensidades también.

En el árbol presentado en la figura 4.3, el valor inicial de la medida de contenido de información u opuesto de la entropía es -0.630191 y el valor final es -0.412277. En la figura 4.4, los nodos muestran los pesos y la probabilidad estimada para los empleados $no - administrativo$. Los nodos decisionales mues-

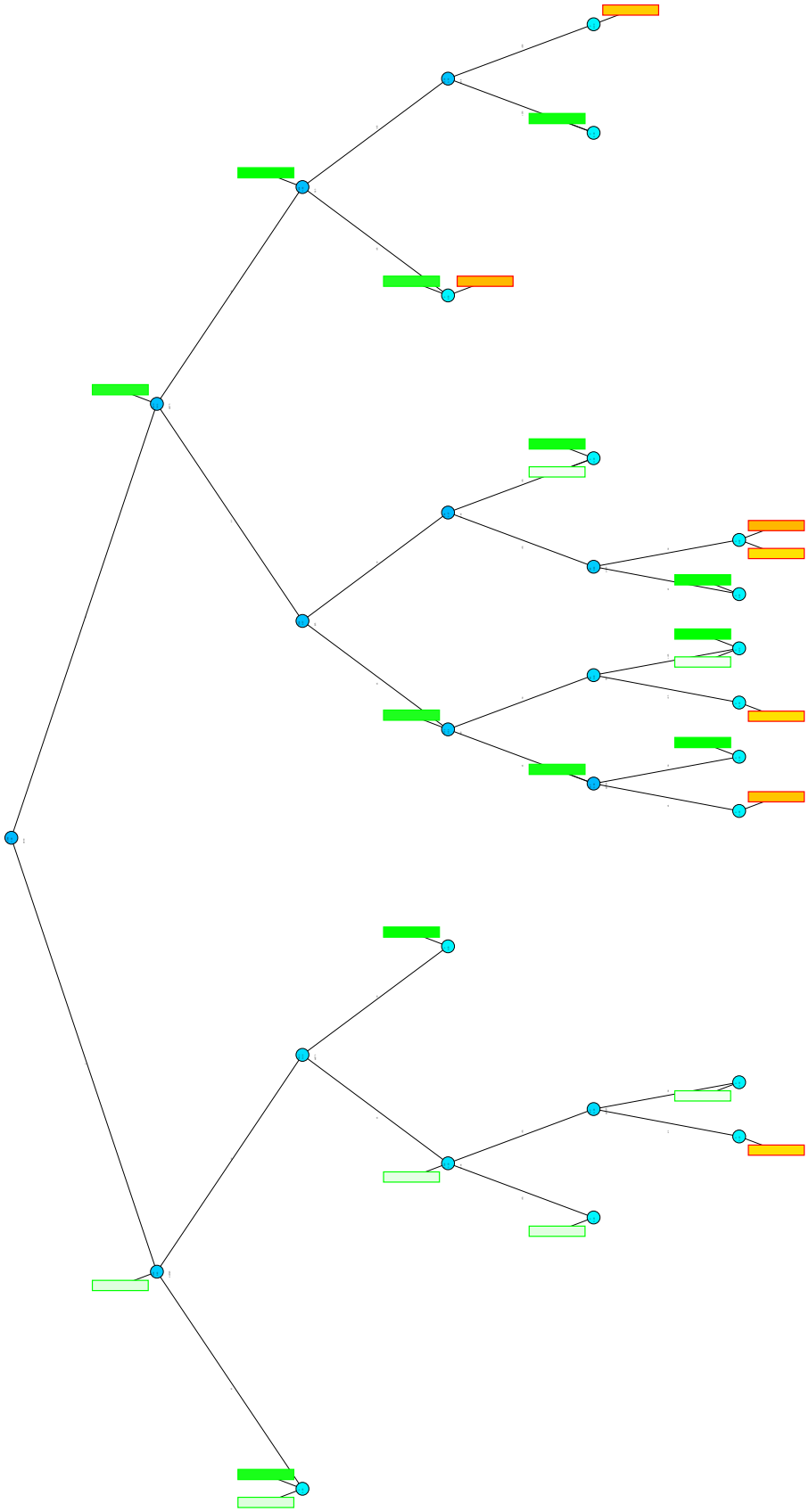


Figura 4.3: Árbol de Segmentación para los datos SES, 5 niveles

tran también los sectores $NACE$. En este árbol, el valor final de la medida de contenido de información es -0.455099.

El conjunto de nodos decisionales constituyen la descripción del árbol. Este conjunto (véase (3.5)) es:

$$T = \{t_k\}_{k=1,\dots,K} = \{\beta_k \wedge \alpha_k \wedge \mu_k\}_{k=1,\dots,K} \tag{4.161}$$

con K el número de nodos decisionales cuando el árbol construido tiene un máximo de 3 niveles. Un nodo del árbol se describe según (3.6) como:

$$\begin{aligned} t_1(10d0) = \beta_1 \wedge \alpha_1 \wedge \mu_1 = [sexo = f] \wedge [administrativo \sim (no(0.09), sí(0.91))] \\ \wedge [NACE = construcción] \end{aligned} \tag{4.162}$$

con $\beta_1 = [sexo = f]$ la aserción (en este caso evento) definida sobre los predictores, $\mu_1 = [NACE = construcción]$ el evento definido sobre la variable estrato y $\alpha_1 = [administrativo \sim (no(0.09), sí(0.91))]$, el evento probabilista definido sobre la variable clase. La aserción $\beta_1 \wedge \mu_1$ representa un conjunto de individuos para los que la predicción de profesión es *administrativo*, con probabilidad 0.91, es decir, para los estratos de la descripción de μ_1, en este caso el estrato *construcción*: si un empleado es *mujer*, entonces es *administrativa* con probabilidad 0.91.

Identificación de los nodos del árbol

En estos ejemplos, para facilitar la ubicación de los nodos en el árbol y la interpretación de los mismos se identifican los nodos en (4.161) como una secuencia:

$$ij\delta c \tag{4.163}$$

con i el nivel del nodo explorable del que procede, $j \in \{0, ..., 2^i - 1\}$ el índice que indica la posición teórica (tomada de izquierda a derecha en un nivel del árbol) del nodo explorable del que procede; $\delta \in \{d, td, t\}$ identifica si el nodo procede de comprobar la condición de nodo decisional (véase 3.3.4), de un proceso terminal-divide (véase 4.6.2) o de comprobar la condición de parada de los estratos (véase 3.3.5), respectivamente; $c \in \{0, 1\}$ identifica la clase de mayor probabilidad de predicción. En este ejemplo, la clase $c = 0$ se identifica con la profesión *administrativo*. La posición relativa de un nodo del árbol viene referida a la posición relativa del nodo explorable del que procede y se expresa por los índices i, j. El número de nodos explorables teóricos[18] en un nivel i del árbol es 2^i. Dos nodos identificados por un mismo par i, j proceden del mismo nodo explorable y tienen en común todos los cortes relativos a los predictores. Dos nodos identificados por un mísmo índice i e índice $j \in \{2^k, 2^k + 1\}$ con $k \in \{0, ..., 2^i - 1\}$ indica dos nodos que tienen en común todos los antecedentes relativos a los predictores salvo el último que hace referencia al mismo predictor en los dos nodos pero a dos conjuntos disjuntos de categorías del mismo.

En la representación del árbol, los nodos identificados con $c = 0$, se situan a la izquierda del nodo explorable del que proceden y tienen color claro. Los nodos identificados con $c = 1$ se situan a la derecha y tienen color oscuro.

Así, la identificación $10d0$ en (4.162) indica su procedencia de un nodo explorable del primer nivel ($i = 1$) (sólo un corte definido en los predictores), en la representación visual es el que está más a la izquierda ($j = 0$), procede de la condición de nodo decisional ($\delta = d$) y predice la clase *administativo* ($c = 0$).

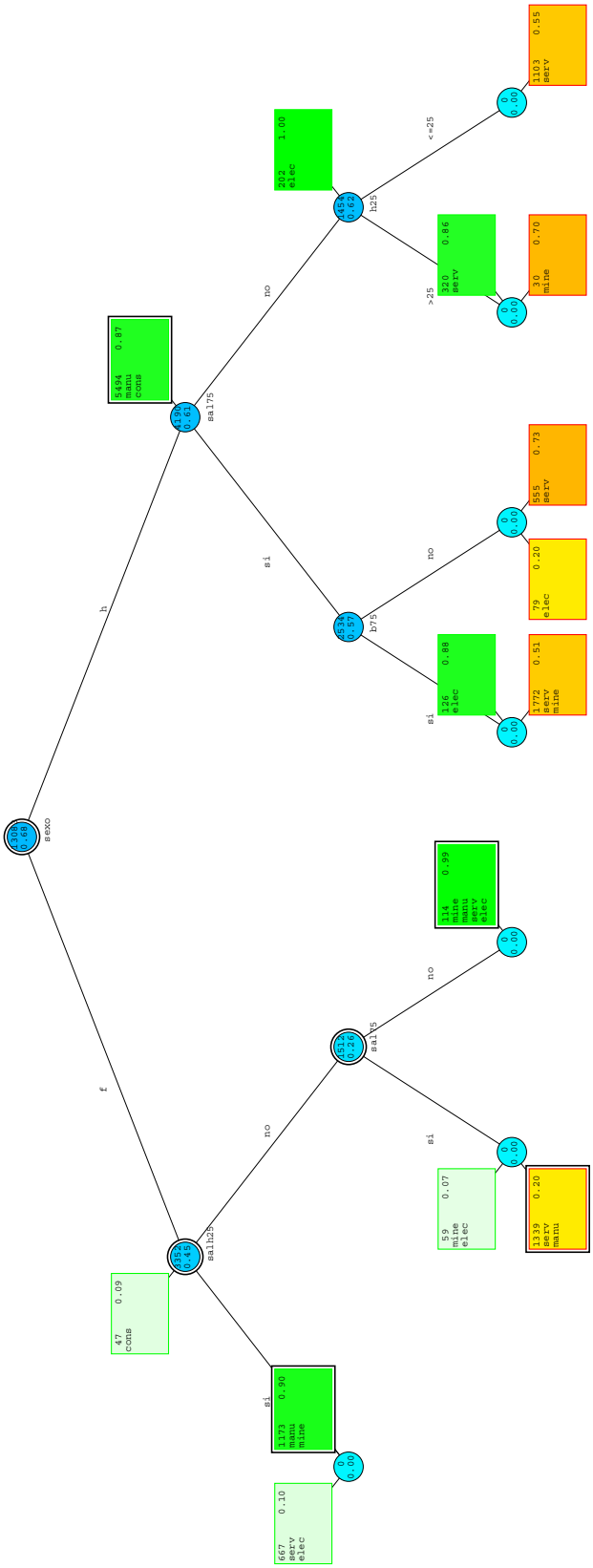


Figura 4.4: Árbol de Segmentación para los datos SES, 3 niveles

Interpretación de los resultados

Además del nodo $t_1(10d0)$, en (4.162), los demás nodos se decisionales describen según (3.6) como:

$t_2(11d1) = \beta_2 \wedge \alpha_2 \wedge \mu_2 = [sexo = h] \wedge [administrativo \sim (no(0.87), sí(0.13))]$
$\wedge [NACE \in \{manufactura, construc\}]$

$t_3(20d0) = [sexo = f] \wedge [salh25 = sí] \wedge [administrativo \sim (no(0.1), sí(0.9))]$
$\wedge [NACE \in \{servicios, electricidad\}]$

$t_4(21d1) = [sexo = f] \wedge [salh25 = sí] \wedge [administrativo \sim (no(0.9), sí(0.1))]$
$\wedge [NACE \in \{manufactura, minería\}]$

$t_5(23d1) = [sexo = h] \wedge [sal75 = no] \wedge [administrativo = no] \wedge [NACE = electric]$

$t_6(32d0) = [sexo = f] \wedge [salh25 = no] \wedge [sal75 = sí] \wedge [administrativo \sim (no(0.07),$
$sí(0.93))] \wedge [NACE \in \{minería, electricidad\}]$

$t_7(32td0) = [sexo = f] \wedge [salh25 = no] \wedge [sal75 = sí] \wedge [administrativo \sim$
$(no(0.2), sí(0.8))] \wedge [NACE \in \{manufactura, servicios\}]$

$t_8(33d1) = [sexo = f] \wedge [salh25 = no] \wedge [sal75 = no] \wedge [administrativo \sim (no(0.99)$
$, sí(0.01))] \wedge [NACE \in \{minería, manufactura, servicios, electricidad\}]$

$t_9(34d1) = [sexo = h] \wedge [sal75 = sí] \wedge [b75 = sí] \wedge [administrativo \sim (no(0.88), sí(0.12))]$
$\wedge [NACE = electricidad]$

$t_{10}(36d1) = [sexo = h] \wedge [sal75 = no] \wedge [h25 = no] \wedge [administrativo \sim (no(0.86)$
$, sí(0.14))] \wedge [NACE = servicios]$

$t_{11}(35td0) = [sexo = h] \wedge [sal75 = sí] \wedge [b75 = no] \wedge [administrativo \sim (no(0.2), sí(0.8))]$
$\wedge [NACE = electricidad]$

$t_{12}(35td1) = [sexo = h] \wedge [sal75 = sí] \wedge [b75 = no] \wedge [administrativo \sim (no(0.73),$
$sí(0.27))] \wedge [NACE = servicios]$

$t_{13}(36td1) = [sexo = h] \wedge [sal75 = no] \wedge [h25 = no] \wedge [administrativo \sim (no(0.7),$
$sí(0.3))] \wedge [NACE = minería]$

[18] Es posible que no todos teóricos de un nivel existan, por cumplirse las condiciones de parada de un antecedente suyo en un nivel anterior (no más cortes admisibles, incremento relativo de medida de contenido de información pequeño o tamaño 0)

La descripción de un estrato S_i, $i = 1, ..., 5$ según (3.11) se expresa como (véase 4.4):

$$S_i : \{w_k^i \ (\beta_k \wedge \alpha_k)\}_{k=1,...,K} \tag{4.164}$$

con $w_k^i \in [0, 1]$ el peso del nodo decisional k en el estrato S_i o contribución relativa del nodo k al estrato S_i (véase (4.99)).

La figura 4.4 señala los nodos decisionales para el sector $manufactura$, con un marco doble. La descripción obtenida para este sector es:

$$\begin{aligned} manufactura : \{&0.74[sexo = h] \wedge [administrativo \sim (no(0.87), sí(013))], \\ &0.17[sexo = f] \wedge [salh25 = sí] \wedge [administrativo \sim (no(0.9), sí(0.1))], \\ &0.08[sexo = f] \wedge [salh25 = no] \wedge [sal75 = sí] \\ &\wedge [administrativo \sim (no(0.2), sí(0.8))]\} \end{aligned} \tag{4.165}$$

El sector $manufactura$ (de peso inicial 6675) se describe por las reglas: *si un empleado es hombre, entonces su profesión es* $no - administrativo$ *(probabilidad* 0.87*); si un empleado es mujer con salario bruto medio hora menor que el primer cuartil, entonces su profesión es* $no - administrativo$ *(prob.* 0.9*); y finalmente, si un empleado es mujer con salario bruto medio hora menor que el tercer cuartil último, entonces su profesión es administrativo (prob.* 0.8*).* Los pesos respectivos de estas reglas son 0.74, 0.17 y 0.08. Hay otro nodo con peso irrelevante (0.01) en la descripción de este sector, que no se detalla (véase tabla 4.4).

La tabla 4.4 muestra las contribuciones relativas de los nodos del árbol a los estratos, bajo el epígrafe Cr y las contribuciones absolutas de los estratos a los nodos, bajo el epígrafe Ca. Ambas contribuciones están definidas en (4.99) y (4.100), respectivamente.

Las *contribuciones relativas* permiten describir los estratos por las reglas de predicción. Fijado un estrato, la suma de los elementos de la *columna* correspon-

	minería		*manufact.*		*electric.*		*construc.*		*servicios*	
	Cr	Ca	Cr	Ca	Cr	Ca	Cr	Ca	Cr	Ca
t_1_10d0							0.08	1		
t_2_11d1			0.74	0.9			0.92	0.1		
t_3_20d0					×	0.01			0.13	0.99
t_4_20d1	0.16	0.04	0.17	0.96						
t_5_23d1					0.44	1				
t_6_32d0	0.08	0.32			0.08	0.68				
t_7_32td0			0.08	0.42					0.15	0.58
t_8_33d1	0.05	0.1	×	0.22	×	0.07			×	0.6
t_9_34d1					0.27	1				
t_{10}_36d1									0.06	1
t_{11}_35td0					0.17	1				
t_{12}_35td1									0.1	1
t_{13}_36td1	0.13	1								
Otros	0.55								0.52	

El símbolo × especifica un valor muy pequeño

Tabla 4.4: Tabla de contribuciones relativas y absolutas. Datos SES.

diente debe ser 1. Los valores altos se corresponden con los nodos/reglas que caracterizan un estrato.

Las *contribuciones absolutas* permiten caracterizar las reglas de predicción en el sentido de determinar la importancia de los estratos en las mismas. Fijado un nodo, la suma de los elementos de la *fila* correspondiente bajo los epígrafes Ca debe ser 1. Los valores altos se corresponden con los estratos que caracterizan un nodo. La fila *Otros* refiere nodos de nivel superior a tres. Pare éstos globalmente no puede establecerse un valor de contribución absoluta.

Así, observando las contribuciones relativas, se concluye que el sector *minería* se caracteriza por los nodos t_4, t_6, t_8, t_{13} y otros de nivel superior a 3, con pesos respectivos 0.16, 0.08, 0.05, 0.13 y 0.55, el sector $manufactura$ se catacteriza por los nodos t_2, t_4 y t_7, de pesos respectivos 0.74, 0.17 y 0.09 como ya se ha expresado con anterioridad en (4.165), el sector *electricidad*, gas y agua por los nodos t_5, t_6, t_9 y t_{11}, con pesos respectivos 0.44, 0.08, 0.27 y 0.17, el sector *construcción* por los nodos t_1 y t_2, de pesos respectivos 0.08 y 0.92, y el sector *servicios* por

los nodos t_3, t_7, t_{10}, t_{12} y otros de nivel superior a 3, con pesos respectivos 0.13, 0.15, 0.06, 0.1 y 0.52.

Se muestran a continuación algunos casos concretos de interpretación de los resultados de este ejemplo.

Si se observan los sectores $manufactura$ y *servicios* (de peso inicial 5141) se puede apreciar, por las contribuciones relativas no nulas en el mismo nodo simultáneamente, que las reglas que comparten son las dadas por los nodos $t_7(32td0)$ y $t_8(33d1)$. La regla t_7 predice la clase *administrativo* y tiene una importancia relativa del 9 y 15 por ciento en estos dos estratos respectivamente. Además, esta regla es exclusiva de estos dos estratos ya que la suma de ambas contribuciones absolutas es 1 (0.42+0.58). La regla t_8 predice la clase $no-administrativo$ y si bien la importancia relativa de la misma en ambos sectores es pequeña ($Cr = \times$), ambos sectores caracterizan casi exclusivamente esta regla, ya que la suma de sus contribuciones absolutas es 0.82. Esto sucede por ser el peso del nodo pequeño.

La regla especificada por el nodo t_7 es: *si un empleado es mujer con salario bruto medio hora mayor que el primer cuartil y salario bruto medio mensual menor que el último cuartil, entonces su profesión es administrativo (prob.* 0.8*).*

Por otra parte, el sector $manufactura$ caracteriza las reglas $t_2(11d1)$ ($Ca = 0.9$) y $t_4(20d1)$ ($Ca = 0.96$), teniendo ambas una importancia relativa en el sector del 74 y 17 por ciento ($Cr = 0.74$ y $Cr = 0.17$), respectivamente.

El sector *servicios* caracteriza la regla $t_3(20d0)$ ($Ca = 0.99$) que tiene una importancia relativa del 13 por ciento ($Cr = 0.13$).

Gracias a la forma de identificación de los nodos se observa que los nodos $t_3(20d0)$ y $t_4(20d1)$ difieren sólo en la clase que predicen y en los estratos para los que están definidos (el antecedente 20 indica una posición relativa en el árbol caracterizada por determinados cortes en los predictores).

Así, los sectores $manufactura$ y *servicios* presentan reglas antagónicas expresadas por t_4 y t_3: *si un empleado es mujer y el salario bruto medio por*

hora es menor que el primer cuartil, entonces en el sector manufactura, es no − administrativo (prob. 0.9*) y en el sector servicios, es administrativo (prob.* 0.9*).* Estas dos reglas tienen un peso relativo en los sectores *manufactura* y *servicios* de 0.17 y 0.13, respectivamente. Además, la primera regla de ellas también se da para el estrato *minería* con una importancia relativa en el mismo del 16 por ciento ($Cr = 0.16$), si bien este estrato caracteriza menos la regla ($Ca = 0.04$). Y, la segunda se da para el estrato *electricidad* aunque con importancia relativa mínima ($Cr = \times, Ca = 0.01$).

El sector *construcción* se caracteriza únicamente por dos reglas de predicción, las especificadas por $t_1(10d0)$ y $t_2(11d1)$ de pesos relativos 0.08 y 0.92 respectivamente. Los antecedentes 10 y 11 indican que estos nodos del nivel 1 difieren sólo en un antecedente predictor (como no podría ser de otro modo en un primer nivel). Ambas reglas se derivan del sexo del empleado. Este es un sector con un 8 por ciento de mujeres, la mayoría de ellas *administrativo* y un 92 por ciento varones, la mayoría de ellos $no - administrativo$. La regla t_1 es exclusiva ($Ca = 1$) de este sector: *si el empleado es mujer, entonces es administrativo* (*prob.* 0.91). La regla t_2 es compartida con el sector $manufactura$ ($Ca = 0.9$), para el cual tiene una importancia relativa del 74 por ciento ($Cr = 0.74$).

Los sectores *electricidad* y *servicios* presentan también dos reglas antagónicas, las expresadas por $t_{11}(35td0)$ y $t_{12}(35td1)$ respectivamente y que tienen una importancia relativa en ambos sectores de 0.17 y 0.1, respectivamente.

Ayudas a la interpretación al árbol son las proporcionadas por el gráfico del árbol en las figuras 4.3 y 4.4, por la identificación de los nodos según (4.163) y por la tabla 4.4 de contribuciones relativas y absolutas. Para determinar las reglas de predicción concretas se deben consultar los objetos simbólicos que las describen (véase 4.162). Con mayor facilidad se describen los estratos si se presentan con las reglas de predicción detalladas (véase 4.165). En el capítulo 5, se presenta un editor gráfico del árbol que permite, entre otras utilidades, identificar las clases de predicción según los colores de los nodos, y navegar por los nodos del árbol,

acceder a información detallada de los nodos e identificar los nodos que describen los estratos de forma interactiva. Todas estas utilidades facilitan asimismo la interpretación de los resultados.

4.9.2 Normas de interpretación

Como se ha visto en el ejemplo anterior, se aplican las siguientes ayudas a la interpretación de los resultados:

- La visualización del árbol permite ver los nodos del mismo, profundidad, identificación, descripción, etc... (véase figuras 4.3 y 4.4 y capítulo 5). La intensidad de los colores de los nodos y su posición relativa con respecto a los nodos explorables del que proceden, permiten identificar las clases de predicción.

- La identificación de los nodos permite conocer su posición relativa en la visualización del árbol, su procedencia (condición de nodo decisional, terminal o por proceso terminal-divide) y la clase que predice (véase 4.163).

- Los nodos se describen por objetos simbólicos que son reglas de predicción para algunos estratos (véase 4.162).

- Las *contribuciones absolutas* permiten caracterizar las reglas de predicción en el sentido de determinar la importancia de los estratos en las mismas. Se inspeccionan los elementos bajo los epígrafes Ca de las filas de la tabla de contribuciones (véase tabla 4.4) para identificar cuáles son los estratos que caracterizan las reglas.

- Los estratos se describen por varias reglas de predicción (véase 4.164 y 4.165).

- Las *contribuciones relativas* permiten caracterizar los estratos por las reglas de predicción. Se inspeccionan las columnas con el epígrafe Cr de la tabla

de contribuciones (véase tabla 4.4) para conocer la importancia relativa en el estrato de las reglas correspondientes.

- Inspeccionando la tabla de contribuciones absolutas y relativas se pueden detectar estratos que comparten reglas, es decir, aquellos con contribuciones relativas no nulas en el mismo nodo simultáneamente. También, la importancia de los estratos en estas reglas mediante las *contribuciones absolutas*.

- Para detectar conjuntos de estratos con reglas antagónicas, es decir, con antecendentes iguales en los predictores que predicen distinta clase, se localizan los nodos con igual posición relativa en el árbol, es decir, con idénticos índices antecedentes i, j de su identificación $ij\delta c$ (véase 4.163) y con clase c distinta. En la visualización de árbol son nodos que proceden del mismo nodo explorable y se sitúan a derecha e izquierda del mismo.

- Para detectar conjuntos de estratos que predicen la misma clase teniendo los valores del último predictor antagónicos, se localizan los nodos de posiciones relativas con igual índice i y de índices $j = 2^k$ y $j = 2^k + 1$, con $k \in \{0, ..., 2^i - 1\}$ de su identificación $ij\delta c$ (véase 4.163) y con igual clase c. En la visualización de árbol son nodos que proceden de dos nodos explorables, éstos hijos de un mismo padre, y la posición relativa de ambos con respecto a su nodo explorable es a la derecha o a la izquierda de los mismos.

4.9.3 Apreciación de los municipios

El ejemplo introducido en el ejemplo 3.3 hace referencia a una encuesta tomada a un conjunto Ω de $n = 4606$ personas de un conjunto E de $m = 19$ municipios o estratos con el objetivo de analizar la apreciación global de la población acerca de su municipio a partir de aspectos parciales de apreciación. El conjunto Ω esta descrito por $p = 8$ predictores monoevaluados. Sean Y_j la variable que mide un aspecto de apreciación parcial, Z mide la apreciación global de un individuos a

su propio municipio y la variable M identifica la pertenencia a un municipio. Las apreciaciones parciales de los municipios se refieren a las puntuaciones factoriales resultantes de un Análisis Factorial de Correspondencias. Los aspectos parciales de apreciación considerados se corresponden con los 8 primeros ejes factoriales. Estos son: Y_1 : integración en el municipio, Y_2 : habitabilidad, Y_3 : espacio en el municipio, Y_4 : optimismo social, Y_5 : conformismo social, Y_6 : integración inmediata, Y_7 : percepción del trabajo y Y_8 : percepción de la salud. Resultados de esta aplicación pueden consultarse en Bravo y García-Santesmases (1997, 1998).

En este ejemplo existe de partida bastante diferencia entre los municipios en la apreciación global de los individuos hacia su municipio. Es por esta razón que en algunos municipios las apreciaciones globales positivas o negativas son muy mayoritarias. A un nivel inicial, por ejemplo, se excluyen del proceso recursivo los municipios S_3, S_{16} y S_{18}, por tener una apreciación global extremadamente negativa ($p = 0.08$) y cumplir por si mismos la condición de nodo decisional. El valor de p indica la probabilidad estimada de apreciación global positiva.

Un individuo $\omega \in \Omega$ puede estar representado por:

$$\omega : (Y_1(\omega) = (-), Y_2(\omega) = (+), Y_3(\omega) = (+), Y_4(\omega) = (+), Y_5(\omega) = (-),$$
$$Y_6(\omega) = (+), Y_7(\omega) = (-), Y_8(\omega) = (-), Z(\omega) = (+), M(\omega) = 13)$$

que representa un individuo del municipio S_{13} que tiene una apreciación global positiva y aspectos de apreciación parcial positiva para Y_2, Y_3, Y_4, Y_6, es decir, la habitabilidad, el espacio del municipio, el optimismo social y la integración inmediata. Y, negativa para el resto, es decir, la integración en el municipio, el conformismo social y las percepciones del trabajo y la salud.

Las figuras 4.5 y 4.6 representan el árbol de decisional obtenido a 5 y 3 niveles respectivamente. La identificación de colores y formas son idénticas a la aplicación de 4.9.1. Los nodos claros, situados a la izquierda del nodo explorable del que proceden, se identifican con una apreciación global negativa del munici-

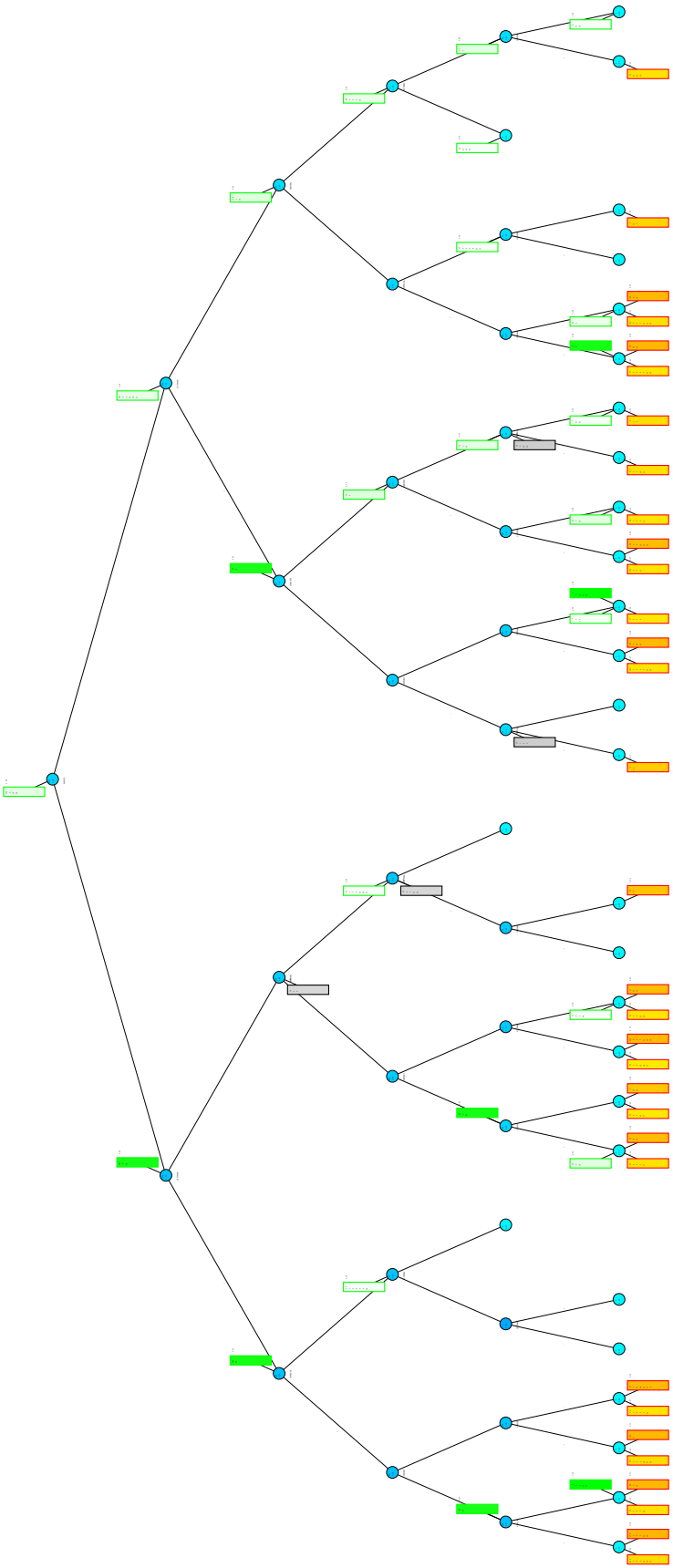


Figura 4.5: Árbol de Segmentación para datos municipales, 5 niveles

pio, mientras que los oscuros, a la derecha, se identifican con una apreciación global positiva. Los nodos de color gris proceden de la condición de parada de los estratos (véase 3.3.5). La probabilidad que se muestra en los nodos es la probabilidad estimada de apreciación global positiva.

La medida de contenido de información inicial es -0.644426 y la final es -0.427898 en el árbol de 3 niveles y -0.398596 en el de 5 niveles. En ambas medidas, se considera el nodo obtenido en el paso inicial para los municipios S_3, S_{16} y S_{18}.

Los nodos decisionales obtenidos después de la tercera iteración del algoritmo son:

$00d0 : [global \sim ((+)0.08, (-)0.92)] \wedge [municipio \in \{3, 16, 18\}]$

$10d1 : [espacio = (+)] \wedge [global \sim ((+)0.92, (-)0.08)] \wedge [municipio \in \{8, 10\}]$

$11d0 : [espacio = (-)] \wedge [global \sim ((+)0.05, (-)0.95)] \wedge [municipio \in \{4, 9, 12, 14, 17\}]$

$20d1 : [espacio = (+)] \wedge [optimismo = (+)] \wedge [global \sim ((+)0.92, (-)0.08)]$
$\wedge [municipio = 19]$

$22d1 : [espacio = (-)] \wedge [optimismo = (+)] \wedge [global \sim ((+)0.90, (-)0.10)]$
$\wedge [municipio = 8]$

$23d0 : [espacio = (-)] \wedge [optimismo = (-)] \wedge [global \sim ((+)0.09, (-)0.91)]$
$\wedge [municipio \in \{1, 19\}]$

$31d0 : [espacio = (+)] \wedge [optimismo = (+)] \wedge [integra = (-)] \wedge [global = (-)]$
$\wedge [municipio \in \{1, 2, 4, 5, 6, 14\}]$

$33d0 : [espacio = (+)] \wedge [optimismo = (-)] \wedge [habitabi = (-)] \wedge [global = (-)]$
$\wedge [municipio \in \{1, 7, 9, 12, 14, 17\}]$

$35d0 : [espacio = (-)] \wedge [optimismo = (+)] \wedge [conformi = (-)] \wedge [global \sim ((+)0.10, (-)0.90)]$
$\wedge [municipio = 6]$

$37d0 : [espacio = (-)] \wedge [optimismo = (-)] \wedge [habitabi = (-)] \wedge [global \sim ((+)0.05, (-)0.95)]$
$\wedge [municipio \in \{2, 5, 6, 10\}]$

Para no hacer demasiado extensa las lista de nodos, se omiten los nodos

procedentes del proceso terminal-divide (al alcanzar al condición de máximo nivel (véase 4.6.2) que se encuentran en el nivel 3 del árbol). Se detallan los mismos en los casos de interpretación que se presentan a modo de ejemplo. Para mayor detalle de los mismos, véase la figura 4.6.

En algunos municipios, una apreciación positiva o negativa del espacio en el municipio es suficiente para determinar una apreciación global positiva o negativa en el mismo y del mismo signo. En los municipios S_8 ($Cr = 0.5$) y S_{10} ($Cr = 0.46$) una apreciación positiva del espacio en su municipio refleja una apreciación global positiva, deducida del nodo $(10d1)$ ($p = 0.92$). La importancia relativa de los dos municipios en la regla es de 0.53 y 0.47, respectivamente. La importancia de esta reglas en sus municipios es del 50 y el 46 por ciento, respectivamente.

Por el contratrio, en los municipios S_4, S_9, S_{12}, S_{14} y S_{17}, una apreciación negativa del espacio en su municipio explica una apreciación global negativa, dada por el nodo $t_3(11d0)$ ($p = 0.05$). Las contribuciones absolutas de los estratos al nodo son valores entre 0.19 y 0.21, por lo que todos tienen la misma importancia en la regla. Las contribuciones relativas del nodo en los estratos son respectivamente 0.75, 0.69, 0.69, 0.75 y 0.71, con lo que esta regla representa porcentajes de alrededor el 70 por ciento de estos municipios.

A un lado y otro del árbol, hay municipios que precisan seguir el proceso recursivo para obtener una explicación de la apreciación global y cuáles son los aspectos parciales de apreciación que inducen una apreciación global positiva o negativa. Se destacan aquí, algunos nodos para los que aún teniendo apreciaciones parciales negativas expliquen una apreciación global positiva; y, para los que aún teniendo apreciaciones parciales positivas expliquen apreciación global negativa.

Así, por ejemplo en este último caso se encuentra el municipio S_8, a la derecha del árbol. Si se observa el nodo $22d1$, para este municipio, la apreciación negativa en el espacio y positiva en el optimismo social predicen una apreciación global positiva ($p = 0.9$) que representa el 29 por cierto del municipio ($Cr = 0.29$).

Si se prosigue el proceso recursivo a partir del nodo explorable en la posición

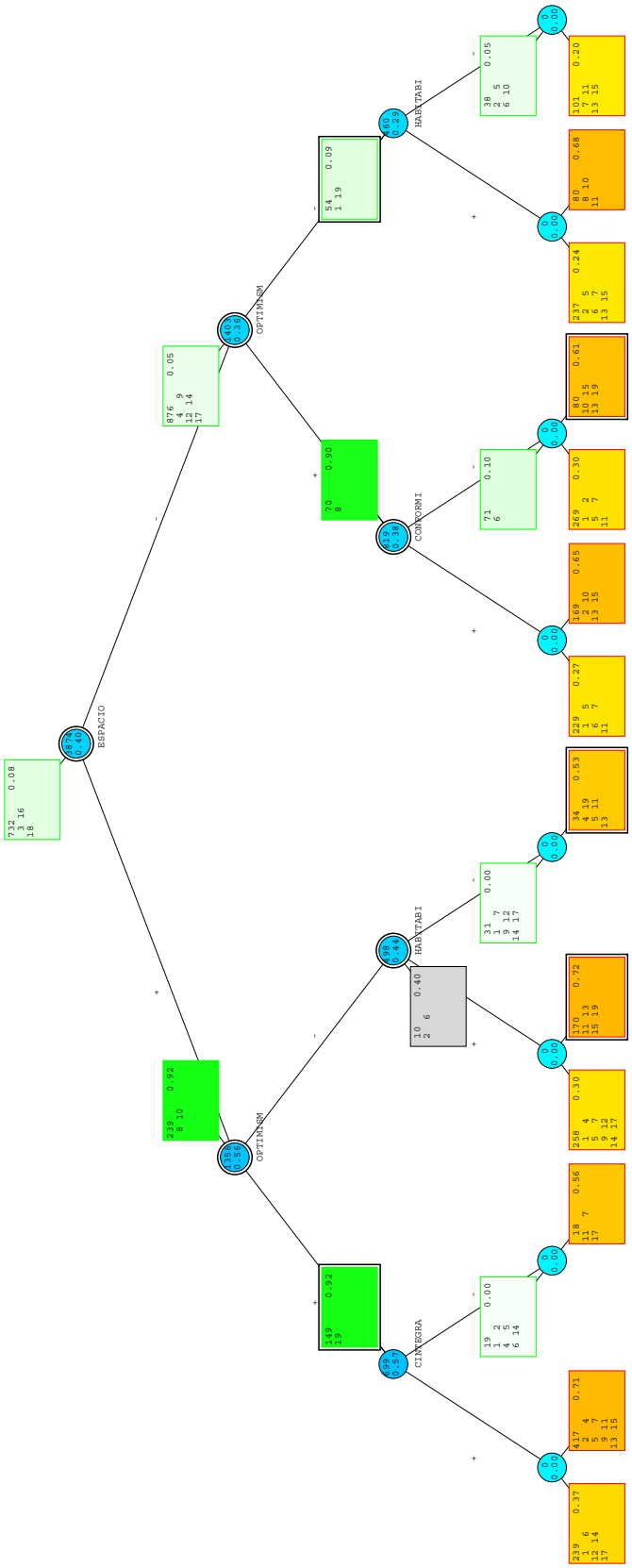


Figura 4.6: Árbol de Segmentación para datos municipales, 3 niveles

2,2 para el resto de los municipios y se observan los nodos que explican una apreciación global negativa aún teniendo apreciaciones parciales positivaS, ésto les sucede al municipio S_6 de una parte y a los municipios S_1, S_2, S_5, S_7 y S_9 de otra, que tienen una apreciación negativa del espacio en el municipio, positiva del optimismo social y negativa del conformismo social.

A la izquierda del árbol, si se observa el nodo $33d1$, para los municipios S_1, S_7, S_9, S_{12},S_{14} y S_{17} la apreciación positiva del espacio del municipio y negativa del optimismo social y de la habitabilidad explican una apreciación global negativa ($p = 0$), si bien la importancia relativa en sus respectivos municipios no supera el 5 por ciento para cada uno de ellos. Los cuatro últimos municipios compartían tambien la regla ($11d0$), anteriormente citada.

En la tabla siguiente, se presenta un extracto de la tabla de contribuciones absolutas y relativas correspondientes a los municipios S_4 y S_9. Se observa que comparten las 3 reglas que los caracterizan ya que la suma de las contribuciones relativas de estos nodos en ambos municipios es de 0.98.

	S_4		S_9	
	Cr	Ca	Cr	Ca
$11d0$	0.75	0.2	0.69	0.19
$30td1$	0.14	0.08	0.07	0.04
$32td0$	0.09	0.09	0.22	0.21

Se puede describir el municipio S_4 por el conjunto de objetos simbólicos ponderados:

$$S_4 : \{0.75[espacio = (-)] \wedge [global \sim ((+)0.05, (-)0.95)],$$

$$0.14[espacio = (+)] \wedge [optimismo = (+)] \wedge [integra = (+)] \wedge [global \sim ((+)0.71, (-)0.29)],$$

$$0.09[espacio = (+)] \wedge [optimismo = (-)] \wedge [habitabi = (+)] \wedge [global \sim ((+)0.3, (-)0.7)]\}$$

En (4.95) puede verse una descripción por objetos simbólicos del municipio S_{19}, que se refiere a un árbol de mayor número de niveles. Los ejemplos 3.4 y 4.1 son salidas referidas al árbol global. En la figura 4.6, se pueden observar más reglas de predicción.

4.9.4 Datos relacionados con la actividad laboral

Este ejemplo ilustra la aplicación de la técnica para datos simbólicos modales probabilistas y se tratan datos provEnientes de la encuesta de relación con la actividad laboral de un país de la Unión Europea. Esta encuesta se toma a todas las personas mayores de 16 años de los hogares seleccionados por un muestreo estratificado por ubicación geográfica. Se seleccionaron de esta base de datos los datos relativos a un trimestre y las personas que se encontraban trabajando en el momento de la encuesta. Cada unidad de datos es ponderada por un peso. A partir de este conjunto de datos original de 21198 individuos, se crearon 5305 unidades de datos descritos por datos simbólicos modales probabilistas, cruzando las variables región, sexo, edad (5 categorías), estado social, sector económico, profesión (10 categorías), si es asalariado o tiene otra relación con la actividad (ej: autónomo, empresario) y si trabaja jornada completa o parcial. En la creación de los datos simbólicos modales probabilistas correspondientes a los segmentos identificados por el cruce de las variables, se consideró el peso de las unidades muestrales originales (véase Stéphan et al., 2000).

Este ejemplo ilustra la explicación de las categorías de edad por las variables relacionadas con la actividad, derivada de la actividad económica (variable estrato). Ilustra cómo se puede aplicar la técnica presentada en esta Memoria para estas explicaciones de la edad de una parte y para la imputación de datos, de otra. Si bien estas imputaciones de edad, sexo, etc... se realizan, en las Oficinas de Estadística, en otro tipo de bases de datos, como son los datos censales. También se realiza imputación de datos en las encuestas industriales.

La variable de estratificación considerada en este ejemplo es la actividad económica (*cae*) y la variable clase la categoría de edad del empleado, considerados los menores de 35 años y los mayores de 35. Los predictores considerados son: tipo de empresa, pública o privada (*tipo*); búsqueda de trabajo (*busca*); cotización a la seguridad social (*segsoc*); sobrecualificación para la actividad realizada ($sobrecualif$); jornada laboral normal ($njornada$); deseo de cambio de jornada (*cambio*); experiencia, primer trabajo u otro (*experiencia*); estudia o está en formación en la actualidad (*estudia*); estado civil, soltero u otro (*estatus*); situación profesional, asalariado u otra relación con la actividad ($sprof$); y jornada completa o parcial ($jornada$).

Los sectores económicos[19] considerados son: 1, agricultura, ganadería y pesca; 3, manufactura; 4, electricidad, gas y agua; 6, comercio y reparaciones; 7, restauración; 8, transporte y comunicaciones; 10, finanzas; 11, administración pública; y 12, otros servicios.

Ejemplos de elementos del conjunto Ω son:

$$\omega : (tipo(\omega) = (priv.(0.54), púb.(0.46)), busca(\omega) = no, jornada(\omega) = completa,$$
$$segsoc(\omega) = contrib., sobrecual(\omega) = (sí0.05, no0.95), cambio(\omega) = (no0.95, sí0.05),$$
$$njornada(\omega) = (usual0.95, inusual0.05), exper(\omega) = (1ertra0.27, no1ert0.73),$$
$$estudio(\omega) = (sí0.06, no0.94), sexo(\omega) = hombre, status(\omega) = nosolt,$$
$$sprof(\omega) = asalariado, cae(\omega) = 12, edad(\omega) = 35 - 64)$$

[19] Se eliminaron en este ejemplo los trabajadores de los sectores 2, minería y 9: intermediarios financieros; por no tener demasiada representación. También se prescindió del sector 5, construcción. En el conjunto de 21198 unidades de datos originales, estos sectores ya han sido eliminados.

$$\omega : (tipo(\omega) = privado, busca(\omega) = (sí0.42, no0.58), jornada(\omega) = parcial,$$
$$segsoc(\omega) = (contribuyente0.47, no0.53), sobrecualif(\omega) = (sí0.56, no0.44),$$
$$cambio(\omega) = (no0.41, sí0.59), njornada(\omega) = (usual0.26, inusual0.74),$$
$$exper(\omega) = (1ertra0.68, no1ert0.32), estudio(\omega) = (sí0.28, no0.72), sexo(\omega) = f,$$
$$status(\omega) = soltero, sprof(\omega) = asalariado, cae(\omega) = 6, edad(\omega) = 15 - 34)$$

La figura 4.7 muestra el árbol obtenido hasta el nivel 4 del árbol. La medida de contenido de información inicial es de -0.678264 y la final de -0.582698. La disminución de la entropía no es demasiado acusada indicando la necesidad de incorporar nuevas variables como predictores si se desea una calidad del árbol mayor.

Los nodos decisionales hasta el nivel 4 del árbol son:

$21d1 : [estatus = soltero] \wedge [estudio = estudia] \wedge [edad \sim ((15 - 34)0.97,$
$(35 - 64)0.03)] \wedge [cae \in \{1, 4, 12, 10, 6, 11, 3, 8, 7\}]$

$23d0 : [estatus = nosolte] \wedge [sprof = anoasal] \wedge [edad \sim ((< 34)0.18,$
$(35 - 64)0.82] \wedge [cae \in \{11, 7\}]$

$30d1 : [estatus = soltero] \wedge [estudio = noestud] \wedge [experien = 1ertrab] \wedge$
$[edad \sim ((15 - 34)0.84, (35 - 64)0.16)] \wedge [cae \in \{4, 10, 6, 8, 7\}]$

$36d0 : [estatus = nosolte] \wedge [sprof = anoasal] \wedge [experien = 1ertrab]$
$\wedge [edad \sim ((15 - 34)0.18, (35 - 64)0.82)] \wedge [cae = 3]$

$37d0 : [estatus = nosolte] \wedge [sprof = anoasal] \wedge [experien = no1ert]$
$\wedge [edad \sim ((15 - 34)0.18, (35 - 64)0.82)] \wedge [cae \in \{1, 10\}]$

$40d1 : [estatus = soltero] \wedge [estudio = noestud] \wedge [experien = 1ertrab] \wedge [sexo$
$= hombre] \wedge [edad \sim ((15 - 34)0.89, (35 - 64)0.11)] \wedge [cae = 3]$

$48d0 : [estatus = nosolte] \wedge [sprof = asalariado] \wedge [buscatr = no] \wedge [experien$
$= 1ertrab] \wedge [edad \sim ((15 - 34)0.15, (35 - 64)0.85)] \wedge [cae = 8]$

$4, 12d0 : [estatus = nosolte] \wedge [sprof = anoasal] \wedge [experien = 1ertrab] \wedge [sexo$
$= hombre] \wedge [edad \sim ((15 - 34)0.09, (35 - 64)0.91)] \wedge [cae = 12]$

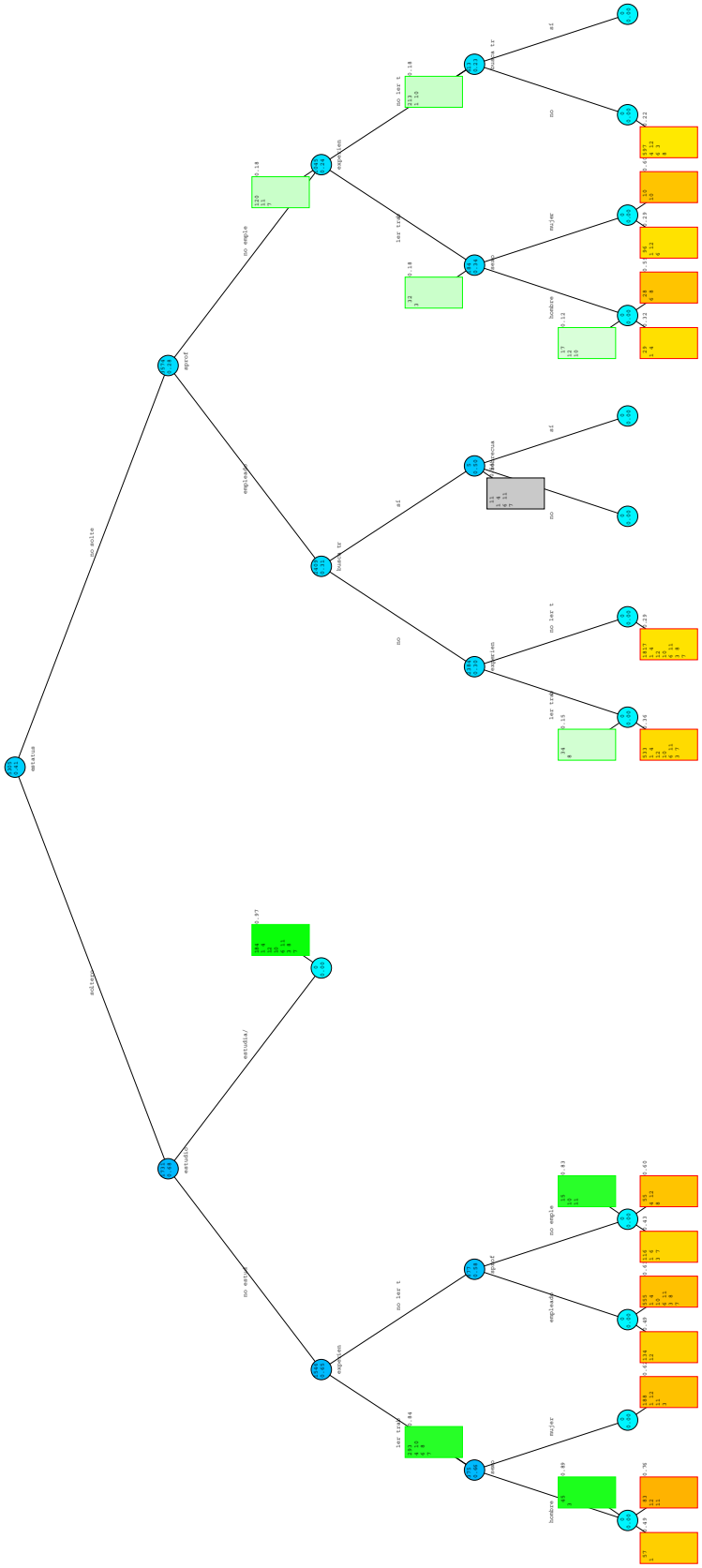


Figura 4.7: Árbol de Segmentación para datos relacionados con la actividad, 4 niveles

Se excluyen de la lista los nodos decisionales de muy bajo peso y la lista completa de los nodos procedentes de un proceso terminal-divide al alcanzarse la condición de máximo nivel. Estos nodos pueden observarse en el árbol mostrado en la figura 4.7. Las especificaciones en cuanto a la forma y color de los nodos y su contenido viene explicada en 4.9.1, así como la explicación de la identificación de los nodos. Esta identificación termina con el número 0, si el nodo explica la clase mayor o igual que 35 años (de intensidad oscura) y con un 1 si el nodo explica la clase menor que 35 años (de intensidad clara). Los pesos de los nodos aparecen sin cifras decimales por aproximación al entero más próximo. Estos números no son necesariamente enteros al ser la suma de las probabilidades de los individuos en los nodos, ya que los datos de entrada son modales probabilistas.

La variable más informativa para determinar la edad es el estado civil. Para los de estado civil soltero, a la izquierda del árbol, la siguiente variable más informativa es si están estudiando o en formación o no. En todos los sectores se verifica la regla $21d1$, que son los que estudian o están en formación, que explica la categoría menor de 35 años que tienen de contribuciones relativas entre 0.02 y 0.06, con lo que representan porcentajes pequeños de explicación de la edad, en estas actividades económicas. Para los que no están estudiando, la más informativa es la experiencia laboral. Para los que desempeñan su primer trabajo, el sexo y para los demás si son asalariados o tienen otra relación con la actividad.

Para los no solteros, a la derecha del árbol, la siguiente variable más informativa es si son asalariados o tienen otra relación con la actividad. Para los asalariados, si buscan otro trabajo o no y finalmente la experiencia profesional. Para los no asalariados, la experiencia profesional a un primer nivel, y el sexo y la búsqueda de otro empleo, respectivamente para los no expertos y los experimentados. Los nodos $36d0$ y $37d0$, ambos predicen la clase mayor o igual a 35 años. Tienen todos los predictores comunes, sin embargo, las categorías que definen las reglas en el último nivel son antagónicas. Así, para el sector 3: manufactura, se deduce que *si una persona es no soltera, no asalariada y es su primer trabajo*

entonces es mayor de 35 años ($Cr = 0.05$). Pero para el sector 1: agricultura, ganadería y pesca y 10: finanzas, se deduce que *si una persona es no soltera, no asalariada y no es su primer trabajo entonces es mayor de 35 años* ($Cr = 0.23$ para el sector agricultura, ganadería y pesca, $Cr = 0.19$ para el sector finanzas).

Como ejemplo ilustrativo de la descripción de un sector, el sector 7, restauración se representa por:

$$\begin{aligned} \textit{Restauración} \quad : \quad & \{0.03(21d1), 0.28(23d0), 0.11(30d1), \\ & 0.12(42td1), 0.03(43td0), 0.09(48td0), 0.31(49td0)\} \end{aligned}$$

Es decir, se puede describir el sector restauración por el conjunto de objetos simbólicos ponderados:

$$\{0.03[estatus = soltero] \wedge [estudio = estudia] \wedge [edad \sim ((< 35)0.97, (\geq 35)0.03)],$$
$$0.28[estatus = nosolte] \wedge [sprof = noasasal] \wedge [edad \sim ((< 35)0.18, (\geq 35)0.82)],$$
$$0.11[estatus = nosolte] \wedge [sprof = noasasal] \wedge [experien = 1ertrab] \wedge$$
$$[edad \sim ((15 - 34)0.18, (35 - 64)0.82)],$$
$$0.31[estatus = nosolte] \wedge [sprof = asal.] \wedge [busca = no] \wedge [experien = no1ert]$$
$$\wedge [edad \sim ((15 - 34)0.29, (35 - 64)0.71)], 0.27Otros_nodos\}$$

excluídos los nodos $42td1$, $43td0$ y $48td0$ al tener una medida de contenido de información baja. El sector restauración se explica por estas cuatro reglas que representan los pesos respectivos 0.3, 0.28, 0.11 y 0.31. El sector no queda explicado en un 27 por ciento a este nivel del árbol.

4.9.5 Datos probabilistas SES, 1995

El algoritmo de Segmentación para datos estratificados ha sido aplicado a un conjunto de datos modales probabilistas obtenidos a partir del conjunto de datos

de la aplicación 4.9.1 en relación a la base de datos *"T25IT Italy: Monthly earnings por local unit size, NACE (economic activity) and ISCO (profession)"*[20]. Se obtuvieron 720 unidades de datos descritos por datos modales probabilistas considerando los pesos de las unidades de datos originales cruzando las variables sector económico (22 categorías), profesión (7 categorías) y tamaño de la empresa (5 categorías).

El conjunto Ω contiene 720 ($n = 720$) unidades de datos, el conjunto $E \subset \mathcal{P}(\Omega)$ contiene subconjuntos de unidades de datos que pertenecen al mismo sector económico $NACE$. El número de estratos es 5 al reducir la taxonomía original a 5 categorías ($m = 5$). La variable clase Z es la variable binaria $manual$[21] ($s = 2$), y los predictores son variables relacionadas con el sexo, salarios y horas trabajadas por los empleados: $sexo$, $h50$, $sal75$, $b25$, $salh50$ y cvm. Los sectores $NACE$ considerados son: *minería*; $manufactura$; *electricidad*, gas y agua; *construcción*; y, *servicios*. Véase 4.9.1 para una descripción más detallada de los datos de este ejemplo. Un ejemplo de unidad de datos se representa por los datos modales probabilistas:

$$\omega : (sexo(\omega) = (f(0.64), h(0.36)), sal75(\omega) = sí, b25(\omega) = (sí(0.04), no(0.96))$$
$$salh50(\omega) = (sí(0.64), no(0.36)), nace(\omega) = servicios, manual(\omega) = no)$$

que representa un conjunto de individuos del sector servicios de profesión no manual con salario bruto medio hora menor que el último cuartil y con distribuciones de probabilidad para sexo $(f(0.64), h(0.36))$, para $b25(\omega)$, $(sí(0.04), no(0.96))$ y para $salh50(\omega)$, $(sí(0.64), no(0.36))$

La figura 4.8 muestra el árbol obtenido hasta el tercer nivel del árbol. La medida de contenido de información inicial es -0.677260 y la final -0.432755.

[20] SES: Statistics on the Structure and Distribution of Earnings.

[21] Las cuatro grandes categorías de profesión son: profesionales liberales, técnicos, administrivos y empleados manuales.

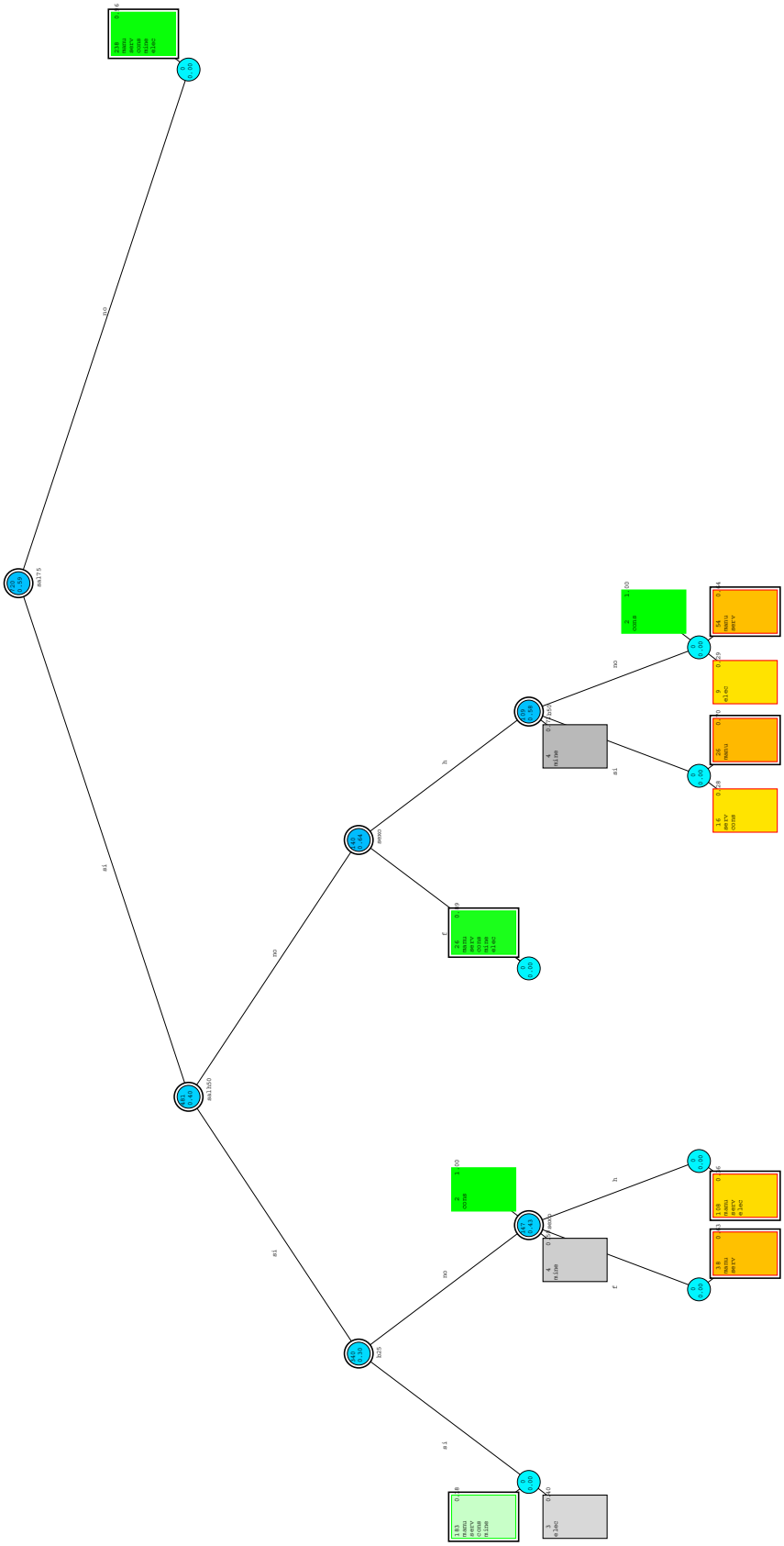


Figura 4.8: Árbol de Segmentación para los datos probabilistas SES, 4 niveles

Los nodos de intensidad clara predicen la clase manual y los de intensidad oscura, la clase no manual. Los nodos decisionales obtenidos hasta este nivel son:

$11d1 : [sal75 = no] \wedge [manual \sim (no0.96, sí0.04)] \wedge [nace \in \{manufact, servicios, construc, minería, electric\}]$

$30d0 : [sal75 = sí] \wedge [salh50 = sí] \wedge [b25 = sí] \wedge [manual \sim (no0.18, sí0.82)] \wedge [nace \in \{manufact, servicios, construc, minería\}]$

$31d1 : [sal75 = sí] \wedge [salh50 = sí] \wedge [b25 = no] \wedge [manual = no] \wedge [nace = construc]$

$32d1 : [sal75 = sí] \wedge [salh50 = no] \wedge [sexo = f] \wedge [manual \sim (no0.89, sí0.11)] \wedge [nace \in \{manufact, servicios, construc, minería, electric\}]$

$47d1 : [sal75 = sí] \wedge [salh50 = no] \wedge [sexo = h] \wedge [b50 = no] \wedge [manual = no] \wedge [nace = construc]$

La tabla 4.5 muestra los valores de las contribuciones relativas y absolutas, para estos nodos decisionales. Observando esta tabla se puede deducir que todos los sectores económicos comparten la regla $11d1$ teniendo una importancia relativa en los mismos que ronda del 30 al 35 por ciento. La regla 30d0 es compartida por los sectores *minería*, *manufactura*, *construcción* y *servicios*, teniendo una importancia relativa en los tres primeros sectores de alrededor un 30 por ciento, mientras que en el sector *servicios* decrece a un 11 por ciento. Todos los sectores menos *construcción* comparten la regla $32d1$ con una importancia relativa del 4 por ciento en todos los sectores.

	min		*manu*		*elec*		*const*		*serv*	
	Cr	Ca	Cr	Ca	Cr	Ca	Cr	Ca	Cr	Ca
$11d1$	0.34	0.05	0.32	0.57	0.34	0.05	0.3	0.05	0.37	0.28
$30d0$	0.37	0.05	0.32	0.77			0.28	0.05	0.11	0.13
$31d1$							0.08	1		
$32d1$	0.04	0.05	0.04	0.6	0.04	0.05	$\times$	0.04	0.04	0.26

El símbolo $\times$ especifica un valor muy pequeño

Tabla 4.5: Datos SES probabilistas. Tabla de contribuciones relativas y absolutas

Los sectores *minería* y *manufactura* comparten tres reglas $11d1$, $30d0$ y $32d1$

con importancias relativas parecidas en ambos estratos, ($Cr = 0.34$, 0.37 y 0.04 para el sector *minería* y $Cr = 0.32$, 0.32 y 0.04 para el sector $manufactura$), con lo que se puede concluir que al 70 por ciento la explicación de la profesión manual en ambos sectores, se debe a las mismas reglas de predicción. El sector *servicios* también comparte estas tres reglas. La primera y la tercera tienen importancias similares a los sectores anteriores, si bien la importancia relativa de la segunda decae al 11 por ciento.

Las contribuciones absolutas reflejan la importancia relativa que tienen los sectores en la explicación de cada uno de los nodos.

Como ejemplo ilustrativo, el sector *minería* (al nivel 3) se describe por:

$$\text{minería} : \{0.34[sal75 = no] \wedge [manual \sim (no(0.96), sí(0.04)],$$
$$0.37[sal75 = sí] \wedge [salh50 = sí] \wedge [b25 = sí] \wedge [manual \sim (no(0.18), sí(0.82))],$$
$$0.04[sal75 = sí] \wedge [salh50 = no] \wedge [sexo = f] \wedge [manual \sim (no(0.89), sí(0.11))],$$
$$0.25Otros\}$$

4.9.6 Conclusiones

El método presentado en esta Memoria se ha aplicado a diversas bases de datos, algunas de ellas incluidas en el proyecto SODAS[22] que han servido de bancos de pruebas para validar los métodos. Esta sección presenta algunas conclusiones de líneas muy generales desde un enfoque global del Analista de Datos y no desde un enfoque especializado de los datos y presenta una visión panorámica de las aplicaciones realizadas.

En primer lugar cabe destacar que los datos almacenados en las Oficinas de Estadística, son en ocasiones datos poblacionales y si no es así, por lo general son representativos de las poblaciones que representan. Por tanto, la utilidad descriptiva del método expuesto es de gran importancia.

[22] ESPRIT IV- 20821 SODAS

Las consolidaciones de datos que se realizan en la actualidad tanto para reducir los volúmenes de datos como para preservar la confidencialidad de los mismos, por lo general haceN referencia a datos medios y coeficientes de variación de grupos de unidades muestrales. En general, estas agrupaciones se realizan por combinación de las categorías de varias variables. Cada combinación de categorías identifica el grupo o segmento que representa. Este es el caso de la base de datos presentada en 4.9.1, en la que la consolidación se realiza por agrupación de unidades muestrales que son ponderados por su peso. Para el resto de variables, las consolidaciones son representadas por medias y coeficientes de variación de las unidades muestrales que representan.

Con el Análisis de Datos Simbólicos, se enriquecen los medios de consolidación de datos. De una parte, por la generalización de grupos de individuos por datos simbólicos (véase 1.4.5) y de otra porque estos datos pueden ser analizados. Estos datos simbólicos consolidan más información y son de mayor complejidad que las consolidaciones anteriores, ya que son datos multievaluados, distribuciones de probabilidad o datos de intervalo. Las ventajas de este nuevo sistema de representación están sintetizadas en el Prólogo y en 1.1 y 1.6. Gracias a esto, otras consolidaciones de datos se pueden realizar almacenando para subgrupos de unidades muestrales los datos simbólicos que los representen (véanse aplicaciones 4.9.4 y 4.9.5).

Estos datos permiten conservar la confidencialidad de los datos, reducen los volúmenes de las bases de datos y representan la *variabilidad* de las unidades muestrales del grupo que consolidan. Además, permiten comparar con los mismos criterios de consolidación, unidades de datos consolidadas de distintas bases de datos.

También, la consolidación por datos simbólicos se puede realizar longitudinalmente en encuestas que las unidades muestrales sean entrevistadas periódicamente. Por ejemplo, en la encuesta de relación con la actividad en la que los mismos individuos son entrevistados trimestralmente.

Independientemente de la consolidación, las encuestas en general suelen tener elementos controlados, ya sea por estratificación o no. Así por ejemplo, la encuesta de relación con la actividad que se aplica a los hogares (y dentro de cada hogar, a todos los mayores de 16 años) el muestreo considera como variables de estratificación de los hogares, dependiendo de los paises, la ubicación geográfica, el tamaño de los municipios, el nivel social de los hogares,... y se observan en los individuos variables socio-demográficas y variables que tienen relación con la actividad.

En otros encuestas industriales se consideran como variables de estratificación el sector económico de la empresa, el tamaño de la misma según diversos parámetros, etc...

En otras encuestas, como la del uso del tiempo, son elementos controlados de la encuesta para los que se establecen cuotas uniformes, el día de la semana a que se refiere la encuesta, el sexo y la edad del entrevistado. Las variables de la encuesta se refieren al uso del tiempo, profesión, etc...

La consolidación de datos mediante segmentos identificados por variables suelen contener, en el caso de que existan, las variables de estratificación o postestratificación y las variables controladas y pueden tener además otras variables de identificación de subgrupos.

Descendiendo al método para datos estratificados de esta Memoria, éste se puede aplicar a datos no consolidados (véase aplicación 4.9.3) en que las variables son monoevaluadas, con independencia de que existan estos segmentos de identificación o no. También se puede aplicar el método a datos consolidados no probabilistas, según criterios de consolidación previos a la consolidación por datos simbólicos (véase aplicación 4.9.1). Y, finalmente se puede aplicar a datos consolidados por datos simbólicos, siendo éstos modales probabilistas para los predictores y monoevaluados para las variables clase y estrato (véanse aplicaciones 4.9.4 y 4.9.5). Además, en todos los casos se permite en el método que las unidades de análisis sean ponderados por pesos.

Como en todo Análisis de Datos, tanto la semántica de las variables que constituyen la estratificación en el caso de que ésta exista, como de las variables que definen los segmentos de consolidación y del resto de variables es de importancia capital para determinar el papel que las variables juegan en el método. Una variable estrato es por lo general una variable o combinación de variables que definen la estratificación o los segmentos de la población. La variable clase es una variable de interés en explicar. La variable clase puede ser una de las variables que definen los segmentos (Ej: *administrativo* en 4.9.1) o bien una de las variables no controladas por la encuesta o no identificadora de un segmento (Ej: *apreciación_global* en 4.9.3). Las variables predictoras pueden ser tanto variables que definen los segmentos como del conjunto del resto de variables. El interés de la explicación de la variable clase puede ser meramente descriptivo de las clases por los predictores, teniendo en cuenta los estratos. Se describen los estratos por reglas de predicción de las clases, se detectan las diferencias de los estratos respecto a estas reglas, las reglas que comparten, etc...

En otros casos, existe de forma natural una única variable de estratificación (Ej: *municipio* en 4.9.3) y tanto la variable clase como las variables predictoras son observadas.

También este método proporciona un mecanismo de imputación de datos ausentes en las encuestas que se aplican y puede ser de aplicación en datos censales, encuestas industrales, etc...

Si bien por criterios descriptivos puede estar justificado el uso de una variable no controlada por la encuesta como variable estrato, puede ser aventurado su utilización, dado que puede ser más bien una variable que influye por sí misma en la explicación de la variable clase, impidiendo ver la influencia directa de las demás. A modo de ejemplo ilustrativo, si en la encuesta de uso del tiempo, se desea explicar el uso del tiempo en ambos sexos, derivado de la profesión que tienen, se tiene que al no estar controlada la variable profesión en los datos de la encuesta, la profesión determina como un predictor el sexo. Si se desea

una explicación de uso del tiempo en ambos sexos, derivado de la profesión, y eliminando la influencia de la profesión como predictor del sexo, se debería tener en la muestra diseño una distribución uniforme de ambos sexos en cada profesión. Aún estas consideraciones, también puede estar justificado el uso del método en los datos originales si se desea una mera descripción de los datos, sin ambiciones acerca de las potencialidades del método en cuanto a la explicación de los estratos.

Esta técnica se puede aplicar en muchos otros ámbitos en los que existe una variable clase de interés y una variable de estratificación, como pueden ser en marketing, publicidad, análisis de riesgos, donde tradicionalmente los métodos de Segmentación se utilizan habitualmente para la toma de decisiones. La variable estrato puede ser además cualquier variable categórica. En particular, es de gran utilidad si estas variables categóricas tienen un elevado número de categorías.

4.10 Conclusión

En este capítulo se han proporcionado criterios específicos para el método de Segmentación para datos estratificados presentado en el capítulo 3, que comprenden datos de entrada monoevaluados y modales probabilistas, se ha caracterizado el árbol obtenido y se ha destacado su relevancia frente a los algoritmos tradicionales de Segmentación. Se ha descrito la generalización de los estratos por conjuntos de objetos simbólicos ponderados y se han proporcionado medidas para la interpretación de los mismos, así como medidas de interpretación de compartición de reglas de predicción para conjuntos de estratos.

Se han incorporado algunas extensiones como el tratamiento de unidades de datos ponderadas y la incorporación de probabilidades 'a priori' de las clases. Se han propuesto normas de predicción para nuevos individuos. También se han propuesto algunas mejoras al método presentado en el capítulo 3 que proporcionan árboles de mayor calidad. Se ha destacado cómo el marco general simbólico del método y la representación del árbol y nodos por objetos simbólicos permiten

su extensión a otros tipos de datos simbólicos y otras expresiones de incertidumbre. Se han propuesto tipos de objetos simbólicos, relaciones producto y medidas concretas relacionando estas propuestas con los antecedentes del tratamiento de la incertidumbre en Segmentación presentada en el capítulo 2 y con los conceptos de datos y objetos simbólicos del capítulo 1.

Finalmente se han presentado algunas aplicaciones destacándose normas de interpretación del árbol, nodos y estratos y se han aportado algunas conclusiones de estas aplicaciones.

Capítulo 5

Implementación del Método

Los métodos presentados en esta Memoria han sido implementados en un programa de *software* llamado SDT (*Strata Decisión Tree*) para la construcción del árbol de segmentación para datos estratificados y SDTEDITOR, un editor gráfico para estos árboles. Los *software* SDT V2.22b y SDTEDITOR[1] V2.22 se han incorporado al *software* SODAS[2] 1.04 dentro del proyecto ESPRIT 20821-SODAS (Symbolic Official Data Analysis System) del IV Programa Marco de la Comunidad Europea.

5.1 Especificaciones

La elaboración del *software* ha seguido el plan de gestión y calidad de *software* aprobado por el consorcio de SODAS y expuestos en Bouillet y Grandin, 1997 y Muenier, 1997, respectivamente. Los estándares establecidos por el consorcio exigían realizar una implementación del algoritmo en programación orientada a objetos con el lenguaje *Visual C++*[3] *v 5.0*, así como el uso de las librerías *MFC* (*Microsoft Foundation Class*) de *Microsoft*.

[1] SDT 2.22b y SDTEDITOR 2.22, Copyright Universidad Complutense de Madrid, 1999.
[2] SODAS 1.04, Copyright CISIA Ceresta, 1999. http://www.cisia.com/
[3] Copyright 1994-1997 Microsoft Corporation

Todos los módulos desarrollados comparten el uso de unas librerías (SOM[4]-*Symbolic Object Management*) que gestionan los objetos y datos de entrada (véase Csernel, 1998). Los conjuntos de datos que se analizan en SODAS se llaman ficheros SODAS. Para una descripción del formato de este fichero, véase Csernel, 1998. La librería SOM proporciona así mismo un *parser* o analizador sintáctico que comprueba la sintaxis de los ficheros SODAS. Se ha establecido un lenguaje común de representación de los objetos simbólicos en los ficheros de salida (Csernel, 1998).

El módulo DB2SO[5] 2.0 (*Extraction of Symbolic Objects from Data Bases*), incluido en el *software* SODAS 1.04, permite crear ficheros de datos SODAS a partir de bases de datos relacionales (véase Hébrail, 1999).

El *workbench* (WB), desarrollado por CISIA, es el *núcleo* final del *software* SODAS construido a partir de los módulos que lo componen. Muestra una interfaz gráfica que recoge los parámetros de los módulos, pasa los parámetros a los módulos mediante un fichero y envía los módulos a ejecución, mostrando posteriormente el fichero de resultados si la ejecución es satisfactoria o el fichero de diagnósticos en caso contrario. Los módulos que producen su interfaz gráfica propia, el *workbench* la envía a ejecución. También muestra la ayuda en HTML de los módulos que la proporcionan. La interfaz de SODAS gestiona los métodos de análisis que se aplican a los ficheros de datos SODAS, mediante un encadenamiento gráfico de módulos de SODAS, permitiendo al usuario añadir o quitar módulos a su elección.

En la cabecera de cada encadenamiento se encuentra el módulo BASE que selecciona el conjunto de datos de análisis. Para una descripción del *workbench* véase Morineau, 1998. Para una descripción de la interfaz final de usuario véase Morineau y Leprince, 1999.

Para la interfaz gráfica de SDTEDITOR se ha seguido la guía de estilo de la

[4]SOM v1.5, Copyright INRIA, 1999
[5]DB2SO 2.0, Copyright EDF-DER, 1998.

interfaz de usuario (véase Noirhomme-Fraiture y Rouard, 1997).

Las interfaces externas de SDT y SDTEDITOR se muestran en la figura 5.1. La interfaz de SOM a SDT permite la transferencia de la descripción de

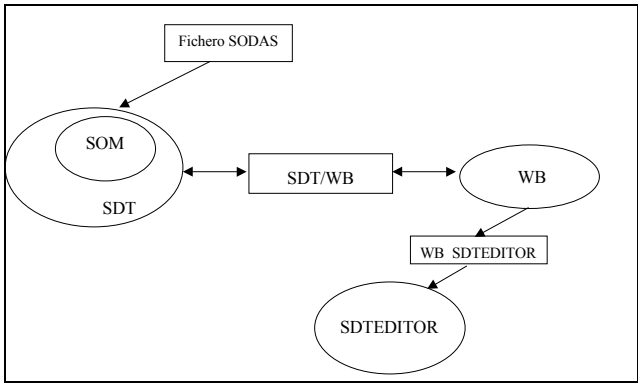


Figura 5.1: Interfaces externas de SDT y SDTEDITOR en SODAS

los datos de entrada y la librería SOM. La interfaz de WB a SDT permite la transmisión del fichero de parámetros. La interfaz de SDT a WB permite la transferencia de los ficheros de resultados, diagnósticos y gráfico. La interfaz de WB a SDTEDITOR permite la transmisión del fichero gráfico. El *workbench* lanza la ejecución de SDT y SDTEDITOR y la ayuda en HTML del módulo SDT. Esta ayuda contiene información general del método y tipos de variables de entrada. Para una descripción más detallada de estas interfaces, véase Bravo (2000a, 2000b).

El programa SDT lee un fichero SODAS y un fichero de parámetros que le proporciona el *workbench* y crea el árbol para datos estratificados correspondiente. Como resultado de la ejecución produce un fichero de resultados, un fichero de diagnósticos y un fichero gráfico. El programa SDTEDITOR es un editor gráfico que lee el fichero gráfico salida del programa SDT y visualiza el árbol construido.

Nota 5.1 ***Tipos de nodos****. Se distinguen en el árbol cuatro tipos de nodos: los explorables; los decisionales, que se obtienen por la aplicación de la condición de*

nodo decisional (véase 3.3.4); los terminales, que se obtienen por la aplicación de la condición de parada de los estratos (véase 3.3.5); y los terminal-divide, que se obtienen por alguna condición de parada del algoritmo a partir del proceso terminal-divide (véase 4.6.2).

5.2 Entrada

5.2.1 SDT v2.22b

La interfaz gráfica de SODAS permite al usuario identificar el fichero de datos SODAS de entrada, los predictores, la variable clase y la variable estrato, así como los parámetros de entrada al programa SDT. Todos los parámetros recogidos por el *workbench* tienen valores enteros. Estos parámetros son:

- *Máximo porcentaje de valores no observados.* Cuando el porcentaje de valores no observados en un predictor es menor del valor especificado, entonces los individuos con valor no observado en ese predictor son eliminados del análisis. En caso contrario, el predictor no es analizado. Valores posibles de 0 a 100. Valor por defecto: 10.

- *Condición de nodo decisional.* Probabilidad mínima de una clase para un estrato para que el estrato forme parte de un nodo decisional. El valor especificado se divide por 100. Valores posibles de 0 a 100. Valor por defecto: 80 (véase 3.2.4).

- *Condición de parada para los estratos.* Estratos con peso menor en un nodo explorable al especificado, no continuan el proceso recursivo. Valor por defecto: 5. (véase 3.2.4).

- *Máximo nivel de profundidad.* Valores posibles de 1 a 15. Valor por defecto: 5. (véase 4.6.2).

- *Mínimo incremento relativo de la medida de contenido de información.* Valores posibles de 1 a 10000. El valor especificado se divide por 10000. Valor por defecto: 1. Este valor es equivalente a no utilizar esta condición de parada (véase 4.6.2).

- *Salida de un fichero de resultados corto o largo.* Valor por defecto: largo.

- *Mínima importancia relativa de un nodo en la descripción de un estrato.* El valor especificado se divide por 100. Los nodos cuya contribución relativa sea menor que la especificada no se muestran en la descripción de los estratos mediante objetos simbólicos. Valores posibles de 0 a 100. Valor por defecto: 10. (véase 3.2.3 y 4.4.3).

- Existen además dos parámetros ocultos en la interfaz de usuarios de SODAS 1.04. El primero de ellos es el *procesamiento el primer nodo* por la condición de nodo decisional. Por defecto, los estratos con una frecuencia menor que el parámetro *condición de parada para los estratos* no entran a formar parte del nodo inicial. Con el primer parámetro se impide esta exclusión de estratos en el nodo inicial. El segundo parámetro oculto es la obtención de un *fichero adicional de salida* con resultados intermedios del proceso.

5.2.2 Ficheros de datos SODAS

La obtención de datos y objetos simbólicos se puede realizar a través del módulo DB2SO (véase Stéphan et al., 2000 y Hébrail, 1999) a partir de bases de datos relacionales mediante consultas a la base de datos, después de establecer una conexión ODBC. Los datos simbólicos pueden ser así mismo proporcionados por un experto siempre y cuando se suministren en una base de datos relacional. Los ficheros de datos SODAS contienen además metadatos, como por ejemplo reglas de no aplicabilidad, que puede ser analizados por SDT. Véase Bravo, 1999g para una descripción detallada de la construcción de ficheros de datos SODAS entrada

de SDT a partir de DB2SO y véase Hébrail, 1999 para una descripción más detallada del uso del módulo DB2SO.

Los datos que se puede analizar pueden ser datos agregados, conocidas una variable estrato y una variable clase (véase 4.9.6). Las agregaciones dan lugar a distribuciones de probabilidad en los predictores.

5.2.3 SDTEDITOR v2.22

La interfaz gráfica de SDTEDITOR lee automáticamente el fichero gráfico producido por SDT, que le proporciona el *workbench* y permite a partir de menús, listas deplegables, cajas y botones cambiar los parámetros. Algunos de estos parámetros influyen en el tamaño del árbol en la ventana de visualización. Los parámetros referidos a nodos se refieren a nodos decisionales, nodos terminal-divide y terminales. Los parámetros de SDTEDITOR son:

- *Número máximo de estratos* mostrados en un nodo. Valores posibles de 1 a 10. Valor por defecto: 4.
- Número máximo de *caracteres por estrato* que se muestran en un nodo. Valores posibles de 0 a 4. Valor por defecto: 4.
- Número máximo de *caracteres por línea* mostrada en un nodo. Valores posibles de 1 a 12. Valor por defecto: 10.
- *Tamaño* y *tipo de fuente*.
- *Mínimo peso en un nodo* para ser mostrado. Valor por defecto: *Valor de relevancia* que es calculado por SDT y es:

$$Valor\ de\ relevancia = \min_{1,..,m} \{\frac{2.5}{100} Card(S_i)\} \tag{5.1}$$

  el 2.5% del peso mínimo de uno de los estratos analizados en la muestra diseño.

- Posibilidad o no de visualizar los nodos decisionales, terminales o terminal-divide e información relativa a estos nodos. Véase en figura 5.2, un ejemplo de un árbol que contiene sólo nodos explorables.

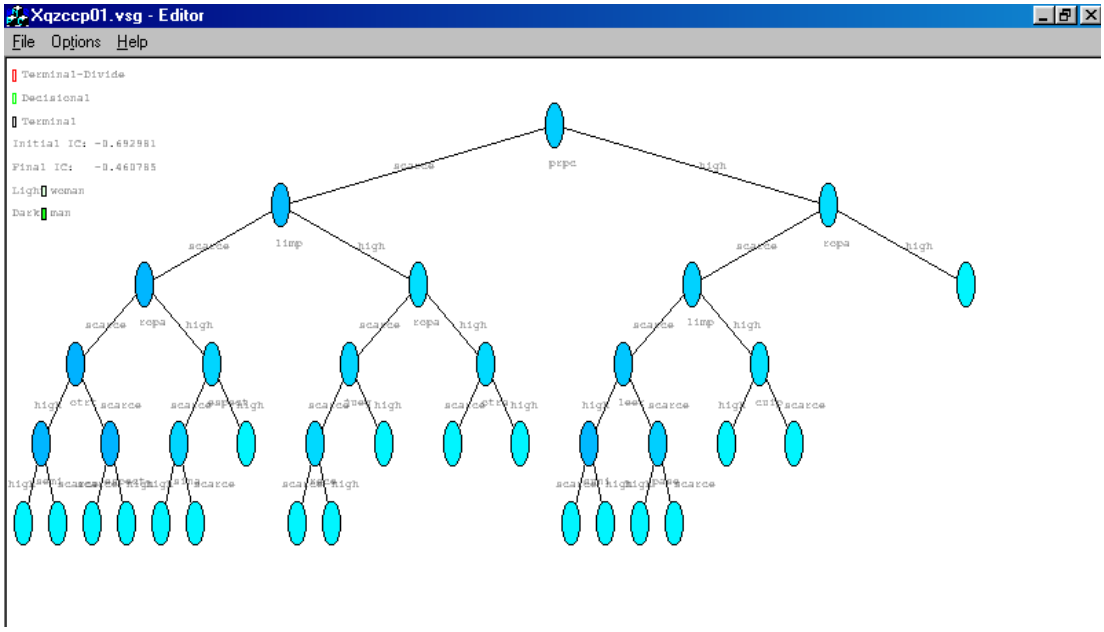


Figura 5.2: SDTEDITOR. Árbol con nodos explorables.

- Opciones de impresión: previsualización, opciones de la impresora e impresión.

5.3 Requisitos y criterios adoptados en la implementación

Los requisitos mínimos para el *software* dependen de la máquina donde el programa es ejecutado y de la gestión de la memoria. Los requisitos mínimos para SDT son: PC Pentium Windows 95, 133 Mz, 32 Mg. RAM. Se ha establecido un límite de 50.000 individuos de datos simbólicos. El programa se ha ejecutado

satisfactoriamente con 10.000 individuos de datos simbólicos y 10 variables y con 5000 individuos y 32 variables. Se producen mensajes de error internos cuando existe falta de memoria. En SDT se cargan en memoria los ficheros de datos con la librería SOM. Acerca de los límites y mensajes de error de esta librería debe consultarse Csernel, 1998.

Se han incorporado todos los criterios presentados en esta Memoria, salvo la consideración de peso en los individuos y probabilidades 'a priori' de las clases distintas de las empíricas de la muestra diseño (véase 4.7.2 y 4.7.3). El algoritmo se encuentra descrito en 3.3 y los criterios adoptados en 4.1 y 4.2. También se consideran los criterios de parada de 4.6.2. La predicción se ha implementado según 4.5.2 y 4.5.3. No se ha implementado la predicción de observaciones incompletas con predictores monoevaluados. La predicción de observaciones incompletas con predictores modales probabilistas se ha implementado aportando un nivel de relación del individuo con el árbol en las predicciones (véase 4.5.1).

El programa se ha desarrollado para predictores y variable clase binarios y permite la mezcla de predictores monoevaluados y modales probabilistas. A continuación se especifican los criterios adoptados para valores no observados y no aplicables[6] de un fichero SODAS.

- Tratamiento de *valores no observados*:
  - Individuos con valores no observados en la variable estrato son eliminados del análisis.
  - Individuos con valores no observados en la variable clase constituyen la muestra de predicción.
  - Individuos con valores no observados en un predictor, su permanencia en el análisis o no depende del parámetro de entrada *máximo porcentaje de valores no observados.*

[6] En general, son derivados de una regla de no aplicabilidad, pero el analizador sintáctico de SOM permite valores no aplicables sin reglas de no aplicabilidad asociadas.

- Tratamiento de *valores no aplicables*:
  - Individuos con valores no aplicables en la variable estrato o la variable clase son eliminados del análisis.
  - Individuos con valores no aplicables en un predictor y sin regla de no-aplicabilidad asociada, son tratados como si tuvieran valor desconocido.
  - Individuos con valores no aplicables en un predictor y con regla de no-aplicabilidad asociada. Se consideran tres situaciones:
    * El antecedente de la regla es un predictor. Afecta a la admisibilidad del predictor consecuente relacionado con el predictor antecedente. Véase punto (6) de 4.1.2.
    * El antecedente es la variable clase. Se produce un mensaje informativo y se elimina el predictor del análisis.
    * El antecedente es la variable estrato. Se produce un mensaje informativo y se elimina el predictor del análisis. Se sugiere al usuario que elimine dicho estrato del análisis si desea analizar el predictor.

La idea básica de la implentación orientada a objetos consiste en considerar el árbol como un conjunto de nodos y una serie de relaciones entre los nodos. Cada nodo es un objeto que *conoce* su padre y sus hijos mediante punteros a otros objetos nodo. La construcción del árbol se reduce a añadir nodos hijos a los ya existentes. En los apéndices A y B se encuentran los diseños de los programas SDT y SDTEDITOR, respectivamente.

Detalles de la implementación pueden consultarse en la documentación del *software* elaborada: el plan de desarrollo del *software* en Bravo, 1999e, la especificación de requerimientos en Bravo, 1999a, las interfaces con otros módulos de SODAS en Bravo, 1999b, el diseño del *software* los apéndices A y B, la descripción de la evaluación del *software* en Bravo, 1999c, los resultados de la evaluación del

software en Bravo, 1999d y la descripción de la versión implementada en SODAS 1.04 en Bravo, 1999f. El manual de usuario se encuentra en Bravo, 1999g.

5.4 Salida

5.4.1 Fichero de resultados

Este fichero de resultados contiene:

- *Información de parámetros de entrada y variables e individuos.* Es decir, la lista de parámetros de entrada, variables de análisis, peso y porcentaje de valores desconocidos en los predictores y en la variable estrato, número y lista de individuos no analizados (aquellos con valores no observados en las variables estrato y clase, los no observados en un predictor si el porcentaje de los no observados en ese predictor es menor o igual al valor del parámetro *máximo porcentaje de valores no observados*, aquellos con valor de no aplicabilidad en las variables estrato o clase, aquellos pertenecientes a estratos de peso inicial menor que el valor del parámetro *condición de parada para los estratos*), número y lista de individuos de la muestra de predicción (aquellos con valor desconocido en la variable clase) y la lista de predictores no analizados (por no verificar los requisitos de variables binarias monoevaluadas o modales probabilistas o por tener un porcentaje mayor de no observación del valor de parámetro *máximo porcentaje de valores no observados*).

- *Información de la medida de contenido de información.* La inicial y final, así como el peso inicial de la muestra diseño efectivamente analizada.

- *Información y descripción de los nodos.* Es decir, número de nodos explorados, descripción de los nodos decisionales, terminales y terminal-divide

por los objetos simbólicos que los describen, su peso, su medida de contenido de información, la contribución del nodo a la medida de contenido de información del árbol y el peso relativo que tienen los estratos en su definición. La identificación de los nodos permite conocer el nivel y lugar del mismo en el árbol, si es decisional, terminal-divide o terminal (DEC, TD y TER, respectivamente) y si la clase de probabilidad estimada superior para el nodo es la primera o la segunda (véase (4.163)). El orden de los nodos en la lista es el siguiente: los nodos decisionales, los terminal-divide, los terminales y los nodos no relevantes. Estos nodos son aquellos con un peso menor que *valor de relevancia,* expresado en (5.1).

- *Descripción de los estratos.* Éstos se describen por los nodos decisionales, terminal-divide y terminales que los definen, así como por su peso relativo. Sólo se muestran los nodos con un peso relativo mayor que el valor del parámetro *mínima importancia relativa de un nodo en la descripción de un estrato.*

- *Información acerca de la predicción.* Número de individuos para los que se realiza una estimación de la variable clase, información de las predicciones, de las predicciones para individuos cuyo nivel de relación con el árbol es menor que 1, y de individuos imposibles de asignar una predicción, incluidos aquéllos con valor no observado o no aplicable en la variable estrato (véase 4.5.2 y 4.5.3). Para cada una de las clases predichas: lista de los individuos asignados a la clase con la probabilidad estimada e identificación del nodo decisional o terminal-divide con nivel de relación superior con el individuo. Para las predicciones de nivel de relación con el árbol menor que 1, se proporciona la misma información que en el caso anterior junto con este valor de nivel de relación.

- En el fichero de resultados largo, se incluye además información adicional

del proceso de creación del árbol.

5.4.2 Fichero de diagnósticos

Este fichero contiene la lista de estratos, las reglas de no aplicabilidad presentes en el fichero SODAS, información de tiempo de procesamiento entre cada nivel de profundidad del árbol, información de procesamiento de nodos y mensajes de error e informativos.

Los mensajes de error e informativos se identifican por un código y dan información de la clase y método donde se producen, la causa que los origina y la acción que puede tomar el usuario para evitarlos. Los mensajes de error se dividen en internos y externos. Los mensajes de error internos son aquellos que resultan de una ejecución anormal del programa y se deben al ordenador, el programa o el sistema operativo. Los mensajes de error externos se deben a datos incorrectos, como por ejemplo parámetros incorrectos.

SDT v2.22b reporta 104 mensajes de error internos, 55 externos y 10 informativos. SDTEDITOR v2.22 reporta 11 mensajes de error internos y 9 externos (véase Bravo, 1999g).

Ejemplo 5.1 ***Mensaje de error interno.***

I0016: "Node_SDT_B.Get_Variable"

"Not enought memory free for Weight_Right[]"

Action to be taken: Free PC resources, with special attention to memory resources and execute CSCI SDT again.

Ejemplo 5.2 ***Mensaje de error externo.***

E0013: "SDT.Build"

"Unable to open dump file: Dump_Nodes.dat"

Action to be taken: Check for free disk space and execute CSCI SDT again.

Ejemplo 5.3 ***Mensaje informativo.***

W0008 SODAS_File.Get_Statistics

"Individual %d has a NA value in predictor variable and will be removed. No rule associated with NA values in a predictor"

"Cause: The individual has a NA value in a predictor and no rule is associated to this NA value."

Action to be taken: If desired that this individual is analysed, give a rule of NA values for the predictor, (the predictor in the consequent part). Look at SUM CSCI DB2SO [7] for more details to generate rules associated to NA values.

5.4.3 Fichero gráfico y visualización gráfica del árbol

El fichero gráfico, en ASCII o binario, contiene información acerca del árbol para datos estratificados creado por el programa SDT. El programa SDTEDITOR visualiza este fichero gráfico. Algunas características del editor gráfico SDTEDITOR son:

- La ventana de visualización permite el desplazamientos de izquierda a derecha y de arriba a abajo y sus viceversas.
- Los nodos se representan con distintas formas y colores dependiendo del tipo de nodo y de la clase de mayor probabilidad obtenida: Los círculos representan nodos explorables y los rectangulares, nodos decisionales, terminal-divide o terminales. Se distinguen cuatro colores dependiendo del tipo de nodo: azul para los explorables, verde para los decisionales, naranja para los terminal-divide y gris para los terminales. La intensidad de color indica cual de las dos clases de Z es de mayor probabilidad estimada en los nodos. Las intensidades débiles indican una probabilidad estimada alta para la primera clase de predicción, mientras que las fuertes lo indican para la segunda clase de predicción. Además, los nodos rectangulares de intensidad de color débil se sitúan a la izquierda del nodo explorable del que proceden y los de intensidad de color fuerte, a la derecha.

- En la esquina superior izquierda de la ventana de visualización se muestra información de la medida de contenido de información inicial y final, y de la relación de los colores con la clases estimadas. Véase figura 5.3.

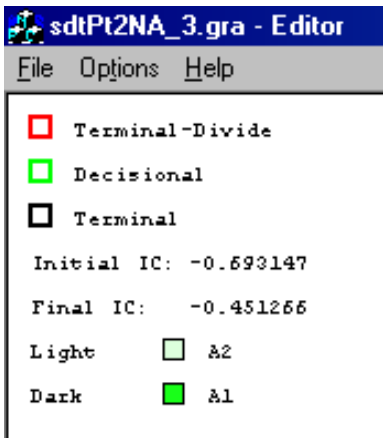


Figura 5.3: SDTEDITOR. Información general del árbol.

- En el árbol, se muestra información de los predictores y categorías que definen las ramas.

- La información que se muestra en los nodos es: para los explorables, el peso del nodo y la probabilidad estimada de la primera clase; y para los demás, el peso del nodo, la probabilidad estimada de la primera clase y los estratos presentes en el mismo.

- Se muestra mayor información de los nodos del árbol en una ventana independiente, pulsando el botón izquierdo del ratón sobre un nodo. Esta información es: el peso del nodo, la probabilidad estimada de la primera clase, los estratos presentes en el nodo, la medida de contenido de información del nodo y la contribución del nodo a la medida de contenido de información del árbol si el nodo es decisional, terminal-divide o terminal, la lista de predictores y categorías que definen el nodo, la lista de estratos

que lo componen con su peso y el peso de la primera clase en ellos, y, cuando sea aplicable, la identificación de los nodos explorables hijos o nodos decisionales, terminal-divide o terminales derivados. Véase figura 5.4.

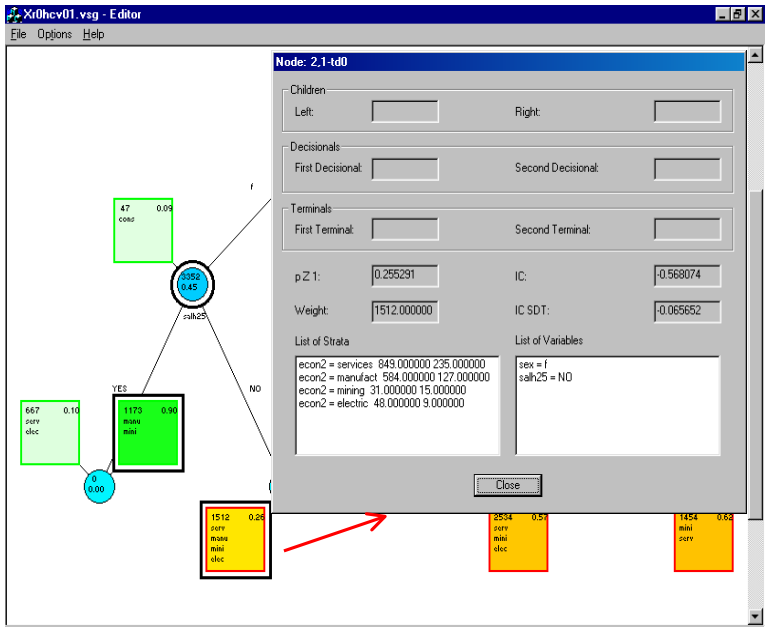


Figura 5.4: SDTEDITOR. Enmarcación de un estrato e infomación de nodos

- Posibilidad de enmarcar los nodos que definen un estrato. Véase figura 5.4.

- Posibilidad de mostrar el árbol completo en una ventana mediante menú o pulsando el botón derecho del ratón (véase figura 5.5). Para volver a la ventana de escala original, se hace mediante menú o pulsando el botón izquierdo del ratón. Si esta acción se realiza mediante el ratón, el lugar que se muestra del árbol tiene como centro el lugar donde se ha pulsado el ratón.

5.5 Adaptaciones posibles

Con la versión actual del *software* SDT 2.22b se pueden realizar algunas adaptaciones a casos más generales. Estas adaptaciones son:

- *Predictores monoevaluados no binarios.* Esta aplicación se puede realizar creando externamente tantas variables cualitativas binarias como particiones binarias sean posibles con las categorías del predictor no binario. Se deben añadir en el módulo DB2SO (véase Hébrail, 1999) las reglas de no aplicabilidad correspondientes.

- *Peso en los individuos.* Se puede conseguir este efecto realizando externamente replicados de las observaciones de forma proporcional a su peso (véase aplicación 4.9.1).

- *Simulación de cortes de una variable de rango continua u ordinal monoevaluada.* Estos cortes se pueden simular, creando variables binarias del tipo *menor que un valor*. Esta variable binaria, vale 1 si la observación es menor que el valor y 0 en caso contrario. Puede verse una aplicación de cortes simulados en los percentiles 25, 50 y 75 en la aplicación 4.9.1.

5.5.1 Implementaciones futuras

En cuanto a las implementaciones futuras del *software*, se encuentran las siguientes:

- Incorporación de una ayuda más extensa con hipertexto e hiperenlaces. La ayuda actual contiene información acerca de los datos de entrada al programa y los parámetros.

- Incorporación de predictores cualitativos no binarios, tanto variables monoevaluadas como variables modales probabilistas

- Incorporación de predictores ordinales.
- Incorporación de predictores de intervalo.
- Incorporación de predictores multievaluados.
- Incorporación de tratamiento de predictores de distintos tipos simultáneamente. En la versión actual, se analizan simultáneamente predictores monoevaluados y modales probabilistas.
- Incorporación en la predicción de datos monoevaluados considerando la falta de observación en algunos predictores.

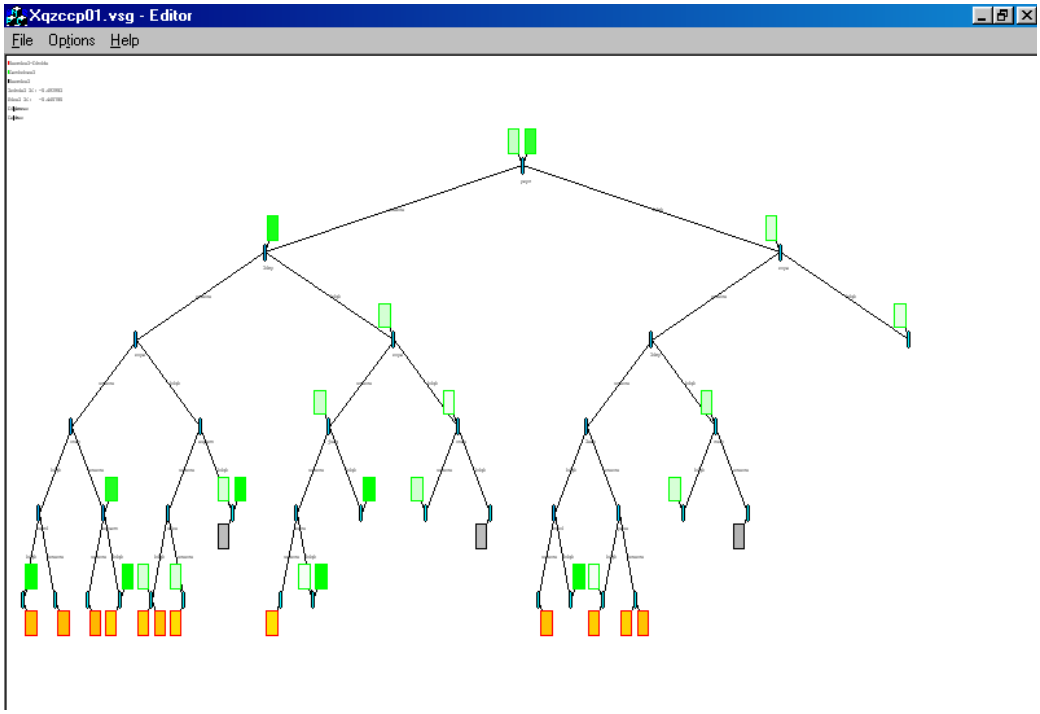


Figura 5.5: SDTEDITOR. Árbol de cinco niveles en una ventana

- Consideración de pesos en las unidades de análisis.
- Consideración de probabilidades 'a priori' de las clases en los estratos.

- Incorporación de la poda.

- Realización de una interfaz gráfica de usuario desde la cual el usuario pueda dirigir interactivamente el proceso de construcción del árbol y acceder a la ayuda. Es decir, incorporar los métodos del programa SDT en esta nueva interfaz gráfica que sustituiría el actual editor gráfico SDTEDITOR.

En esta Memoria, se han propuesto soluciones y medidas concretas para todas ellas, con exclusión del tratamiento de predictores de distintos tipos simultáneamente ya que no se han propuesto las relaciones de dominio correspondientes y un detalle completo de la poda del árbol.

Conclusiones

Conclusiones

Los datos simbólicos aportan una solución a la consolidación datos de un *data warehouse*, más compleja y rica que las que se realizan en la actualidad. Se representan grupos de individuos (con observaciones únicas) mediante intenciones o conceptos formalizados mediante conjunciones de propiedades. La representación mediante datos y objetos simbólicos, reduce el volumen original de los datos, respeta la confidencialidad de los mismos y representa la variación interna del grupo de individuos que se representa mediante intervalos, distribuciones de probabilidad, datos multievaluados, etc... Esta representación del conocimiento engloba además otras semáticas de incertidumbre como las distribuciones de posibilidad, los conjuntos difusos y las funciones de creencia.

Los objetos simbólicos permiten la propagación de conceptos mediante las extensiones de los objetos simbólicos. Además se pueden comparar datos de diversas fuentes y realizar *emparejamiento* estadístico para grupos de individuos con datos procedentes de diversas fuentes.

En esta Memoria se ha propuesto una nueva técnica de Analisis de Datos Simbólicos, el Análisis de Segmentación para datos simbólicos estratificados. El método de Segmentación presentado, junto a otras técnicas de Análisis de Datos Simbólicos, contribuye a la explotación y el análisis de datos consolidados. En concreto, la técnica presentada extrae conocimiento estructurado en forma de objetos simbólicos que son reglas de predicción. La información consolidada y el conocimiento extraido de la misma se expresan por un único formalismo.

Las técnicas de Análisis de Datos Simbólicos son técnicas de *minería de datos simbólicos* y también de *minería de conocimientos.*

Se ha incorporado de forma novedosa la información de los estratos en los algoritmos de Segmentación, alcanzándose el doble objetivo de explicación de una variable clase, de una parte y una clasificación de los estratos por su comportamiento homogéneo en la explicación de la variable clase de otra.

El marco único de representación de los datos, del árbol y de los estratos mediante datos y objetos simbólicos ha permitido identificar los tres grandes propósitos de un algoritmo general de Análisis de Datos Simbólicos: la organización de los datos, la organización del conocimiento y la extracción de conocimiento a partir de datos y / o conocimiento. En la técnica presentada, se adquiere conocimiento estructurado en forma de conceptos expresados mediante reglas de predicción.

Con el método de Segmentación para datos estratificados propuesto se ha descrito por medio de objetos simbólicos cada uno de los estratos, mediante las reglas de predición de las clases que se verifican en los estratos y los pesos que dichas reglas tienen en estos estratos. Los estratos además se han clasificado por reglas de predicción comunes. Se han propuesto medidas y normas de interpretación que permiten interpretar los estratos e idenficar la importancia de las reglas en los mismos y los estratos que caracterizan mayoritariamente una clasificación. Se ha obtenido una descripción simbólica de grupos de individuos, generalizando los mismos por reglas de predicción. Se han incorporado al método metadatos, como son las dependencias jerárquicas entre variables o reglas de no aplicabilidad.

Se ha formalizado el método de forma general y se han propuesto criterios concretos para datos monoevaluados y datos modales probabilistas. Se ha destacado su relevancia frente a los árboles de Segmentación tradicionales.

La formalización de la entrada y salida del método mediante datos y objetos simbólicos representa un único formalismo de fácil interpretación para el usuario. Se ha mostrado cómo esta representación del conocimiento facilita la extensión

del método a otros datos simbólicos, incluidas otras representaciones de incertidumbre. Se han propuesto algunas medidas concretas en estas extensiones.

También se ha incluido un *software* desarrollado que permite la utilización del método propuesto y que facilita la investigación futura para la creación de nuevos criterios que mejoren la técnica desarrollada en esta Memoria y la incorporación de otros tipos de datos simbólicos.

En cuanto a las Oficinas de Estadística se da una solución a los requerimientos de las mismas. Los datos simbólicos permiten reducir el volumen de los datos almacenados y preservan la confidencialidad de los datos. La generalización por datos y objetos simbólicos se puede aplicar a unidades de datos ponderadas. Además, el método propuesto puede analizar directamente las unidades de datos ponderadas, sean éstos simbólicos o no. Generalmente, realizan estratificación en la toma de los datos y también sus bases de datos poseen variables cualitativas de muchas categorías, tales como NACE (categorización de sectores económicos) e IFSCO (categorización de profesiones). Se ha mostrado mediante algunas aplicaciones que las categorías de estas variables pueden representar los estratos. Además, la dispersión de datos nacionales y supranacionales que poseen hacen interesante que las localizaciones geográficas sean los estratos del método propuesto.

Los datos almacenados puede ser datos censales o representar toda una población. El aspecto descriptivo del método propuesto puede ser de gran importancia y único. Además, se puede utilizar en la imputación de variables no observadas o recogidas.

Frecuentemente, sus cuestionarios tienen reglas de no aplicabilidad que pueden ser tratadas por la técnica propuesta. Bastante a menudo, sus bases de datos tienen variables taxonómicas y dependencias lógicas entre variables que pueden ser tratadas por algunas técnicas de Análisis de Datos Simbólicos.

En otros entornos de bases de datos, el aspecto predictivo del método, puede ser de notable interés, además del aspecto descriptivo que favorece la identifi-

cación de las clases para la toma de decisiones. Éste puede ser el campo de la publicidad o el *marketing* donde las técnicas de segmentación se utilizan abundantemente, la predicción de grupos de riesgo, etc...

En cuanto a líneas futuras de investigación, éstas incluyen mejoras o incorporación de nuevos criterios para datos monoevaluados y probabilistas, la incorporación de la poda del árbol, la extensión del método a otros datos simbólicos, incluidas otras medidas de incertidumbre, el estudio de similaridades entre estratos y la predicción en estratos desconocidos. También, la incorporación de las novedades al *software* ya existente en la actualidad.

En cuanto a los criterios se propone el estudio de nuevas medidas de contenido de información extendida que combinen la medida de contenido de información de un nodo con respecto a los estratos con la aportación final a la medida de contenido de información de los nodos obtenidos después de la exploración en una iteración; y, nuevas medidas de contenido de información en el caso de los datos modales probabilistas. Para estos datos, también se propone la incorporación de una medida de combinación de niveles de relación que asigne los individuos a un único nodo del árbol.

En cuanto a los estratos, sería interesante estudiar la obtención de similaridades entre los mismos según su representación de salida del árbol. El cálculo de similaridades entre estratos consideraría no sólo la representación final de los estratos por los nodos decisionales del mismo, sino que incoporaría información de la representación de los mismos en nodos intermedios del árbol.

La validación del método propuesta puede mejorarse con las técnicas de poda por validación cruzada.

La extensión del método a otros datos simbólicos como son las variables de intervalo o multievaluadas como se ha visto es casi directa y sólo necesita su incorporación al *software* al que también se pueden añadir otros tipos de datos con incertidumbre como son los datos posibilistas o difusos, para los que se han propuesto algunas medidas concretas. También el *software* precisa algunas incor-

poraciones que se han detallado en el contenido de esta Memoria.

Una extensión más completa incorporaría la combinación en los datos de entrada de distintas semánticas. Esto sería posible siempre que se definan las relaciones producto correspondientes, la representación de los nodos y el árbol con los datos de entrada. Tambien deberían adaptarse nuevos criterios.

Finalmente, otra línea futura de investigación prometedora es la propagación de las reglas de predicción a otros estratos. En estos casos, se incorporaría información adicional de los estratos exógena a la obtenida a partir de los individuos, por ejemplo en el caso de que los estratos fueran municipios, características de los mismos. A partir de estas variables explicativas se podrían obtener similaridades entre los estratos que permitieran a nuevos estratos ser incorporados a las reglas de predicción, detectando el estrato más próximo.

Apéndices

Apéndice A. Diseño del programa SDT

Introducción

Este apéndice contiene el diseño detallado del *software* SDT v. 2.22b[7] para construir arboles binarios de Segmentación para datos estratificados a partir de un conjunto de individuos, realizando cortes binarios por una de las variables descriptoras. El diseño del *software* se ha realizado utilizando la metodología de Análisis y Diseño Orientado a Objetos OMT. Esta metodología principalmente usa tres tipos de diagramas:

- El modelo de objetos, muestra la estructura estática del sistema como un conjunto de clases interrelacionadas.
- El modelo dinámico, muestra la estructura dinámica del sistema con un diagrama de estados que muestra la secuencia de eventos que se producen en el sistema y como éste responde a ellos, ejecutando acciones en respuesta a eventos.
- El modelo funcional, muestra las transformaciones de datos que se realizan en el sistema. Describe el conjunto de procesos que transforman los datos de entrada en los de salida.

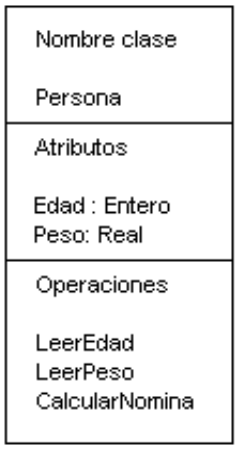
Es de destacar en la elaboración de este diseño la colaboración de Alberto Fernández García, al que se debe la codificación del *software*.

En este caso sólo se utiliza el modelo de objetos y el funcional ya que no existe ninguna clase en el sistema con un conjunto de estados suficientes para justificar la utilización del modelo dinámico.

[7] SDT v 2.22b está integrado en el software SODAS 1.04, resultado del proyecto ESPRIT IV - 20821 SODAS - *Symbolic Official Data Analysis System.*

SODAS 1.04, Copyright CISIA Ceresta, 1999. http://www.cisia.com/

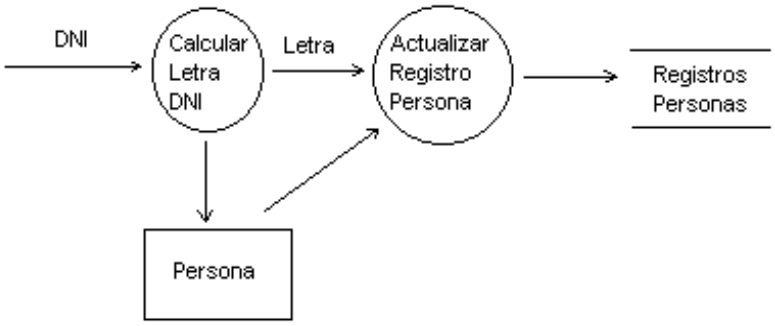
El modelo de objetos contiene las clases más significativas para el diseño del sistema. Estas clases no serán todas las que componen el sistema ya que pueden existir otras clases auxiliares como lista, arrays, etc., que aparecerán posteriormente en la implementación del sistema. Lo que sí contiene son las clases. Este diagrama se compone de una serie de rectángulos divididos entre tres partes horizontalmente, que se corresponden con clases. En la primera se especifica el nombre de la clase, en la segunda los atributos más significativos y en la tercera las operaciones que se pueden realizar sobre los atributos de la clase. Por ejemplo:



También, describe las relaciones existentes entre clases como la herencia, las asociaciones, la agregación, etc.

El modelo funcional describe las transformaciones de datos que se realizan en el sistema. Para ello utiliza una serie de procesos que posteriormente se podrán construir con operaciones de clases. Los procesos actúan como cajas negras, es decir, toman unos datos de entrada, los procesan y producen unos datos de salida. Este diagrama se compone de procesos, flujos de datos, actores (clases) y almacenes. Los procesos se dibujan como círculos con un nombre significativo según la función que realiza. Los flujos de datos son flechas con un nombre significativo que indican la dirección en que se mueven los datos. Los actores que son clases, son los receptores o productores de información, se dibujan como rectángulos con el nombre de la clase dentro. Y los almacenes son cualquier tipo de ente que almacena información, ésta posteriormente se puede leer, se dibujan como dos líneas horizontales con el nombre de los que almacena entre las dos líneas. Los procesos definen operaciones elementales. Cuando un proceso es suficientemente complicado se puede descomponer en otro diagrama de flujo de datos que describe en más detalle el proceso de nivel superior.

El siguiente ejemplo muestra el cálculo de la letra asociado al DNI de una persona y su posterior almacenamiento en un almacén, que podría ser una base de datos.

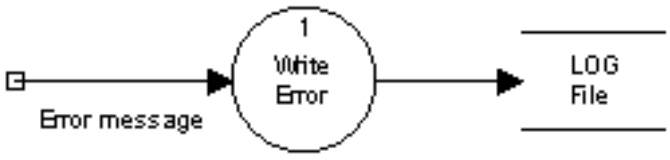


Modelo funcional

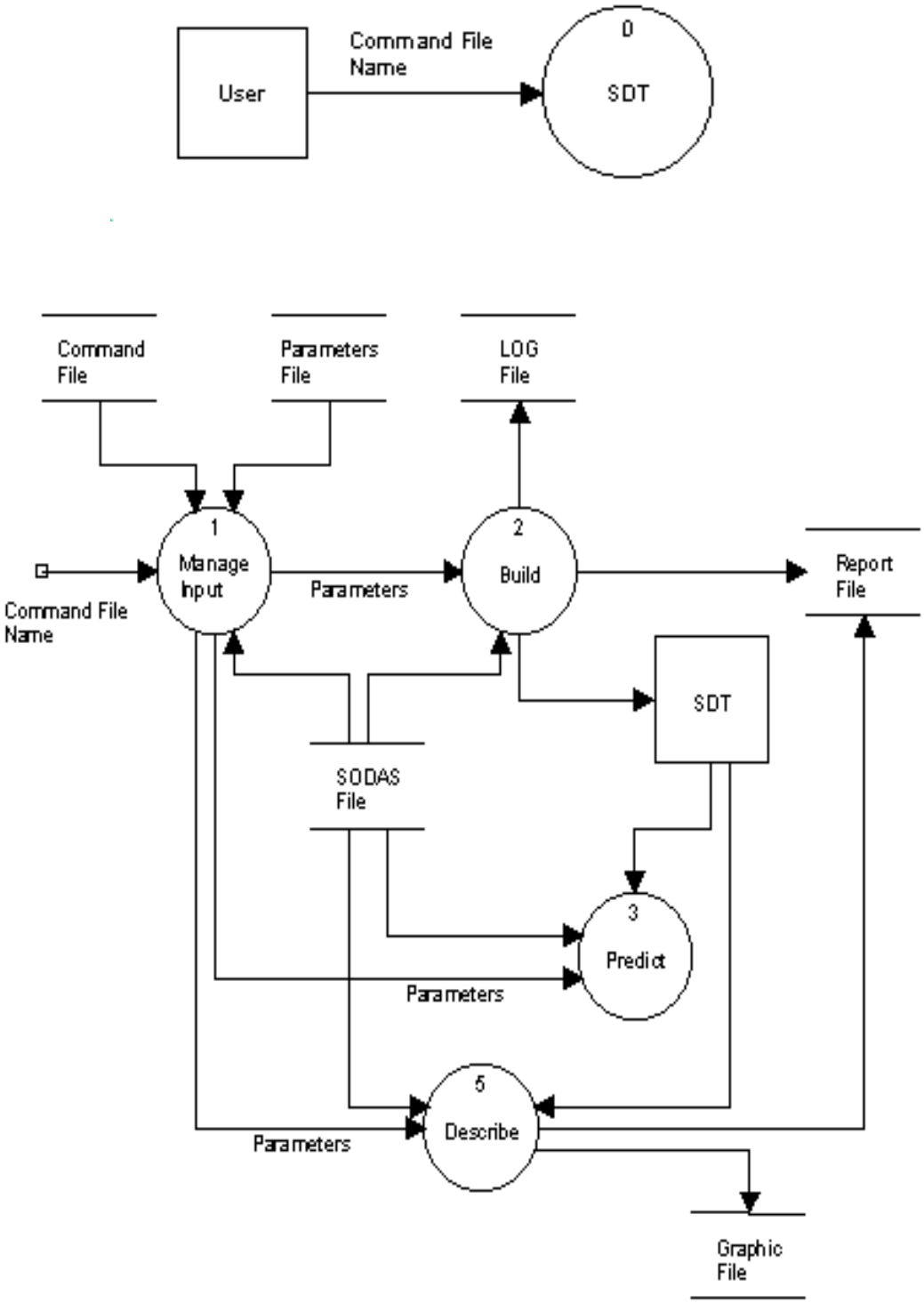
El objetivo del sistema es construir un árbol binario a partir de un fichero de individuos (fichero SODAS) y de unos parámetros de funcionamiento. Una vez construido el árbol, posteriormente se utilizará para predecir nuevos datos. Existen dos casos a tratar: en el primero, caso A, las variables serán del tipo *Classical Nominal Value* y en el segundo, caso B, *Symbolic Nominal Multiple-Modal Value*.

La salida del proceso se almacenará en un fichero gráfico que contiene el árbol para su posterior visualización. También, se escribirá una serie de datos sobre el funcionamiento del sistema en un fichero de informe (Report).

Al principio de la ejecución del sistema se grabará en el fichero del *Log* o fichero de diagnósticos una línea indicando el comienzo del sistema junto con la hora. Cuando acabe la ejecución se grabará en el fichero del *Log* otra línea indicando el final correcto del sistema junto con la hora de finalización. En cualquier momento que se produzca un error se escribirá un mensaje de error en un fichero de *Log* para su posterior análisis.



El modelo de más alto nivel del sistema se muestra en la siguiente figura. Se observa cómo el usuario interactúa con el sistema enviándole el nombre de un fichero de comandos que utilizará el sistema para obtener todos los ficheros y parámetros que necesita el sistema para construir el árbol. El proceso se detalla en la segunda figura.


User
Command File Name
0
SDT
Command File
Parameters File
LOG File
1
Manage Input
Command File Name
Parameters
2
Build
Report File
SDT
SODAS File
3
Predict
Parameters
5
Describe
Parameters
Graphic File

Este diagrama muestra la transformación principal que realiza el sistema. Los principales procesos son:

- **Manage Input**. Este proceso se encarga de la lectura de los parámetros de entrada al sistema y de su verificación. Envía los parámetros a aquellos procesos que los necesitan. (Ver siguientes diagramas).
- **Build.** Proceso principal del sistema, construye el árbol binario a partir de los datos del fichero SODAS y de los parámetros de funcionamiento. (Ver siguientes diagramas).
- **Describe.** Genera todos los ficheros de salida del sistema, para ello utilizará el árbol construido. (Ver siguientes diagramas).
- **Predict**. Utiliza una serie de individuos del fichero SODAS, marcados en una variable (*predictive*), para predecir el valor que tomará dicha variable, su categoría.

A continuación se describen estos procesos.

Proceso Manage Input

Flujos de entrada

Command File name, nombre del fichero de comandos (CMD).

Flujos de salida

Parameters, estructura con los parámetros de funcionamiento.

Proceso

No aplicable.

Process Command File

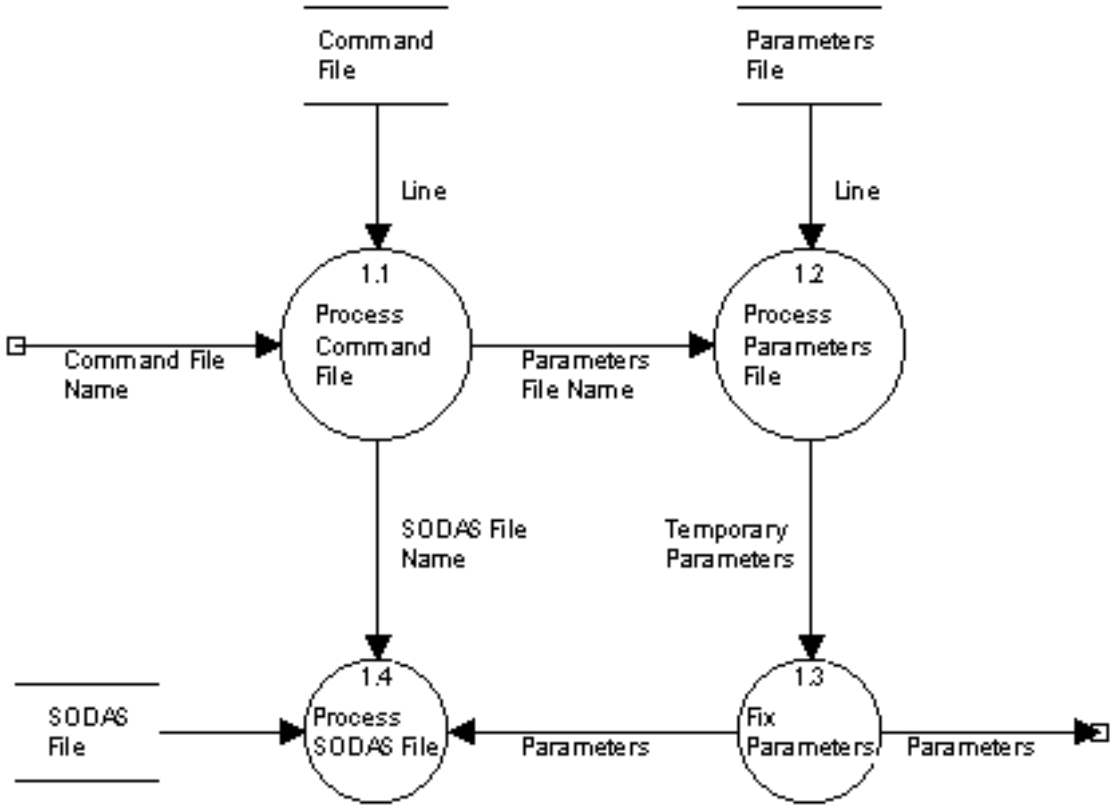
Flujos de entrada

Command File name, nombre del fichero de comandos (CMD). El fichero de comandos es un fichero ASCII con las siguientes líneas:

Fichero de parámetros particulares
Directorio de salida
Nombre de la cadena
Nombre más directorio del fichero SODAS de entrada
Nombre más directorio del fichero de parámetros (PAD)
Versión de SODAS
Fecha

Flujos de salida

Parameters File Name, nombre y directorio del fichero de parámetros.
SODAS File Name, nombre y directorio del fichero SODAS.


Command File
Parameters File
Line
Line
1.1
Process Command File
1.2
Process Parameters File
Command File Name
Parameters File Name
SODAS File Name
Temporary Parameters
1.4
Process SODAS File
1.3
Fix Parameters
SODAS File
Parameters
Parameters

Proceso

Comprueba el nombre del fichero de comandos, puede ser nulo.

Abre el fichero de comandos para ello usa el nombre del fichero de comandos. Lee todas las líneas del fichero y las almacena en la estructura de parámetros del sistema.

Envía al proceso "***Process Parameter File***" el nombre y directorio del fichero de parámetros leído del fichero de comandos.

A continuación envía al proceso "***Process SODAS File***" el nombre y el directorio del fichero SODAS.

Process Parameters File

Flujos de entrada

Parameters File name, nombre y directorio del fichero de parámetros (.PAD). Es un fichero ASCII formado por líneas, cada línea contiene la definición de un parámetro con la sintaxis:

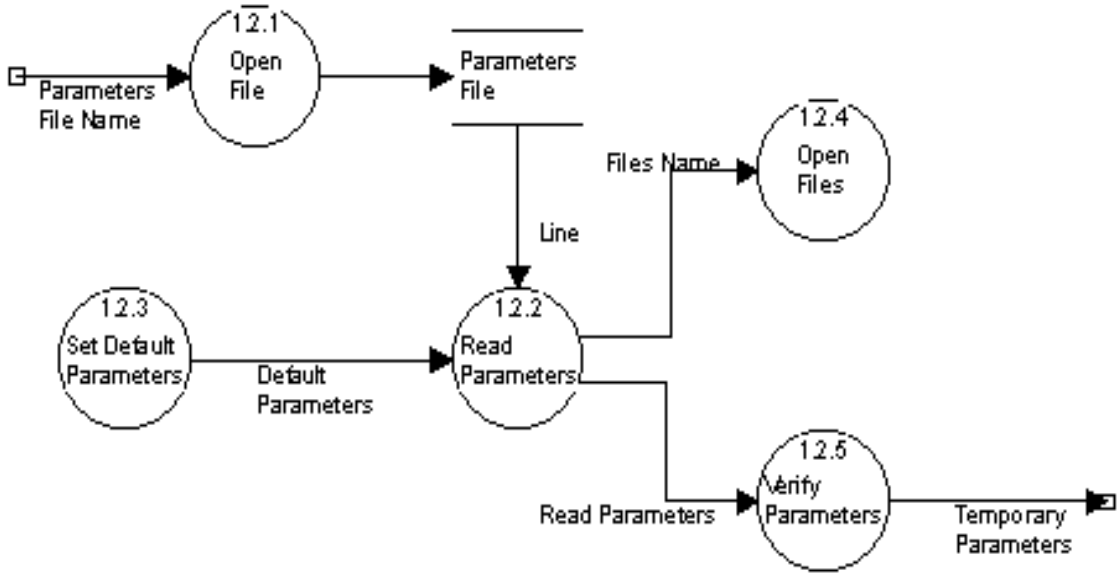
KEYWORD = VALUE

Flujos de salida

Temporary Parameters, estructura temporal con los parámetros.

Proceso

No aplicable.



Open File

Flujos de entrada

Parameters File name, nombre y directorio del fichero de parámetros (.PAD).

Proceso

Comprueba el nombre del fichero de parámetros, si este es nulo escribe un error. Abre el fichero de parámetros.

Set Defaults Parameters

Flujos de salida

Default Parameters, estructura de datos idéntica a la de parámetros pero rellenada con los valores por defecto de los parámetros.

Proceso

Inicializa la estructura de parámetros temporales con los valores por defecto. Los campos que se inicializan son:

p_min (20)
p_max (80)
min_weight (5)
max_number_levels (5)
ic_increment (200)
process_first (TRUE)
strata_minimum (10)
dump (FALSE)
report_type (SHORT)
variable_percentage (0)

Read Parameters

Flujos de entrada

Line, una línea leída del fichero de parámetros, 256 caracteres máximo.

Default Parameters, estructura de datos idéntica a la de parámetros pero rellenada con los valores por defecto de los parámetros.

Flujos de salida

File Names, nombre y directorio de los otros ficheros: fichero del *Log*, del informe y del gráfico.

Read Parameters, estructura con los parámetros leídos del fichero.

Proceso

Lee el fichero de parámetros línea a línea hasta que encuentre el fin del fichero de parámetros. Cada línea la separa en dos partes, la parte izquierda y la parte derecha. El criterio de separación es el signo igual "=", lo que este a su izquierda y lo que este a su derecha.

La parte izquierda se compara con una serie de cadenas de caracteres predifinidas que son idénticas al nombre de los parámetros en el fichero. La parte

derecha es el valor del parámetro, en función del nombre del parámetro se almacena el valor en el campo de la estructura de parámetros correspondiente. Se efectúan las conversiones de tipo necesarias. También se realiza el procesamiento de los símbolos especiales como "—" y ">".

Open Files

Flujos de entrada

File Names, nombre y directorio de los otros ficheros: fichero del *Log*, del informe y del gráfico.

Proceso

Recibe como parámetro de entrada los nombres de los tres ficheros de salida del sistema y los abre uno a uno. Para cada uno de ellos comprueba que el nombre no sea nulo.

Verify Parameters

Flujos de entrada

Read Parameters, estructura con los parámetros leídos del fichero.

Flujos de salida

Temporary Parameters, parámetros leídos del fichero ya verificados.

Proceso

Verifica los parámetros que acaba de leer del fichero. Comprueba que cada parámetro esta dentro de su rango valido de valores, si no es así, se escribe un mensaje de error y se para el proceso.

Fix Parameters

Flujos de entrada

Temporary Parameters, parámetros leídos del fichero ya verificados.

Flujos de salida

Parameters, estructura con los parámetros de funcionamiento.

Proceso

Divide los parámetros p_min y p_max por cien (100). También, divide el parámetro IC increment por diez mil (10000).

Process SODAS File

Flujos de entrada

SODAS File Name, nombre y directorio del fichero SODAS.

Parameters, estructura con los parámetros de funcionamiento.

Proceso

Primero se abre el fichero SODAS en modo lectura. Se crea un objeto del tipo CompSymbMat, de la librería SOM, para gestionar el fichero SODAS. A la vez

que se crea el objeto se verifica sintácticamente y semánticamente el fichero, esto lo hace la librería SOM. Después, se comprueba si hay algún error en el fichero.

Se inicializan algunos arrays internos para la gestión de valores nulos.

A continuación, se realizan una serie de verificaciones con el fichero SODAS. Se comprueba:

Que el número de individuos no supere un valor determinado (50000).

Se comprueba que el índice de la variable *predictive* esté entre 0 y el número de variables del fichero SODAS.

Idem para la variable estrato.

Idem para el número de *predictors*.

Idem para cada *predictor*.

Se comprueba que el número de categorías de cada *predictor* no sea mayor que dos (2). Si es así, se eliminará el *predictor* de los parámetros.

Se comprueba que el tipo de la variable *predictive* sea Nominal. También que el número de categorías de la variable sea menor o igual que dos (2).

Se comprueba que el tipo de la variable estrato sea Nominal.

Se averigua el tipo de algoritmo que se utilizará, caso A o caso B. Para ello, se comprueba el tipo de las variables *predictors* y se coteja entre ellas.

Para cada *predictor* se comprueba el porcentaje de nulos que tiene la variable, si alguna sobrepasa el límite se eliminará.

Si, al final, no queda ningún *predictor* para utilizar se escribirá un mensaje de error y se abortará el proceso.

Proceso Build

Flujos de entrada

Parameters, estructura con los parámetros de funcionamiento.

Proceso

No aplicable.

Get Statistics

Flujos de entrada

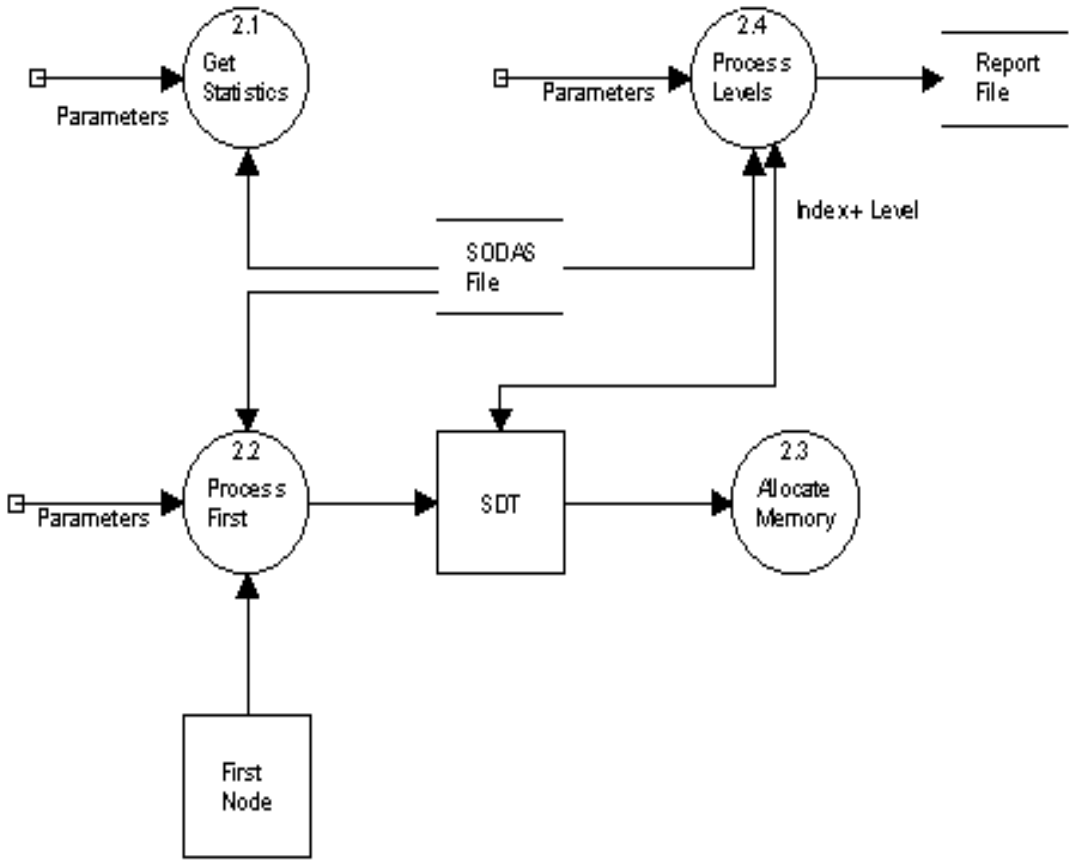
Parameters, estructura con los parámetros de funcionamiento.

Proceso

Reserva memoria para almacenar las estadísticas que va a realizar e inicializa los valores de las estadísticas a ceros.

Comprueba si el fichero SODAS incluye alguna regla, ya que si no la incluye y aparece algún valor No Aplicable (NA), entonces es un error.

A continuación procesa todos los individuos del fichero SODAS. Para cada individuo lee en orden los valores de *predictive*, *strata* y *predictors*.


2.1
Get
Statistics
Parameters
2.4
Process
Levels
Parameters
Report
File
Index+ Level
SODAS
File
2.2
Process
First
Parameters
SDT
2.3
Allocate
Memory
First
Node

Predictive. Si tiene un número de nulos mayor que cero (0) se activa el *flag* o indicador de Z nula para ese individuo y se suma uno (1) al contador de individuos con Z nula.

Si tiene un número de No Aplicables (NA) mayor que cero (0) se activa el *indicador* de alguna variable nula para ese individuo y se suma uno (1) al contador de individuos con Z nula.

Strata. Si el número de nulos es mayor que cero (0) se activa el *indicador* de alguna variable nula para ese individuo y se suma uno (1) al contador de estratos nulos.

Si tiene un número de No Aplicables (NA) mayor que cero (0) se activa el *indicador* de alguna variable nula para ese individuo y se suma uno (1) al contador de estratos nulos.

Predictors. Para cada *predictor*:

Si el número de nulos es mayor que cero (0) se activa el *indicador* de alguna variable nula para ese individuo y se suma uno (1) al contador de nulos para ese variable.

Si el número de No Aplicables (NA) es mayor que cero (0) y el *indicador* de que no hay reglas esta activado, entonces se activa el *indicador* de alguna variable nula para ese individuo y se suma uno (1) al contador de nulos para ese variable.

Al final del procesamiento del individuo, si no esta activa ningún *indicador* de nulo se sumará uno al contador del peso del estrato. En caso contrario se sumará uno al contador de nulos. Si el *indicador* de alguna variable esta activado entonces se sumará uno al contador de individuos nulos.

Después de procesar todos los individuos se calculará el peso inicial del fichero SODAS como la diferencia entre el número de individuos total menos el número de nulos.

Process First

Flujos de entrada

Parameters, estructura con los parámetros de funcionamiento.

Proceso

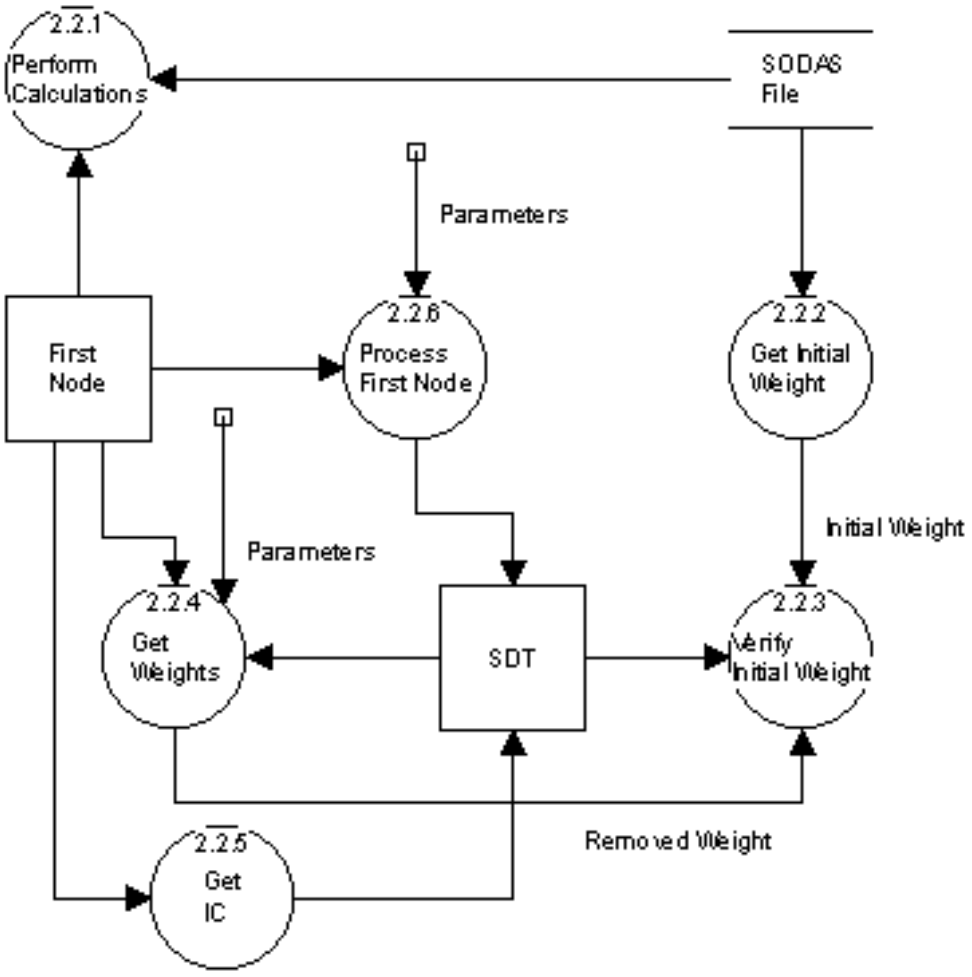
No aplicable.

Perform Calculations

Proceso

Calcula los pesos en el primer nodo. Los pesos que se calculan son el peso total del nodo, el peso de cada estrato del nodo y el peso de Z 1 de cada estrato del nodo.

Calcula los valores del nodo. Estos valores son: p Z 1, IC e IC_SDT. Para calcular los valores utilizará los pesos anteriormente calculados.


2.2.1
Perform Calculations
SODAS File
Parameters
First Node
2.2.6
Process First Node
2.2.2
Get Initial Weight
Initial Weight
Parameters
2.2.4
Get Weights
SDT
2.2.3
Verify Initial Weight
Removed Weight
2.2.5
Get IC

Get Initial Weight

Flujos de salida

Initial Weight, peso inicial del fichero SODAS.

Proceso

Lee el peso inicial del fichero SODAS. Dicho peso se obtiene al calcular las estadísticas del fichero SODAS.

Verify Initial Weight

Flujos de entrada

Initial Weight, peso inicial del fichero SODAS.

Removed Weight, peso eliminado al eliminar estratos.

Proceso

Resta al peso inicial el peso de los estratos que han sido eliminados. Después, comprueba si el peso inicial es menor que cero (0) si es así, escribirá un mensaje de error en el fichero del *Log*.

Get Weights

Flujos de entrada

Parameters, estructura con los parámetros de funcionamiento.

Flujos de salida

Removed Weight, peso eliminado al eliminar estratos.

Proceso

Para cada categoría de la variable *strata*, lee el peso de esa categoría en el primer nodo. Si el peso del estrato es menor que el valor del parámetro de entrada se elimina la categoría, o estrato, del primer nodo, añadiendo el peso del estrato a la suma de peso eliminado (parámetro de salida).

Get IC

Proceso

Lee el valor del IC del primer nodo y lo almacena en el árbol como el primer IC (Initial_IC).

Process First Node

Flujos de entrada

Parameters, estructura con los parámetros de funcionamiento.

Proceso

Si el parámetro de procesar el primer nodo está activado entonces se procesará el primer nodo. Dicho procesamiento incluye:

Obtener los nodos decisionales del primer nodo.

Obtener los nodos terminales del primer nodo.
Calcular los valores del primer nodo (p Z 1, IC e IC_SDT).
Calcular el IC del árbol.

Allocate Memory

Proceso

Primero se reserva memoria en el primer nodo para las variables *predictors*, las variables que van a ser No Aplicables (NA) y la lista de variables ordenadas. Después inicializa la memoria.

Luego, se reserva memoria en el primer nodo para todas las categorías de la variable *strata*, para el peso de cada estrato, para el peso de Z 1 de cada estrato. Después inicializa la memoria.

A continuación se asignan todas las categorías de la variable *strata* al primer nodo.

En función del número máximo de niveles del árbol, se reserva dentro del árbol espacio suficiente para albergar el número máximo de nodos. Y se inicializa el árbol a nulos (0).

Se establece la raíz del árbol con el primer nodo y se hace éste explorable.

Process Levels

Flujos de entrada

Parameters, estructura con los parámetros de funcionamiento.

Proceso

No aplicable.

Get Next Node

Flujos de entrada

Level, índice del nivel que se está procesando.

Proceso

Obtiene el siguiente nodo del nivel que se está procesando no haya sido procesado todavía.

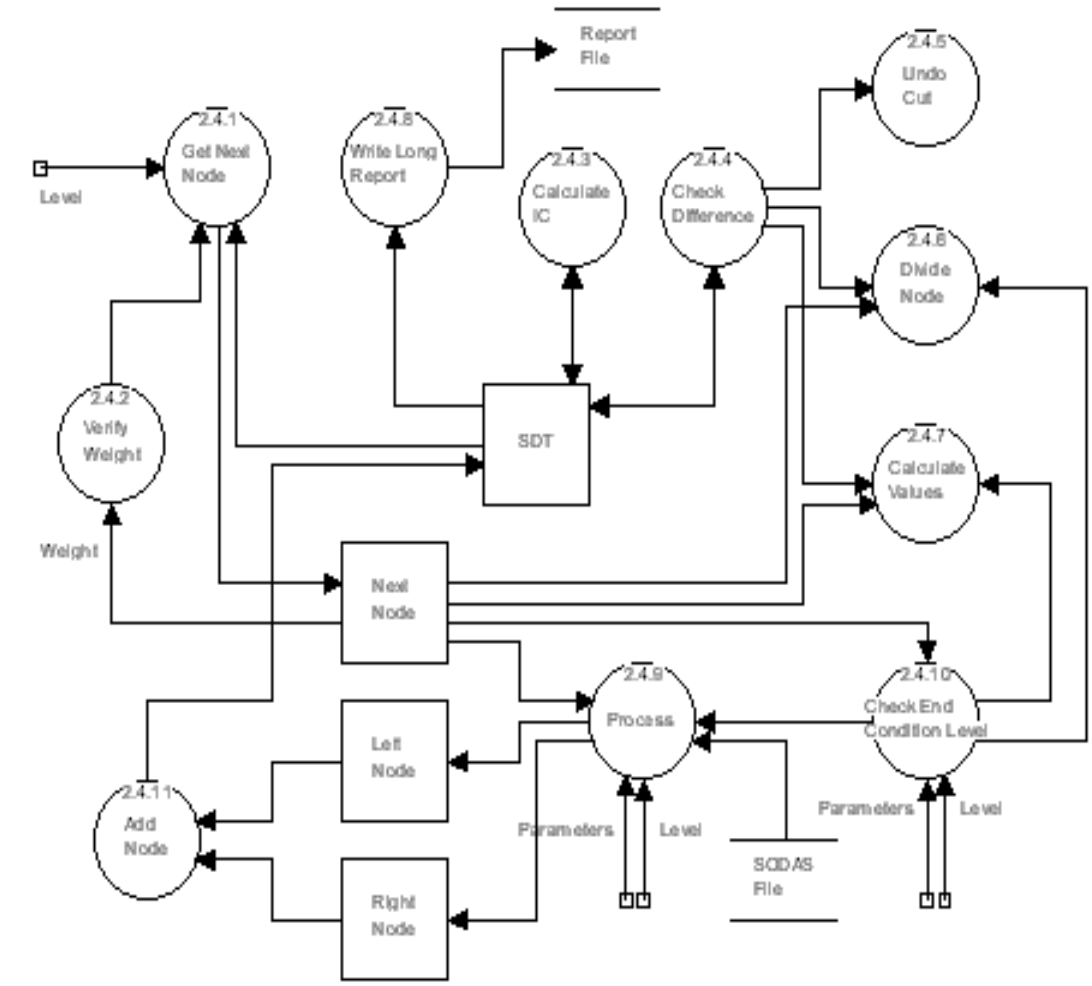
Verify Weight

Flujos de entrada

Weight, peso del nodo actual.

Proceso

Lee el peso del nodo actual, leyéndolo del objeto Next Node. Si el peso del nodo es menor que un valor mínimo (0.000001) entonces, se hace este nodo no explorable, se activa el *indicador* que indica que el nodo ya está procesado y se



llama al proceso "***Get_Next_Node***" para obtener el siguiente nodo de este nivel a procesar.

Calculate IC

Proceso

Se guarda el valor del IC en una variable auxiliar, Last_IC.

Calcula el valor del IC del árbol que se está construyendo.

El IC se calcula como la suma de todas las contribuciones IC_SDT de todos los nodos que son explorables, los decisionales y los terminales.

Además, a la vez que se calcula el valor del IC se comprueba que la suma de los pesos de todos los nodos involucrados en el cálculo del IC es igual a peso inicial del árbol. Si no es así, se escribirá un mensaje de error.

Check Diference

Proceso

Comprueba que la variación del IC es menor que un valor de referencia. La variación del IC se calcula con la formula:

AIC = abs(IC – Last_IC) / abs(Last_IC)

El valor de referencia utilizado es el parámetro de entrada IC_Increment.

Si la variación del IC es mayor que el valor de referencia entonces se deshará el último corte realizado llamando al proceso ***Undo_ Cut***, se dividirá el nodo en dos nodos terminales llamando al proceso "***Divide_ Node***", se recalcularán los valores del nodo llamando al proceso "***Calculate_ Values***" y se volverá a calcular el IC del árbol llamando al proceso "***Calculate_ IC***".

Undo Cut

Proceso

Lo primero se rompe el enlace con los dos posibles nodos hijos, a su izquierda y derecha. Y se deshace todos los cortes y nodos creados en cada uno de los dos posibles hijos creados, izquierda y derecha. Es decir, se limpia todo lo que se hubiera hecho en los dos hijos del nodo actual. Se borra la variable de corte y se destruyen los nodos terminales, decisionales si los tuviera.

Borra los nodos del árbol. También borra la variable de corte en el nodo actual.

Para cada nodo terminal del nodo actual, se vuelven a transferir los estratos al nodo actual y se quita el enlace con el nodo terminal.

Divide Node

Proceso

Divide el nodo actual en dos nuevos nodos terminales asignando los estratos a uno u otro nuevo nodo terminal en función de su valor de p Z 1 del estrato.

Para cada estrato del nodo calcula el valor de p Z 1 del estrato, si éste es menor que 0.5 asigna el estrato al primer nodo terminal, eliminando el estrato del nodo. Si es mayor que 0.5 asigna el estrato al segundo nodo terminal, eliminando el estrato del nodo.

Una vez hecho esto, vuelve a procesar todos los estratos para procesar los estratos con un valor de p Z 1 cercano a 0.5. Para cada estrato vuelve a calcular su p Z 1 y comprueba:

Si alguno de los nodos terminales no está creado se asigna el estrato al que esté creado.

Si no hay ninguno creado, crea el primero y se lo asigna a éste.

Si están los dos creados se lo asignará al que su valor de p Z 1, del nodo, este más cerca al valor de él del estrato.

Al final, para cada nuevo nodo terminal creado, se le hace no explorable, se le hace procesado, se le pone el tipo DECISIONAL, se guardan los valores de las variables en el nodo, se calculan sus valores llamando al proceso "***Calculate_Values***" y se activa el *indicador* de división en el nuevo nodo, "***Divide_Flag***".

Calculate Values

Proceso

Calcula los valores del nodo. Estos valores son: p Z 1, IC e IC_SDT. Para calcular los valores utilizará los pesos del nodo.

Write Long Report

Proceso

Si el tipo de informe que se va a escribir es de tipo largo, LONG, se escribe en el fichero del Report información sobre el proceso que se acaba de realizar en el nodo. Dicha información incluye:

Identificación del nodo

Variables que definen el nodo.

Información del nodo izquierdo. Esta incluye: Identificación del nodo, variables que lo definen e información de los nodos decisionales y terminales. La información de cada uno de ellos incluye:

Identificación

Variables que lo definen

Estratos que contiene

Valor de Z

Peso del nodo

Peso de los estratos

Información del nodo derecho. Idem al izquierdo.

Si el nodo fue divido en dos terminales, se escribe información de cada nodo terminal. La información que se escribe es la misma que el hijo izquierdo o derecho.

Valor de IC y variación del IC

Process

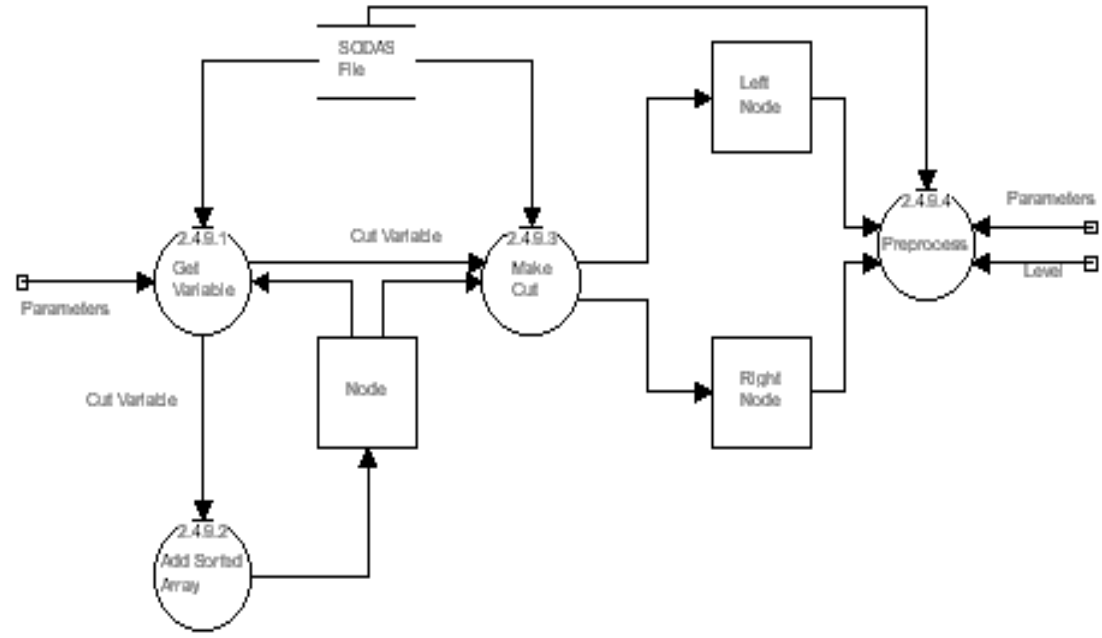
Proceso

No Aplicable

Get Variable

Flujos de entrada

Parameters, estructura con los parámetros de funcionamiento.



Flujos de salida

Cut Variable, índice de la variable para realizar el siguiente corte.

Proceso

Obtiene la siguiente variable de corte.

Lo primero que hace es leer el número de individuos del fichero SODAS. A continuación procesa todas las variables del nodo. Para cada variable:

Si la variable está marcada como no aplicable entonces la ignora, continua con la siguiente.

Comprueba si la variable es aplicable.

Si la variable no ha sido utilizada y es aplicable entonces la utiliza sino la ignora y continua con la siguiente variable.

Si una variable se puede chequear, se procesan todos los individuos del fichero SODAS obteniendo la información del posible nodo izquierdo y derecho (peso del estrato y peso Z 1 del estrato). Después se calcula el EIC y en función de Éste se obtiene la variable de corte. Si es el valor máximo entonces la variable para la que el EIC sea máximo será la variable de corte.

Add Sorted Array

Flujos de entrada

Cut Variable, índice de la variable de corte.

Proceso

Añade la variable de corte al array interno de variables ordenadas. Suma uno al contador de variables ordenadas.

Make Cut

Flujos de entrada

Cut Variable, índice de la variable de corte.

Proceso

Realiza el corte del nodo según la variable de corte. Para ello procesará todos los individuos del fichero SODAS. Para cada individuo leerá sus datos del fichero SODAS (*strata*, *predictive*, *predictors*), si es un individuo nulo continuará con el siguiente individuo, comprobará si pertenece al nodo, y a continuación si el valor la variable de corte es igual a 1 se asignará el estrato al nodo de la izquierda y si vale 2 se asignará al nodo de la derecha.

Por último, para cada posible nodo, izquierdo o derecho, se le hará explorable, se le hará como no procesado, se establecerá su tipo como NORMAL, se guardará el valor de las variables en el nodo y se almacenará la variable de corte.

Preprocess

Proceso

No Aplicable

Perform Calculations

Proceso

Ver apartado **Perform Calculations** en página 316.

Get Decisionals

Flujos de entrada

Parameters, estructura con los parámetros de funcionamiento.

Proceso

Procesa todos los estratos del nodo comprobando si el valor de p Z 1 de cada uno es menor o mayor que los parámetros de entrada. Para cada estrato del nodo calcula el valor de p Z 1 del estrato.

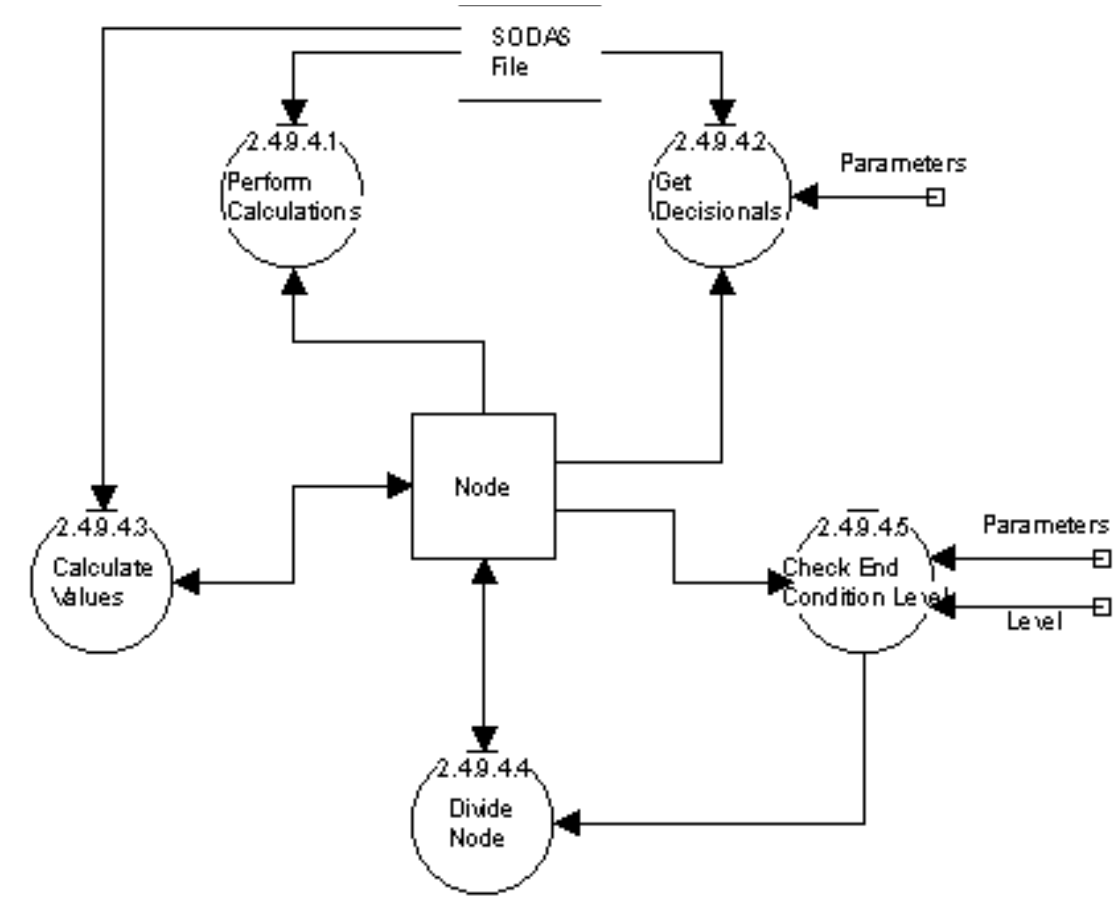
Si el valor de p Z 1 del estrato es menor que el parámetro p min, se asigna el estrato al primer nodo decisional y se elimina del nodo el estrato.

Si el valor de p Z 1 del estrato es mayor que el parámetro p max, se asigna el estrato al segundo nodo decisional y se elimina del nodo el estrato.

Al final de procesar los estratos, para cada nuevo nodo decisional creado se le hace no explorable, se le hace ya procesado, se establece su tipo a DECISIONAL y se recalculan sus valores llamando al proceso "***Calculate_ Values***".

Calculate Values

Proceso


SODAS
File
2.4.9.4.1
Perform
Calculations
2.4.9.4.2
Get
Decisionals
Parameters
Node
2.4.9.4.3
Calculate
Values
2.4.9.4.5
Check End
Condition Level
Parameters
Level
2.4.9.4.4
Divide
Node

Ver apartado **Calculate Values** en la página 322.

Divide Node

Proceso

Ver apartado **Divide Node** en la página 321.

Proceso Check End Condition Level

Proceso

Ver siguiente apartado **Check End Condition Level.**

Check End Condition Level

Flujos de entrada

Parameters, estructura con los parámetros de funcionamiento.

Level, índice del nivel que se esta procesando.

Proceso

Comprueba si se ha llegado al último nivel o que ya no hay ninguna variable sin procesar en el nodo, es decir, todas las variables han sido utilizadas para realizar cortes en el nodo.

Para comprobar el nivel se utiliza el parámetro de entrada, número máximo de niveles a procesar.

Si se cumple alguna de las condiciones anteriores para el nodo actual entonces se llama al proceso "***Divide_ Node***" y se recalculan los valores del nodo llamando al proceso "***Calculate_ Values***".

Add Node

Proceso

Añade un nuevo nodo al árbol en el nivel indicado y colgando del padre indicado.

Proceso Predict

Flujos de entrada

Parameters, estructura con los parámetros de funcionamiento.

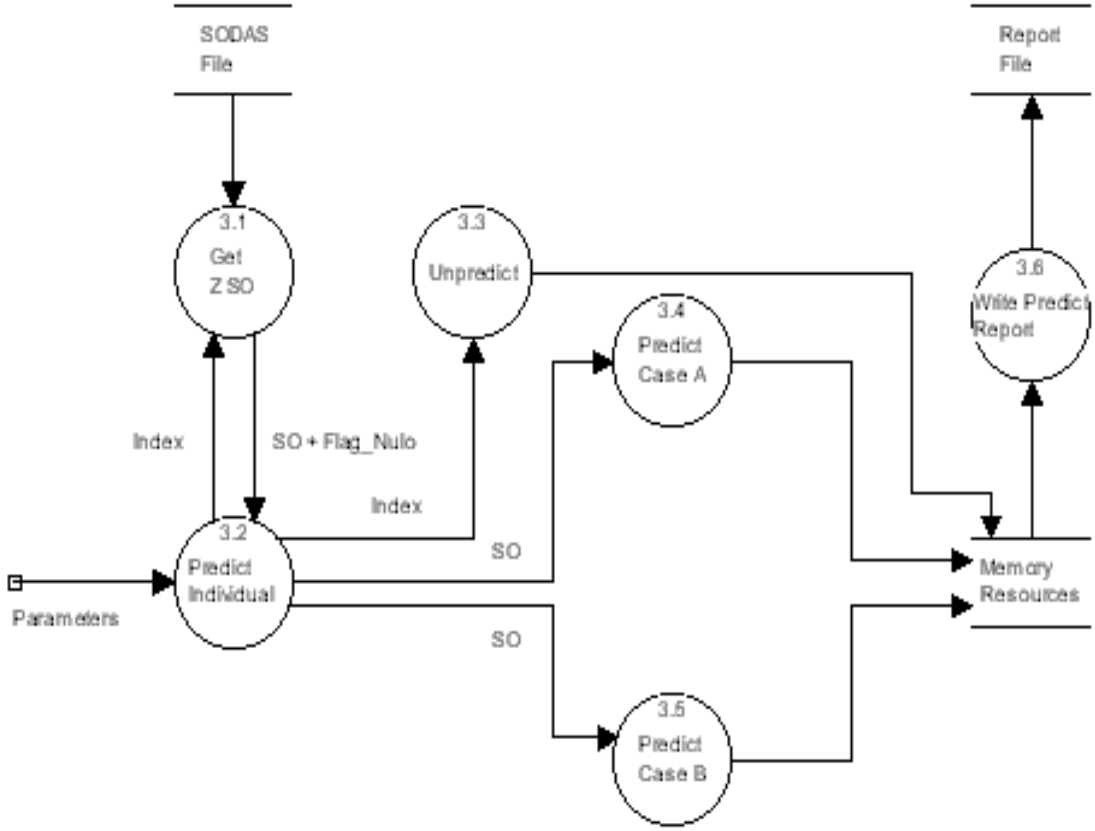
Proceso

No aplicable.

Get Z SO

Flujos de entrada

Index, índice del individuo que se quiere leer.


SODAS
File
Report
File
3.1
Get
Z SO
3.3
Unpredict
3.6
Write Predict
Report
3.4
Predict
Case A
Index
SO + Flag_Nulo
Index
SO
3.2
Predict
Individual
Parameters
Memory
Resources
SO
3.5
Predict
Case B

Flujos de salida

SO, estructura que contendrá los datos del individuo leído.

Flag, indica si el individuo tiene algún valor nulo en sus datos.

Proceso

Primero comprobará que el índice es correcto, es decir, esté entre 0 y el número de individuos del fichero SODAS. Después comprobará si el individuo tiene el valor de la variable *predictive* nulo consultándolo en un array interno.

Por último, limpia la estructura de datos del individuo, lee todos los valores del individuo del fichero SODAS y los almacena en la estructura de salida.

Si el individuo tiene un valor nulo en la variable *strata* se activará el *indicador* de nulo.

Predict Individual

Flujos de entrada

Parameters, estructura con los parámetros de funcionamiento.

SO, estructura que contendrá los datos del individuo leído.

indicador, indica si el individuo tiene algún valor nulo en sus datos.

Flujos de salida

Index, índice del individuo que se quiere leer.

Proceso

Reserva memoria para las estructuras de datos que almacenarán los datos que se escribirán el informe de predicción.

Se procesan todos los individuos del fichero SODAS. Para cada individuo se comprueba si el valor de la variable *predictive* es nulo, si no lo es continua con el siguiente individuo.

Llama al proceso ***Get_ Z_ SO*** con el índice del individuo actual para leer sus valores. Si el *indicador* de nulo está activado se llamará el proceso "***Unpredict***" con el índice del individuo.

Si se está procesando un algoritmo de tipo A se llamará al proceso "***Predict_ Case_ A***" y sino se llamará al proceso "***Predict_ Case_ B***". En ambos casos, se enviará el individuo actual.

Unpredict

Flujos de entrada

Index, índice del individuo que se quiere hace impredecible.

Proceso

Almacena el individuo en el array de impredecibles y suma uno al contador de individuos impredecibles.

Predict Case A

Flujos de entrada

SO, estructura que contendrá los datos del individuo a predecir.

Proceso

Se busca el nodo al que pertenece el individuo. Para ellos se recorre el árbol de nodos, SDT, buscando el nodo decisional o terminal al que pertenece el individuo. Si lo encontramos se obtiene la identificación del nodo y el valor de p Z 1 del nodo. Además se activa el *indicador* de encontrado (*indicador* interno). Cuando lo encontremos en el árbol, se para la búsqueda de nodos.

Después de recorrer todo el árbol si no se ha encontrado un nodo se hace el individuo impredecible. Se almacena el índice del individuo en el array de impredecibles y se suma uno al contador de impredecibles.

Si encontramos el nodo, se suma uno al contador de nodos predichos. Si el valor de p Z 1 del nodo es mayor de 0.5, se almacenan los datos obtenidos (nodo, p Z 1) en el array de datos de individuos de clase 1. Si el valor de p Z 1 es menor o igual que 0.5 se almacenan los datos (nodo, 1 – p Z 1) en el array de individuos de clase 2.

Predict Case B

Flujos de entrada

SO, estructura que contendrá los datos del individuo a predecir.

Proceso

Se buscan todos los nodos del árbol con un nivel de relación no nulo con el individuo. Para ello se recorre todo el árbol buscando los nodos decisionales o terminales con nivel de relación no nulo con el individuo. Cuando se encuentra un nodo, se activa el *indicador* de encontrado (*indicador* interno), se seleccion el nodo de mayor probabilidad, se lee el valor de p Z 1 del nodo y se actualiza el nivel de relación con el árbol (valor de *fiable*).

Después de recorrer todo el árbol si no se ha encontrado un nodo se hace el individuo impredecible. Se almacena el índice del individuo en el array de impredecibles y se suma uno al contador de impredecibles.

Si encuentra algún nodo, lo primero que hace es comprobar si el valor de *fiable* es menor que 0, si lo es, se hace el individuo impredecible.

Si el valor de *fiable* es mayor que 0 entonces se suma 1 al contador de individuos predichos. A continuación se comprueba si el valor de *fiable* es igual a 1. Si no lo es, se comprueba el valor de la probabilidad si es mayor que 0.5 se almacena los datos del individuo (nodo, valor de probabilidad, *fiable*) en el array de datos de individuos *no fiables* (de nivel de relación con el árbol menor que 1) de clase 1, si no es mayor que 0.5 se almacena los datos (nodo, 1 – valor de probabilidad, *fiable*) en el array de datos de individuos *no fiables* de clase 2.

Si el valor de *fiable* es igual a 1, se comprueba el valor de la probabilidad si es mayor que 0.5 se almacena los datos del individuo (nodo, valor de probabilidad) en el array de datos de individuos de clase 1, si no es mayor que 0.5 se almacena

los datos (nodo, 1 – valor de probabilidad) en el array de datos de individuos de clase 2.

Write Predict Report

Proceso

El informe de predicción se compone de varias partes:

Cabecera, contiene el número de individuos predichos.

Individuos predichos de clase 1. Para cada individuo se escribe el nombre del individuo, su valor de p Z 1 y el nodo donde se encuentra el individuo. Para el caso B, el nodo será el de mayor probabilidad.

Individuos predichos de clase 2. Para cada individuo se escribe el nombre, su valor de p Z 1 y el nodo donde se encuentra el individuo. Para el caso B, el nodo será el de mayor probabilidad.

Individuos no predecibles. Para cada individuo se escribe el nombre y la descripción.

Si se esta procesando un algoritmo de tipo B, además se escriben las siguientes partes:

Individuos *no fiables* de clase 1. Para cada individuo se escribe el nombre, su probabilidad, su valor de *fiable* y el nodo en el que aparece.

Individuos *no fiables* de clase 2. Para cada individuo se escribe el nombre, su probabilidad, su valor de *fiable* y el nodo en el que aparece.

Proceso Describe

Flujos de entrada

Parameters, estructura con los parámetros de funcionamiento.

Proceso

No aplicable.

Write Report

Flujos de entrada

Parameters, estructura con los parámetros de funcionamiento.

Proceso

Escribe en el fichero de informe los siguientes apartados o informaciones:

Líneas del Copyright.

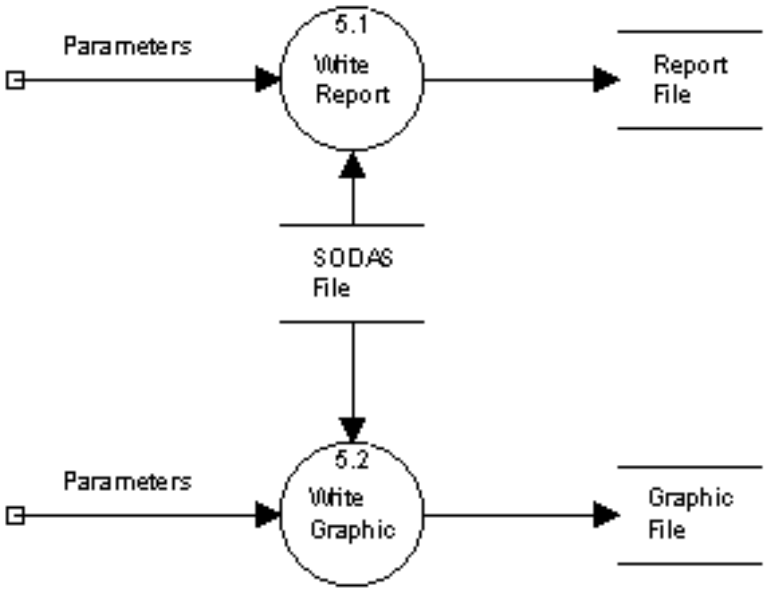
El índice de la variable *strata*, junto con su nombre y su descripción.

El índice de la variable *predictive*, junto con su nombre y su descripción.

El índice de cada *predictor*, junto con su nombre y su descripción.

Los parámetros de entrada. Los que se encuentran en la estructura de parámetros.

El porcentaje de nulos de la variable *strata* y de cada variable *predictor*.



La lista de individuos no usados por tener nulos.

Número de individuos a predecir.

Lista de *predictors* que no han sido utilizados en la construcción del árbol.

IC inicial y final.

Número de nodos explorados.

Lista de nodos decisionals. Cada nodo decisional contiene la siguiente información: identificación del nodo, variables que definen el nodo junto con sus valores, valor de la variable *strata* y *predictive*, peso del nodo, valor de IC y valor de IC_SDT.

Lista de nodos terminales. Cada nodo terminal contiene la siguiente información: identificación del nodo, variables que definen el nodo junto con sus valores, valor de la variable *strata* y *predictive*, peso del nodo, valor de IC y valor de IC_SDT.

Lista de estratos. Para cada estrato se escribe la identificación de los nodos en los que se encuentra dicho estrato. Sólo se buscará el estrato en nodos decisionales o terminales obtenidos por división del nodo. Para cada nodo, se escribe la contribución relativa del nodo para ese estrato.

Descripción detallada de la construcción del árbol. Esta información se va guardando en un fichero auxiliar según se construye el árbol y al final de la escritura del informe se vuelca en el fichero del informe. El fichero auxiliar se elimina.

Write Graphic

Flujos de entrada

Parameters, estructura con los parámetros de funcionamiento.

Proceso

El fichero gráfico de un árbol se compone de tres partes:

La cabecera del fichero.

La cabecera del árbol.

La lista de nodos.

Además, se escriben en este orden en el fichero de salida.

La cabecera del fichero contendrá los siguientes datos: nombre del fichero, fecha y hora de creación, nombre del fichero SODAS, número de *predictors* en el fichero de parámetros, lista de *predictors*, índice de la variable *strata* e índice de la variable *predictive*.

La cabecera del árbol contendrá los siguientes datos: tipo del árbol (NORMAL_TREE), valor inicial de IC, valor final de IC, nodo raíz del árbol, número de niveles y número de nodos.

Cada nodo se compondrá de: identificación del nodo, tipo del nodo (DECISIONAL, TERMINAL, NORMAL, EXPLORABLE), peso del nodo, valor de p Z 1, IC, contribución al IC del árbol (IC_SDT), número de estratos del nodo, lista de los estratos del nodo, peso de cada estrato, peso Z 1 de cada estrato, número de variables, lista de variables que definen el nodo, variable de corte, *indicador* que señala si ha sido divido o no, identificación del nodo izquierdo, identificación del nodo derecho, identificación del primer nodo decisional, identificación del segundo nodo decisional, identificación del primer nodo terminal e identificación del segundo nodo terminal.

Modelo de objetos

El modelo de objetos del sistema se muestra en la figura. En el modelo se puede ver que un árbol binario, SDT, se compone de un conjunto de nodos. Los nodos del árbol pueden ser de varios tipos:

Nodos de caso A (Node), son los nodos del árbol para el caso A pero sin incluir el primer nodo del árbol, su raíz.

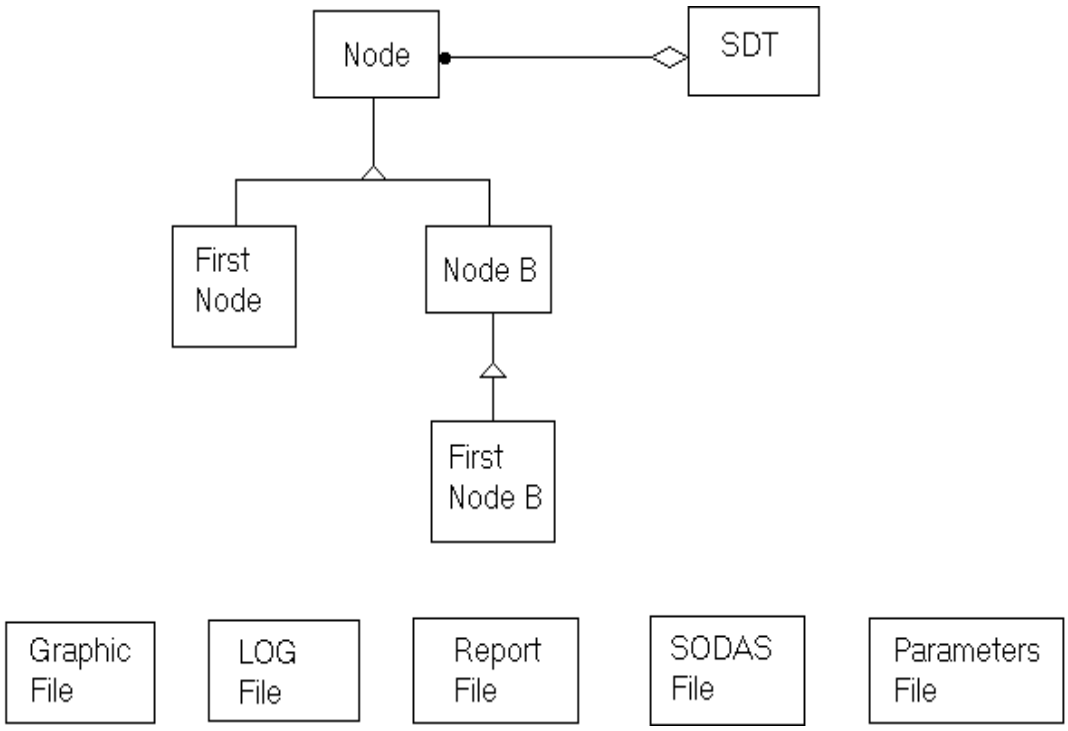
Primer nodo caso A (First Node), primer nodo del árbol para el caso A.

Nodos de caso B (Node B), son los nodos del árbol para el caso B pero sin incluir el primer nodo del árbol, su raíz.

Primer nodo caso B (First Node B), primer nodo del árbol para el caso B.

Existen otras clases en el sistema especializadas cada una de ellas en una tarea o en la gestión de un fichero en concreto. (Report File, LOG File, Graphic File, SODAS File y Parameters File).

La clase más importante es SDT la cuál contiene todos los métodos necesarios para la gestión del árbol.


Node
SDT
First Node
Node B
First Node B
Graphic File
LOG File
Report File
SODAS File
Parameters File

Apéndice B. Diseño del programa SDTEDITOR

Este apéndice contiene el diseño del *software* SDTEDITOR v 2.22[8] que es un editor gráfico para árboles de Segmentación para datos estratificados. Es de destacar en la elaboración de este diseño la colaboración de Alberto Fernández García, que ha realizado la codificación del *software*.

El editor gráfico es una aplicación diseñada entorno a un GUI (*Graphic User Interface*), es por tanto un conjunto de clases orientadas a gestionar los eventos que se generan en el GUI. La filosofía de las aplicaciones *Windows* se basa en gestionar los eventos que recibe la ventana principal de la aplicación. Dichos eventos, al final, son procesados por las clases que genera *Visual C++*[9] *v 5.0*, y dentro de ellos es donde se hace uso de las clases del usuario, las cuales constituyen realmente el corazón del sistema.

Ha sido diseñado utilizando el asistente que proporciona el entorno de desarrollo *Visual C++* por lo que se recomienda dirigirse a la documentación del *Visual C++* para aclarar cualquier duda sobre la forma de utilizarlo, modificaciones en el diseño, etc. Es una aplicación del tipo SDI (*Simple Document Interface*), es decir, gestiona un único documento a través de una única ventana. Se basa en tres clases principales:

- xxxxApp, donde xxxx es un nombre dado por el diseñador, en nuestro caso es **CEditorApp**. Esta clase se encarga de la gestión de la aplicación.
- xxxxDoc, **CEditorDoc**, encargada de gestionar el documento en sí: cargarlo, grabarlo.
- xxxxView, **CEditorView**, encargada de la gestión del proceso de dibujo del documento dentro de la ventana, de la gestión de los menús, etc.

Para la gestión de los árboles se necesitan otras dos clases:

[8] SDTEDITOR v 2.22 está integrado en el software SODAS 1.04, resultado del proyecto ESPRIT IV - 20821 SODAS - *Symbolic Official Data Analysis System.*

SODAS 1.04, Copyright CISIA Ceresta, 1999. http://www.cisia.com/

[9] *Copyright, 1994-97 Microsoft Corporation.*

- **CTtree**, gestiona el árbol; leerlo y dibujar usando **CEditorView**.
- **CNode**, proporciona métodos para gestionar un nodo del árbol.

Además, de las clases antes mencionadas el asistente de Visual C++ para cada dialogo que genera con su editor de diálogos también se genera una clase que lo gestiona:

- **CAboutDlg**, diálogo "About".
- **CDlgChangeOptions**, diálogo que gestiona las opciones de dibujo del árbol.
- **CDlgFindStrata**, diálogo para buscar una determinada categoría de la variable *strata* en el árbol.
- **CDumpNode**, diálogo que contiene el volcado de los datos de un nodo cuando se selecciona el nodo en cuestion.

A continuación se describen las clases componentes del sistema con sus métodos.

CDumpNode

Esta clase controla el cuadro de diálogo que aparece al pulsar con el botón izquierdo dentro de un nodo del árbol. Cuando se construye un objeto de esta clase se utiliza un puntero al nodo, objeto de la clase **CNode**, que se quiere visualizar. Este puntero es almacenado internamente y utilizado por el método *OnInitDialog* para acceder a los datos del nodo y mostrarlos en los elementos del cuadro de diálogo, como etiquetas, título, listas, etc. Es una clase interna, ya que no posee ningún método público. Los ficheros que contienen la definición e implementación de la clase son: DumpNode.cpp y DumpNode.h.

Métodos:

CDumpNode. Constructor. Inicializa los atributos de la clase.

DoDataExchange. Protegido, es gestionado por las MFC (*Microsoft Foundation Classes*).

OnInitDialog. Protegido. Este método es llamado antes de que se pinte la ventana en la pantalla. Se utiliza para rellenar los campos del diálogo antes de ser mostrados. Es decir, establece los valores que deben tener los campos en la ventana antes de dibujar la ventana.

Descripción de los métodos:

Método: **OnInitDialog**

Descripción: Es el constructor de la clase se recibe un puntero al nodo que se quiere mostrar, un objeto de la clase **CNode**. A través de ese puntero se obtienen todos los datos del nodo que se quieren visualizar. Dicho objeto contiene todos los métodos necesarios para leer todos los atributos necesarios.

CDlgFindStrata

Esta clase gestiona el cuadro de diálogo que se utiliza para buscar una categoría de la variable *strata* en los nodos del árbol. El diálogo se compone de un *combobox* o lista deplegable con todas las categorías de la variable estrato para que el usuario seleccione una de la lista y dos botones; uno de buscar que inicia la búsqueda y otro de cancelar que suspende la operación.

La clase solamente proporciona un método público, *Get_strata_Name*, que se llama una vez cerrado el diálogo para leer el nombre de la categoría que el usuario ha seleccionado y que va a ser utilizado en la búsqueda.

La *lista desplegable* se rellena en el método *OnInitDialog* que es llamado internamente por las clases MFC cuando se va a mostrar el diálogo en la pantalla. Este método obtiene un puntero al documento, objeto del tipo **CEditorDoc**, y a través de él un puntero al objeto **CTree** que contiene los datos del árbol que se está dibujando. Usando él objeto **CTree** se leen todos los datos de la variable *strata* entre los que se encuentran las categorías y que se utilizan para rellenar la *lista desplegable*. Los ficheros que contienen la definición e implementación de la clase son: DlgFindStrata.cpp y DlgFindStrata.h.

Métodos:

CDlgFindStrata. Constructor.

DoDataExchange. Protegido, es gestionado por las MFC.

OnInitDialog. Protegido. Este método es llamado antes de que se pinte la ventana en la pantalla. Se utiliza para rellenar los campos del diálogo antes de ser mostrados.

Get_Strata_Name. Devuelve el nombre del estrato que se quiere buscar.

OnCancel. Protegido, se llama cuando se pulsa el botón de Cancelar.

OnEdtStrataName. Protegido, se llama cuando se modifica el texto del nombre del estrato.

OnOK. Protegido, se llama cuando se pulsa el botón OK.

Descripción de los métodos:

Método: **Get_Strata_Name**

Valor de retorno: El nombre del estrato que se quiere buscar.

Descripción: Devuelve el contenido del atributo de la clase que contiene el nombre del estrato que se quiere buscar en el árbol.

Método: **OnOK**

Descripción: Obtiene el índice de la fila seleccionada en la lista de estratos, con él, lee la cadena de caracteres asociada al índice y almacena su valor en un atributo de la clase. Este atributo contiene el nombre del estrato que se quiere buscar.

Método: **OnInitDialog**

Descripción: A través del documento actual, obtiene un puntero al árbol que se está mostrando. Luego, a través del árbol se leen el número de categorías y

para cada categoría se lee su nombre y se añade a la lista de categorías de la ventana.

CDlgChangeOptions

Esta clase gestiona la ventana de diálogo para cambiar las opciones de dibujo del árbol. Tales opciones comprenden: número máximo de estratos por nodo, el número máximo de letras por estado, el número máximo de caracteres por línea, el umbral, etc. La ventana contiene varios campos de edición y varios *checkbox* o casilla de verificación para seleccionar diferentes opciones de dibujo. También, contiene dos botones uno para aceptar las opciones y otro para cancelar los cambios que se han realizado en la ventana. Esta ventana se mostrará cuando se seleccione la opción del menú "Cambiar opciones" y se active el *callback* o llamada de retorno asociada al elemento del menú. Los ficheros que contienen la definición e implementación de la clase son: DlgChangeOptions.cpp y DlgChangeOptions.h.

Métodos:

CDlgChangeOptions. Constructor.

DoDataExchange. Protegido, es gestionado por las MFC.

OnCancel. Protegido. Se llama cuando se pulsa el botón de Cancelar.

OnChkDecNode. Protegido. Activa o desactiva el *flag* o indicador que señala si se deben visualizar los datos en los nodos decisionales.

OnChkTerNode. Protegido. Activa o desactiva el *indicador* que señala si se deben visualizar los datos en los nodos terminales.

OnInitDialog. Protegido. Inicializa los controles de la ventana según los valores de los atributos. Es decir, que radio box deben aparecer marcados y cuales no.

OnOK. Protegido. Se llama cuando se pulsa el botón de Aceptar.

Retrieve_Data. Copia los atributos de la ventana en la estructura indicada en el parámetro de entrada.

Set_Data. Copia los datos de la estructura, que es el parámetro de entrada, en los atributos de la clase.

CAboutDlg

Esta clase gestiona el cuadro del diálogo "About" que contiene un mensaje sobre el autor de la aplicación y el copyright. Solamente contiene un botón de aceptar para cerrar el diálogo. Se utiliza dentro del método *OnAppAbout* de la clase **CEditorApp**. Este método se activará cuando el usuario pulse sobre la opción "Help/About" del menú. Los ficheros que contienen la definición e implementación de la clase son: Editor.cpp y Editor.h.

CTree

Esta clase gestiona los datos de un árbol y todos los nodos que lo forman. Contiene operaciones par a leerlo de un fichero, grabarlo en un fichero, mostrarlo en la pantalla e imprimirlo.Los ficheros que contienen la definición e implementación de la clase son: Tree.cpp y Tree.h.

Métodos:

CTree. Constructor. Inicializa los atributos de la clase.

~CTree. Destructor. Libera los recursos utilizados en la clase. Llama al método Free.

Extract. Protegido. Trozea una línea.

Extract_Ids. Protegido. A partir de un identificador obtiene el nivel, el orden, el tipo del nodo y el índice.

Find. Protegido. Busca un nodo en el árbol y devuelve un puntero al nodo.

Find_Node. Busca el nodo sobre el que se ha pulsado con el ratón y devuelve un puntero al nodo.

Fix_Pointer. Protegido. Transforma los identificadores de los nodos en punteros a los objetos que contiene los datos del nodo.

Free. Protegido. Libera la memoria utilizada por el árbol.

Get_Num_Levels. Devuelve el número de niveles del árbol.

Get_Predictor. Devuelve el índice predictor indicado.

Get_Relevant. Devuelve el valor "relevant threshold" del árbol.

Get_Total_Height. Devuelve la altura total del árbol. Del rectángulo que contiene al dibujo del árbol.

Get_Total_Width. Devuelve la anchura total del árbol.

Get_Variable. Devuelve los datos de una variable.

Load. Carga los datos de un árbol de un fichero.

Print. No implementada.

Read_Header. Protegido. Lee la cabecera de un fichero de un árbol.

Read_Nodes. Protegido. Lee todos los nodos de un árbol.

Read_Tree_Header. Protegido. Lee la cabecera de un árbol.

Recalculate_Tree. Recalcula todos los valores de un árbol.

Save. No implementada.

Show. Muestra un árbol en la pantalla.

Signature. Decodifica un firma "signature".

Descripción de los métodos:

Método: **Extract**
Valor de retorno: código de error.
Método: **Extract_Ids**
Valor de retorno: código de error.
Método: **Fix_Pointer**
Valor de retorno: código de error.
Método: **Load**

Valor de retorno: código de error.
Método: **Read_Header**
Valor de retorno: código de error.
Método: **Read_Nodes**
Valor de retorno: código de error.
Método: **Read_Tree_Header**
Valor de retorno: código de error.
Método: **Recalculate_Tree**
Valor de retorno: código de error.
Método: **Show**
Valor de retorno: código de error.
Método: **Signature**
Valor de retorno: código de error.

CNode

Esta clase gestiona los nodos que componen un árbol. Puede gestionar los tres tipos de nodos: decisionales, terminales y normales. Los nodos del árbol se pueden leer y/o mostrar en la pantalla. También contiene métodos para leer casi todos los atributos de la clase. Un nodo se compone de dos posibles nodos hijos, izquierdo y derecho, dos posibles nodos decisionales y dos posibles nodos terminales. Es decir, un objeto de la clase **CNode** contiene los datos de un nodo y los punteros todos los demás nodos relacionados con el. Los ficheros que contienen la definición e implementación de la clase son: Node.cpp y Node.h.

Métodos:

CNode. Constructor. Almacena el puntero al árbol al que pertenece.

~CNode. Destructor.

Calculate. Estático. Calcula una serie de atributos internos del nodo en función del tipo de letra seleccionado.

Calculate_Boxes. Calcula las posiciones de las cajas que van a contener las diferentes partes del nodo.

Extract. Trocea una línea.

Get_Box_Height. Estático. Devuelve la altura de la caja que contiene al nodo.

Get_Box_Width. Estático. Devuelve la anchura de la caja que contiene al nodo.

Get_Center_X. Devuelve la coordenada x del centro de la caja que contiene al nodo.

Get_Center_Y. Devuelve la coordenada y del centro de la caja que contiene al nodo.

Get_Cut_Variable. Devuelve el nombre de la variable de corte.

Get_IC. Devuelve el valor IC del nodo.

Get_IC_SDT. Devuelve el valor IC_SDT del nodo.

Get_Id. Devuelve la identificación del nodo.

Get_Id_Decisional_1. Devuelve la identificación del primer nodo decisional.

Get_Id_Decisional_2. Devuelve la identificación del segundo nodo decisional.

Get_Id_Left. Devuelve la identificación del nodo hijo izquierdo.

Get_Id_Right. Devuelve la identificación del nodo hijo dererecho.

Get_Id_Terminal_1. Devuelve la identificación del primer nodo terminal.

Get_Id_Terminal_2. Devuelve la identificación del segundo nodo terminal.

Get_Num_Stratas. Devuelve el número de estratos que posee el nodo.

Get_Num_Variables. Devuelve el número de variables que posee el nodo.

Get_p_Z_1. Devuelve el valor de p_Z_1 del nodo.

Get_Parent. Devuelve el puntero al árbol al que pertenece el nodo, objeto de la clase **CTree.**

Get_Strata. Devuelve el nombre de un estrato que pertenece al nodo.

Get_Type. Devuelve el tipo del nodo.

Get_Variable. Devuelve el nombre de una variable del nodo.

Get_Weight. Devuelve el peso del nodo.

Get_Weight_Strata. Devuelve el peso de un estrato del nodo.

Get_Weight_Z_1_Strata. Devuelve el peso de Z 1 de un estrato del nodo.

Load. Carga los datos del nodo de un fichero.

Print. No implementada.

Save. No implementada.

Set_Left_Flag. Establece el *indicador* que señala si hay un nodo decisional a la izquierda. No usado.

Set_Node. Almacena el puntero a un nodo relacionado con éste.

Show. Muestra un nodo en la pantalla.

Show_Decisional. Protegido. Muestra un nodo decisional en la pantalla.

Show_Terminal. Protegido. Muestra un nodo terminal en la pantalla.

Where. Devuelve el puntero al nodo sobre el que se ha pulsado con el ratón.

Descripción de los métodos:

Método: **Calculate**
Valor de retorno: código de error.
Método: **Calculate_Boxes**
Valor de retorno: código de error.
Método: **Extract**
Valor de retorno: código de error.
Método: **Load**
Valor de retorno: código de error.
Método: **SetNode**
Valor de retorno: código de error.

Método: **Show**
Valor de retorno: código de error.
Método: **Show_Terminal**
Valor de retorno: código de error.
Método: **Show_Decisional**
Valor de retorno: código de error.
Método: **Where**
Valor de retorno: código de error.

CEditorDoc

Esta la clase principal de la aplicación, contiene todos los datos del "documento" con el que se está trabajando. En nuestro caso el documento es un árbol de estratos. También contiene la estructura de parámetros que es global para toda la aplicación. Los ficheros que contienen la definición e implementación de la clase son: EditorDoc.cpp y EditorDoc.h.

Métodos:

CEditorDoc. Constructor. Protegido. Llama al método **Initialize_Parameters**.

~CEditorDoc. Destructor.

AssertValid. Se utiliza para depurar.

CopyParameters. Protegido. Guarda los parámetros en una estructura temporal, el atributo Copy.

Dump. Se utiliza para depurar.

Extract. Privado. Procesa una línea de texto para obtener las partes que la forman.

Get_Parameters. Obtiene una copia de la estructura de parámetros.

Get_Tree. Devuelve un puntero al objeto **CTree**, que contiene los datos del árbol que se está visualizando.

GetMode. Devuelve el modo actual de visualización del árbol. Los modos pueden ser "Fit in window" o "Normal size".

Initialize_Parameters. Inicializa la estructura de parámetros con sus valores por defecto.

ModeFitWindow. Establece el modo de visualización a "Fit In Window".

ModeNormalSize. Establece el modo de visualización a "Normal Size".

OnNewDocument. Llamada de retorno que se activa cuando se pulsa la opción de menú "Nuevo documento".

OnOpenDocument. Llamada de retorno que se activa cuando se pulsa la opción de menú "Abrir documento". Lee un fichero de parámetros, con extensión ".vsg", o un fichero de un árbol.

OnSaveDocument. Llamada de retorno que se activa cuando se pulsa la opción de menú "Guardar documento".

Read_Parameters. Privado. Realiza la lectura del fichero de parámetros.

Recalculate. Activa el *indicador* de recalcular, para que se repinte el árbol actual.

RestoreParameters. Restablece la copia temporal de los parámetros, copia el contenido del atributo "copy" en la estructura de parámetros globales.

Serialize. No usado.

Set_Parameters. Cambia los parámetros globales por los que hay en la estructura que se recibe como parámetro de entrada.

Descripción de los métodos:

Método: **Extract**

Parámetros entrada: input_line – Línea que se quiere trozear

Parámetros salida: left_part – Parte de la izquierda; right_part – Parte de la derecha.

Valor de retorno: código de error.

Descripción: Recibe una línea de texto que se supone es de la forma "variable = valor" y la separada en dos partes, utilizando para ello el signo igual "=", es decir, lo que está a la izquierda del igual se almacena en el parámetro de salida left_part y lo que está a la derecha en right_part. Si no aparece el signo igual en la línea se almacena todo en left_part.

Método: **OnNewDocument**

Parámetros entrada: lpszPathName – Nombre del fichero que se quiere abrir.

Valor de retorno: código de error.

Descripción: Este método es llamado cuando se pulsa la opción de menú "Abrir". Puede abrir dos tipos de documentos en función de la extensión del fichero de entrada.

Lo primero que hace es obtener la extensión del fichero. Si la extensión es "vsg" o "VSG" se llama al método **Read_Parameters** y se almacena el código de error que devuelve el método en exit_code. Si la extensión es diferente se llama al método **OnOpenDocument** de la clase base, gestionado por las MFC, a continuación se llama al método **Load** de la clase **CTree** para leer los datos del árbol y almacenarlos dentro del atributo Tree, el código de error que devuelve este método se almacena en la variable exit_code y por último, se llama al método **Initialize_Parameters** y se almacena el "relevant threshold" del árbol, valor devuelto por **Tree.Get_Relevant**, en la estructura de parámetros globales.

Al final, si la variable exit_code contiene el valor SUCCESS se asigna el valor TRUE al campo "recalculate" de la estructura de parámetros globales para que se pinte el árbol que se ha leído.

Método: **Read_Parameters**

Parámetros entrada: file_name – Nombre del fichero de parámetros.

Valor de retorno: código de error.

Descripción: Se comprueba el parámetro de entrada, si es un puntero nulo se devuelve un código de error. Se abre el fichero, usando para ello el nombre del fichero recibido, si se produce un error, se muestra un mensaje en la pantalla y se devuelve un código de error.

Se lee una línea del fichero. El método se llama **Extract** y se utilliza para extraer las dos partes que forman la línea porque se supone que es de la forma "variable = valor".

Si la parte de la izquierda es igual a "PATH_FILIERE", se almacena la parte de la derecha en la variable Input_Path. Si la parte de la derecha es igual a "LISTE_FICHIERS", se almacena en la parte de la derecha en la variable first_name. Se procesa todo el fichero hasta el final.

Se cierra el fichero. Se elminina los blancos a la izquierda y la derecha del nombre del fichero, contenido en la variable first_name. Si el nombre está vacío, se mostrará un mensaje de error en la pantalla y se devolverá un código de error.

Se forma el nombre completo del fichero. Es la suma del path contenido en Input_Path más el nombre del fichero contenido en first_name. El método se llama **Load** de la clase **CTree** para leer los datos del árbol. Por último, se llama al método **Initialize_Parameters** y se almacena el "relevant threshold" del árbol, valor devuelto por **Tree.Get_Relevant**, en la estructura de parámetros globales.

CEditorView

Esta es otra de las clases principales de la aplicación. Controla como se muestra el documento en uso en la ventana de la aplicación. En nuestro caso como se dibuja el árbol en la ventana. Además, controla la pulsación de los botones del ratón sobre el árbol. Los ficheros que contienen la definición e implementación de la clase son: EditorView.cpp y EditorView.h.

Métodos:

CEditorView. Constructor. Protegido.

~CEditorDoc. Destructor.

AssertValid. Se utiliza para depuración.

Dump. Se utiliza para depuración.

GetDocument. Devuelve un puntero al documento actual, un objeto de la clase **CEditorDoc.**

OnBeginPrinting. Este método se llamará antes de iniciar la impresión del árbol. No utilizado.

OnPreparePrinting. No utilizado.

OnEndPrinting. No utilizado.

OnDraw. Este método será llamado cada vez que se quiere redibujar el árbol en la ventana. Es decir, lo pinta en la ventana.

OnInitialUpdate. Se llama la primera vez que se dibuja la ventana, pero antes de que se vea nada en la pantalla.

OnLButtonUp. Este método se llamará cuando se suelte el botón izquierdo del ratón. Es decir, cuando después de pulsar el botón se suelta.

OnRButtonUp. Este método se llamará cuando se suelte el botón derecho del ratón.

PreCreateWindow. Este método se llamará antes de crear la ventana de la aplicación.

Descripción de los métodos:

Método: **OnDraw**

Parámetros entrada: pDC – Puntero al "Display Context" donde se va a dibujar.

Descripción: Lee el puntero al documento actual llamando al método **GetDocument**, lo almacena en la variable pDoc. Lee el puntero al árbol que se quiere dibujar llamando al método **Get_Tree**, usando pDoc, se almacena en la variable pointer. Lee los parámetros llamando al método **GetParameters**, usando pDoc, los parámetros se almacenan en la variable parameters.

Si hay que recalcular el árbol, la variable Recalculate de la estructura de parámetros contiene el valor TRUE, se llama al método **RecalculateTree** con los parámetros leídos, se utiliza el puntero al árbol *pointer*. Se lee el ancho total que ocupa el dibujo del árbol, si es menor de 300, se hace igual a 300. Se lee el alto total del árbol, si es menor de 300, se hace igual a 300. Se desactiva el *indicador* de recalcular, se almacena el valor FALSE en la variable Recalculate de la estructura de parámetros. Después, se guardan los parámetros actualizados en el documento llamando al método **Set_Parameteres** usando pDoc.

Se crea una fuente, font, con los parámetros de tamaño, parameters.Font_Size, y nombre de la fuente, parameters.Face_Name. Se almacena en el objeto tmp_font. Se selecciona la fuente creada en el "Display Context" y se dibuja el árbol en el "Display Context" llamando al método **Show**, usando el puntero al árbol *pointer*.

Método: **OnInitialUpdate**

Valor de retorno: código de error.

Descripción: Establece el tamaño inicial, a 1000 por 1000, del área cliente de la ventana donde se va a dibujar el árbol.

Método: **OnLButtonUp**

Parámetros entrada: nFlags – Flags que indica el tipo de acción que se ha efectuado; point – Posición donde se ha pulsado el ratón.

Valor de retorno: código de error.

Descripción: Primero se lee el puntero al documento actual, llamando al método **GetDocument**, se almacena en pDoc.

Si el modo actual es "Fit Window", se redibuja el árbol a tamaño normal centrándolo en el punto de la pantalla donde se ha pulsado con el ratón y se cambia el modo de visualización a "Normal Size".

Si el modo es "Normal Size" se lee el puntero al árbol, llamando al método **Get_Tree**, se almacena en la variable pTree. Se lee el origen la página de *scroll* o desplazamiento que se está visualizando actualmente, se almacena en la variable origin. Se busca el nodo sobre el que se ha pulsado, llamando al método **Find_Node** usando el puntero pTree. Se utiliza el punto donde se ha pulsado con el ratón transformado según el origen de la página. Si se encuentra el nodo,

se crea un objeto temporal de la clase **CDumpNode**, con el puntero al nodo, y se muestra el diálogo con los datos del nodo.

Método: **OnRButtonUp**

Parámetros entrada: nFlags – Flags que indica el tipo de acción que se ha efectuado; point – Posición donde se ha pulsado el ratón.

Valor de retorno: código de error.

Descripción: Lee el puntero al documento actual, llamando al método **GetDocument**, se guarda en la variable pDoc. Si el modo de visualización es "Fit Window" no se hace nada. En caso contrario, se copian los parámetros globales en una estructura temporal, llamando al método **CopyParameters**, usando pDoc. Se lee los parámetros actuales, llamando al método **Get_Parameters**, usando pDoc. Se desactivan todos los *indicadores* que muestran la información, es decir, no se mostrará información en los nodos decisionales, ni terminales, ni en los nodos. Se almacenarán los nuevos parámetros en el documento, llamando al método **Set_Parameters**. Se redibujará el árbol entero haciendo que encaje en el área cliente de la ventana. Por último, se cambiará el modo de visualización a "Fit Window"

CEditorApp

Esta la clase aplicación, deriva de la clase **CWinApp** que es la clase que base para las aplicaciones de las MFC. Se encarga de construir la ventana principal de la aplicación y mostrarla en la pantalla. Crea las tres clases principales de la aplicación para la gestión del documento, los menús y la ventana. Estas clases son: **CEditorDoc**, **CMainFrame** y **CEditorView**. Los ficheros que contienen la definición e implementación de la clase son: Editor.cpp y Editor.h.

Métodos:

CEditorApp. Constructor.

InitInstance. Construye la ventana y la muestra en la pantalla. También crea la estructura necesaria para la aplicación tipo SDI.

OnAppAbout. Se llama cuando se pulsa la opción de menú "About". Crea un objeto de la clase **CAboutDlg** y muestra en pantalla su ventana. Espera hasta que se cierra dicha ventana.

CMainFrame

Esta clase aplicación es la encargada del gestionar la barra de menús de la aplicación. Contiene los *llamada de retorno*s que se llamarán cuando se pulsen las diversas opciones del menú. Lo normal es que en cada *llamada de retorno* se cree un objeto temporal de la clase que controla el diálogo que se quiere mostrar y se trabaje con ella. Los ficheros que contienen la definición e implementación de la clase son: MainFrm.cpp y MainFrm.h.

Métodos:

CMainFrame. Protegido. Constructor.

~CMainFrame. Desctructor.

AssertValid. Se utilizará para depurar.

Dump. Se utilizará para depurar.

OnChangeFont. Protegido. Esta *llamada de retorno* será llamado cuando se quiera cambiar la fuente de letras.

OnFitWindow. Protegido. Esta *llamada de retorno* será llamado cuando se quiera mostrar todo el árbol encajado dentro del área cliente de la ventana.

OnOptionsChange. Protegido. Esta *llamada de retorno* será llamado cuando se quiera cambiar alguna opción de dibujo del árbol.

OnOptionsFind. Protegido. Esta *llamada de retorno* será llamado cuando se quiera buscar un estrato en el árbol.

OnRestoreSize. Protegido. Esta *llamada de retorno* será llamado cuando se quiera restaurar el tamaño al que se debe de mostrar el árbol.

PreCreateWindow. Se llamará antes de crear la ventana por si se desea modificar algunos de sus atributos.

Descripción de los métodos:

Método: **OnChangeFont**

Descripción: Se lee el puntero al documento actual, llamando al método **GetActiveDocument**, se almacena en la variable pDoc. Se comprueba el modo de visualización, si es "Fit Window" se retornará sin hacer nada. Se leen los parámetros actuales, llamando al método **Get_Parameters**, se almacenan en la variable parameters.

Se crea un objeto temporal de la clase **CFontDialog**, de las MFC, y se muestra en pantalla hasta que se pulse alguno de los botones de Cancelar o Aceptar. Si se pulsa Aceptar u OK, se leerán los datos de la fuente seleccionada, llamando al método **GetCurrentFont**, se almacenan en la variable struct_font. Se actualiza en la variable parameters el nombre de la fuente y su tamaño, ambos campos se leen de la estructura struct_font. Se vuelven a guardar los parámetros en el documento llamando al método **Set_Parameters**, usando pDoc y la variable parameters. Al final, se redibuja el árbol.

Método: **OnFitWindow**

Descripción: Se lee el puntero al documento actual, llamando al método **GetActiveDocument**, se almacena en la variable pDoc. Se comprueba el modo de visualización, si es "Fit Window" se retornará sin hacer nada.

Se copian los parámetros en una estructura temporal. Se lee los parámetros globales llamando al método **Get_Parameters**, se desactivan todos los *indicadores* que muestran los datos de los nodos y vuelve a guardar los parámetros llamando al método **Set_Parameters**. Se redibuja el árbol.

Se obtiene un puntero a la vista actual, objeto del tipo **CView**, y se reduce el dibujo hasta encajarlo dentro del área cliente de la ventana. Se cambia el modo de visualización el árbol a "Fit Window" llamando al método **ModeFitWindow**.

Método: **OnOptionsChange**

Descripción: Se lee el puntero al documento actual, llamando al método **GetActiveDocument**, se almacena en la variable pDoc. Se comprueba el modo de visualización, si es "Fit Window" se retornará sin hacer nada.

Se crea un objeto de la clase **CDlgChangeOptions**. Se leen los parámetros llamando al método **Get_Parameters** y se almacenan dentro del objeto que va a mostrar el diálogo, llamando al método **Set_Data** de la clase **CDlgChangeOptions**. Se muestra la ventana del diálogo en la pantalla y se espera a que el usuario interactue con ella. Si se sale del diálogo pulsando el botón de OK, se leen los nuevos valores de los parámetros llamando al método **Retrieve_Data** de la clase **CDlgChangeOptions**, y se guardan los nuevos parámetros llamando al método **Set_Parameters**. Después se llama al método **Recalculate** para recalcular todos los valores del árbol y se muestra éste en la pantalla.

Método: **OnOptionsFind**

Descripción: Se lee el puntero al documento actual, llamando al método **GetActiveDocument**, se almacena en la variable pDoc. Se comprueba el modo de visualización, si es "Fit Window" se retornará sin hacer nada.

Se crea un objeto de la clase **CDlgFindStrata** y se muestra en la pantalla el diálogo. Se espera hasta que el usuario pulse alguno de los botones.

Si se sale el diálogo pulsando el botón de OK, se lee el nombre del estrato que se quiere buscar llamando al método **Get_Strata_Name**, si la longitud del nombre del estrato es distinta de cero, se activará el *indicador* de búsqueda de un estrato en la estructura de parámetros y se guardará su nombre en la estructura. Se vuelve a guardar los parámetros llamando al método **Set_Parameters**, y finalmente se redibuja el árbol en la pantalla.

Método: **OnRestoreSize**

Descripción: Se lee el puntero al documento actual, llamando al método **GetActiveDocument**, se almacena en la variable pDoc. Se comprueba el modo de visualización, si es "Normal Size" se retornará sin hacer nada.

Se restaura la copia temporal de los parámetros que se almacena en el documento, llamando al método **RestoreParameters**, se recalculan todos los valores del árbol, llamando al método **Recalculate**, y se dibuja el árbol en la pantalla. Al final, se establece el modo de visualización a "Normal Size", llamando al método **ModeNormalSize**.

Bibliografía

Bibliografía

[1] [10]**Del Amo Blanco, A.I.** (1999), *Modelos de Agregación Recursiva y su Aplicación a la Clasificación Difusa de Imágenes Digitales,* Tesis Doctoral, Dpto. de Estadística e Investigación Operativa, Facultad de Ciencias Matemáticas, Universidad Complutense de Madrid.

[2] **Araya, R.** (1995), *Induction of Decision Trees when Examples are Described with Noisy Measurements and with Fuzzy Class Membership,* Seminario INRIA, París.

[3] **Aristotle** (IV a.c.), *Organon,* Vol. 1, Catégories, Vol. 2, De l'interprétation. J. Vrin Ed., Paris, 1994.

[4] **Arnault, A., Nicole, P.** (1662) *La Logique ou l'Art de Penser*, Ed. Fromann, Stuttgart (1965).

[5] **Bacelar-Nicolau, H.** (2000), The Affinity Coefficient, En: Bock, H.H., Diday, E. Eds., 2000a, 160-164.

[6] **Bandemer, H., Näther,W.** (1992), *Fuzzy Data Analysis*, Kluwer, Dordretch.

[7] **Belson, W.A.** (1959), Matching and Prediction on the Principle of Biological Classification, *Applied Statistics,* **VIII**.

[8] **Benzecri, J.P.** (1980), *L'Analyse des Données I. La Taxinomie,* 3ªEd., Dunod.

[9] **Bergomier, H., Boucharenc, L.** (1966), *Une Méthode de Segmentation basée sur la Théorie de l'Information,* Prix Marcel Dassault.

[10] **Bertrand, P., Goupil, F.** (2000), Descriptive Statistics for Symbolic Data. En: Bock, H.H., Diday, E. Eds., 2000a, 106-124.

[10]Nota: Si varios artículos de esta bibliografía pertenecen al mismo volumen, la referencia completa del mismo se presenta como una entrada independiente.

[11] **Bisdorff,R.** (2000), Professional Careers of Retired Wroking Persons, En: Bock, H.H., Diday, E. Eds., 2000a, 356-374.

[12] **Bock, H.H.** (2000a), 1. The Classical Data Situation. 2. Symbolic Data. En: Bock, H.H., Diday, E. Eds., 2000a, 24-53.

[13] **Bock, H.H.** (2000b), Dissimilarity Measures for Probability Distributions. En: Bock, H.H., Diday, E. Eds., 2000a, 153-160.

[14] **Bock, H.H., Diday, E., Editores** (2000a), *Analysis of Symbolic Data. Exploratory Methods for Extracting Statistical Information from Complex Data*. Studies in Classification, Data Analysis and Knowledge Organization, Springer, Heidelberg.

[15] **Bock, H.H., Diday, E.** (2000b), Symbolic Objects, En: Bock, H.H., Diday, E. Eds., 2000a, 54-77.

[16] **Bouillet, C., Grandin, J.F.** (1997), *Management Plan*, *SODAS Project* (20821 DG34/D-3/300536) v02.

[17] **Bouroche, J., Tenehaus,M.** (1970), Quelques Méthodes de Segmentation, *Rev. Française de Informatique e Recherche Operationelle (RIRO)*, **4**, 29-42.

[18] **Bravo Llatas, M.C.** (1991), *Estrategia Estadística en el Sistema de Mejora del Conocimiento MACABE*, Trabajo de Investigación, Dpto. Estadística e Investigación Operativa, Universidad Complutense de Madrid.

[19] ———————— (1994), Building a Knowledge Base for Correspondence Analysis, *Qüestiió: Quaderns d'Estadística, Sistemes, Informàtica i Investigació Operativa*, **18**, 1, 51-73.

[20] **Bravo, M.C.** (1999a), *Software Requirements Specification for the Strata Decision Tree CSCI*, *SODAS Project* (20821 DG34/D-3/300536) v06.

[21] ———————— (1999b), *Interface Requirements Specification for the Strata Decision Tree CSCI*, *SODAS Project* (20821 DG34/D-3/300536) v06.

[22] ———————— (1999c), *Software Test Plan / Description for the Strata Decision Tree CSCI, SODAS Project* (20821 DG34/D-3/300536) v04.

[23] ———————— (1999d), *Software Test Results for the Strata Decision Tree CSCI, SODAS Project* (20821 DG34/D-3/300536) v01.

[24] ———————— (1999e), *Software Development Plan for the Strata Decision Tree CSCI, SODAS Project* (20821 DG34/D-3/300536) v01.

[25] ———————— (1999f), *Version Description Document for the Strata Decision Tree CSCI, SODAS Project* (20821 DG34/D-3/300536) v04.

[26] ———————— (1999g), *Software User Manual for the Strata Decision Tree CSCI, SODAS Project* (20821 DG34/D-3/300536), *v04*. Comm. of the EC-DgIII-Eurostat.

[27] ———————— (2000a), Strata Decision Tree Symbolic Data Analysis Software. En: H.A.L. Kiers, J.P. Rasson, P.J.F Groenen, M. Shader (eds.) *Data Analysis, Classification and Related Methods*. Studies in Classification, Data Analysis and Knowledge Organization, Springer Verlag, Heidelberg, 409-415.

[28] ———————— (2000b), Strata Decision Tree Symbolic Data Analysis Software. En: *IFCS'00 Data Analysis, Classification and Related Methods. Program and Abstracts*, University of Namur, Bélgica, p.31.

[29] **Bravo, M.C., García-Santesmases, J.M.** (1997), Segmentation Trees for Stratified Data. En: Jansen, J., Lauro, C.N. Eds: *Applied Stochastic Models and Data Analysis: The Ins/Outs of Solving Real Problems*. Curto, Nápoles, 37-42.

[30] ———————— (1998), Symbolic Object Description of Strata by Segmentation Trees, En: *NTTS'98, New Techniques & Technologies for Statistics*, Estudio Idea Ed., Nápoles, 85-90.

[31] ———————— (2000a), Segmentation Trees for Stratified Data. En: Bock, H.H., Diday, E. (Eds.) *Analysis of Symbolic Data. Exploratory Methods for Extracting Statistical Information from Complex Data*. Springer Verlag, Heidelberg, 266-293.

[32] ———————— (2000b), Symbolic Object Description of Strata by Segmentation Trees, *Computational Statistics*, Physica Verlag, Heidelberg, **15**, 13-24.

[33] **Bravo Llatas, M.C., Marina Rufas, J.** (1996), A Knowledge Enhancement System for Correspondence Analysis, En: Prat,A., Ripoll,E., Eds. Compstat 1996: Proceedings in *Computational Statistics*, Institut d'Estadística de Catalunya, 163-164.

[34] **Breiman, L., Friedman, J.H., Olshen, R.A. y Stone, C.J.** (1984), *Classification and Regresion Trees,* Wadsworth International Group.

[35] **Brito, P.** (2000), Hierarchichal and Pyramidal Clustering, En: Bock, H.H., Diday, E. Eds., 2000a, 312-324.

[36] **Buntine, W, Niblett, T.** (1992), A Further Comparison of Splitting Rules for Decision Tree Induction, *Machine Learning*, **8**, 75-85.

[37] **De Carvalho, F.A.T.** (1994), Proximity Coefficients between Boolean Symbolic Objects, En: Diday, E. et al., Eds. (1994), 387-394.

[38] **De Carvalho, F.A.T., Diday, E.** (1998), Indices de Proximité entre Objects Symboliques qui tient compte des Constraintes dans l'Espace de Description. En: Diday, E., Kodratoff, Y. Eds., *Induction Symbolique et Numerique à partir de Données.* Cépaduès Editions, Toulouse.

[39] **Casey, R.G., Nagy, G.** (1984), Decision Tree Design using a Probabilistic Model, *IEEE Transactions on Information Theory,* **30**, 93-99.

[40] **Cazes, P., Chouakria, A., Diday, E., Schektman, Y.** (1997), Extension de l'Analyse en Composantes Principales à des Données de type Intervalle, *Revue de Statistique Appliquée*, **XIV**(3), 5-24.

[41] **Celeux, G., Lechevallier, Y.** (1982), Méthodes de Segmentation non Parametriques, *Revue de Statistique Appliquée,* **XXX**, *4, 39-53.*

[42] **Cellard, J.C., Labbé, B., Savitsky, G.** (1967), Le Programme ELISSE - Présentation et Application, *METRA*, **6**, 3, 503-520.

[43] **Chabanon, C., Lechevallier, Y., Milleman, S.** (1992), Proposition d'une construction efficace d'un réseau de neurones á partir d'un arbre de décision, *Traitament des Connaissances "Symboliques-Numériques",* Lise-Ceremade Ed., Université de Paris IX-Dauphine, 259-272.

[44] **Chavent, M.** (1996), Recherche de Partitions en Analyse des Données Symboliques, *École d'Été Sept. 1996*, Lise-Ceremade, Université de Paris-IX Dauphine.

[45] ———————— (2000), Criterion-Based Divisive Clustering for Symbolic Data, En: Bock, H.H., Diday, E. Eds., 2000a, 299-311.

[46] **Chouakria, A., Cazes, P., Diday, E.** (2000), Symbolic Principal Component Analysis, En: Bock, H.H., Diday, E. Eds., 2000a, 200-233.

[47] **Ciampi, A.** (1992), Constructing Prediction Trees from Data: The REPCAM Approach, En: Antoch, J. (Ed.) *Computational Aspects of Model Choice,* Physica-Verlag, Heidelberg, 105-152.

[48] ———————— (1994), Classification and Discrimination: The RECPAM Approach, En: Dutter, R., Grossman, W. Eds. *Compstat 1994*, Physica-Verlag, 129-147.

[49] **Ciampi, A., Diday, E., Lebbe, J. y Vignes, R.** (1993), Recursive Partition and Symbolic Data Analysis*, IFCS'93,* Paris.

[50] ———————— (1994), Recursive Partition and Symbolic Data Analysis*,* En: Diday, E. et al., Eds. (1994), 277-284.

[51] **Ciampi, A., Diday, E., Lebbe, J., Périnel, E., Vignes, R.** (1996), Recursive Partition with Probabilistically Imprecise Data. En: Diday, E., Lechevallier, Y., Opitz, O., Eds.: *Ordinal and Symbolic Data Analysis (OSDA'95)*, Springer Verlag, 201–212.

[52] **Cox, L.A.,Jr.** (1993), Combining the Probability Judgements of Experts: Statistical and Artificial Intelligence Approaches,En: Hand, D.J. Ed. *Artificial Intelligence Frontiers in Statistics*, Chapman & Hall.

[53] **Crawford, S.L.** (1989), Extensions of the CART Algorithm, *International Journal Man-Machine Studies*, **31**, 197-217.

[54] **Csernel, M.** (1998), *Software Requirements Specifications / Interface Requirements Specification for the Symbolic Object Management CSCI, SODAS Project* (20821 DG34/D-3/300536), *v02.*

[55] **Cuesta Álvaro, P.L.** (1989), *Inducción en Bancos de Datos Cualitativos,* Tesis Doctoral, Dpto. de Estadística e Investigación Operativa, Facultad de Ciencias Matemáticas, Universidad Complutense de Madrid.

[56] **Dempster, N.P., Laird, N.M., Rubin, D.B.** (1977), Maximum Likelihood from Incomplete Data via the EM Algorithm, *Journal of the Royal Statistical Society*, Serie B, **39**, 1-38.

[57] **Diday, E.** (1987), Introduction à l'Aproche Symbolique en Analyse de Données, *Première Journées 'Symboliques-Numeriques' pour l'Apprentissage à partir de Données*, CEREMADE, Université Paris IX-Dauphine, 21-56.

[58] ———————— (1988), Introduction à l'Analyse des Donnéss Symboliques: Objetos Symboliques Modaux et Implicites, *Deuxièmes Journées Symboliques-Numériques*, Université d'Orsay, LRI, 127-139.

[59] ———————— (1990), Knowledge Representation and Symbolic Data Analysis. En: Schader M., Gaul, W. Ed., *Knowledege, Data and Computer-Assisted Decisions.* Nato Asi Series, **F.16**, Springer Verlag.

[60] ———————— (1991), Des Objets de l'Analyse de Données à ceux de l'Analyse des Connaissances, En: Kodratoff, Y., Diday, E. Eds. *Induction Symbolique et Numérique à partir de Données*, Cépaduès, Toulouse, 9-75.

[61] ———————— (1993a), *An introduction to Symbolic Data Analysis*, Tutorial of the 4^{th} conference of IFCS, Rapport INRIA **1936**, Roquencourt.

[62] ———————— (1993b), *Quelques Aspects de l'Analyse des Données Symboliques*, Rapport INRIA **1937,** Roquencourt.

[63] ———————— (1995a), Probabilist, Possibilist and Belief Objects, *Annals of Operations Research*, **55,** 227-276.

[64] ———————— (1995c), From Data to Knowledge: Boolean, Probabilist, Possibilist and Belief Objects for Symbolic Data Analysis, Rapport INRIA, Roquencourt.

[65] ———————— (1995b), From Data to Knowledge: Probabilistic Objects for a Symbolic Data Analysis, En: *DIMACS Series in Discrete Mathematics and Theoretical Computer Science,* **19**, 65-88, American Mathematical Society.

[66] ———————— (2000), Symbolic Data Analysis and the SODAS Proyect: Purpose, History, Perspective. En: Bock, H.H., Diday, E. Eds., 2000a, 1-23.

[67] **Diday, E. et al.** (1980), *Optimisation en Classification Automatique*, INRIA , Roquencourt.

[68] **Diday, E., Emilion, R.** (1996), Lattices and Capacities in Analysis of Probabilistic Objects, En: Diday, E., Lechevallier, Y., Opitz, O. Eds., *Ordinal and Symbolic Data Analysis (OSDA'95),* Springer-Verlag, *13-30.*

[69] **Diday, E., Lechevallier, Y., Schader, M., Bertrand, P., Burtschy, B., Editores** (1994), *New Approaches in Classification and Data Analysis*, Studies in Classification, Data Analysis, and Knowledge Organization, Springer-Verlag.

[70] **Diday, E., Lemaire, J., Pouget, J., Testu, F.** (1982), *Éléments d'Analyse de Données*, Dunod.

[71] **Dubois, D., Prade, H.** (1978), Operations on Fuzzy Numbers *Int. J. Systems Sci., **9***, 6, 613-626.

[72] ———————— (1984), The Management of Uncertainty in Expert Systems: The Possibilistic Approach, En: Brans, J.P. (Ed.) *Operational Research'84,* Elsevier Science Pub. 223-238.

[73] ———————— (1989+). On the Combination of Evidence in Various Mathematical Frameworks, Université Paul Sabatier, Toulouse.

[74] **Dubois, D., Xiu, M., Prade, H.** (1991), Fuzzy Discrimination Trees. *Fuzzy Engineering toward Human Friendly Systems. Part II Fuzzy Technology. IFES'91,* 250-260.

[75] **Efron, B., Tibshinari, R.J.** (1993), *An introduccion to the Bootstrap*, Chapman and Hall.

[76] **Esposito, F.** (1994), Conceptual Clustering in Structured Domains: A Theory Guided Approach, En: Diday, E. et al., Eds. (1994), 395-404.

[77] **Esposito, F., Malerba, D., Lisi, F.A.** (2000a), Matching Symbolic Objects. En: Bock, H.H., Diday, E. Eds., 2000a, 186-197.

[78] **Esposito, F., Malerba, D., Tamma, V.** (2000b), Dissimilarity Measures for Symbolic Objects, En: Bock, H.H., Diday, E. Eds., 2000a, 165-185.

[79] **Ferraris, J., Gettler-Summa, M., Pardoux, C., Tong, H.** (1995), Knowledge Extraction using Stochastic Matrices: Application to Elaborate Fishing Strategies, en: Diday, E., Lechevallier, Y., Opitz, O., Eds., *Ordinal and Symbolic Data Analysis (OSDA'95),* Springer-Verlag.

[80] **Fisher, W.D.** (1958), On Grouping for Maximum Homogeneity, *Journal American Statistical Association, **53**,* 789-798.

[81] **Friedman, J.H.** (1977), A Recursive Partitioning Decision Rule for Nonparametric Classification, *IEEE Transactions on Computers,* Abril.

[82] **García-Santesmases Martín-Tesorero, J.M.** (1982), *Cuestiones Notables sobre Discriminación en Variables Cualitativas*, Tesis Doctoral, Dpto. de Estadística e Investigación Operativa, Facultad de Ciencias Matemáticas, Universidad Complutense de Madrid.

[83] **Gettler-Summa, M.** (1992), Factorial Axis Interpretation by Symbolic Objects, *3èmes journées Symboliques-Numériques*, LISE-CEREMADE, Université Paris IX-Dauphine, 53-64.

[84] ———————— (1996), Application d'un Générateur de Règles à l'Interprétation Symbolique, *École d'Été Sept. 1996*, Lise-Ceremade, Université de Paris-IX Dauphine.

[85] ———————— (1999), Marking and Generalization by Symbolic Objects in the Symbolic Official Data Analysis Software, *Cahiers Du Ceremade*, **9935**, Université de Paris-IX Dauphine.

[86] **Gettler-Summa, M., Périnel, E., Ferraris, J.** (1994), Automatic Aid to Symbolic Cluster Interpretation, En: Diday, E. et al., Eds. (1994), 405-413.

[87] **Gil Álvarez, P.** (1981), *Teoría Matemática de la Información*, Colección Matemática Actual, ICE, Madrid.

[88] **Gil, P., Pardo, L., Gil, M.A.** (1993), *Matemáticas de la Incertidumbre y de la Información y sus Aplicaciones Estadísticas,* Publicaciones de la Universidad de Oviedo.

[89] **Gowda, K. C., Diday, E.** (1992) Symbolic Clustering using a new Similarity Measure. *IEEE Transactions on Man, System and Cybernetics*, **22**, 2, 368-378.

[90] **Gower, J.C.** (19), Measures of Similarity, Dissimilarity, and Distance.

[91] **Goodman, L.A., Kruskal, W.H.** (1954), Measures of Association for Cross-Classification, *Journal of the American Statistical Association,* **48**, 732-762.

[92] **Goupil, F., Touati, M., Diday, E., Moult, R.** (2000): Processing Census Data from ONS. En: Bock, H.H., Diday, E. Eds., 2000a, 382-385.

[93] **Guo, H., Belfand, S.B.** (1992), Classification Trees with Neural Network Feature Extraction, *IEEE Transactions on Neural Networks,* **3**, 6, 923-933.

[94] **Hart, A.** (1984), Experience in the use of an Inductive System in Knowledge Engineering, En: Bramer, M. (Ed.), *Research and Development in Expert Systems*, Cambridge University Press.

[95] **Hébrail, G.** (1999), *Software User Manual for the From Database to Symbolic Objects CSCI, SODAS Project* (20821 DG34/D-3/300536), *v02.*

[96] **Hugues, M., Griffon, B., Bouveyron, C.** (1970), *Segmentation et Typologie*, Bordas.

[97] **Hunt, E., Marin, J., Stone, P.** (1966), *Experiments in Induction*, Academic Press, Nueva York.

[98] **Iztueta, A., Calvo, P.** (2000): Comparing European Labour Force Survey Results from the Basque Country and Portugal. En: Bock, H.H., Diday, E. Eds., 2000a, 374-381.

[99] **Kaas, G.** (1980), An exploratory technique fro investigating large quantities of categorical data, *Applied Statistics*, **29**, 119-127.

[100] **Kalbfleish, J.** (1979), *Probability and Statistical Inference*, **2**, Springer-Verlag, New-York.

[101] **Klir, G.J., Folger, T.A.** (1988), *Fuzzy Sets, Uncertainty and Information*, Prentice-Hall Int.

[102] **Kononenko, I., Bratko, I.** (1987), Assistant 86: A Knowledge-Elicitation Tool for Sophisticated Users, *Progress in Machine Learning*, Sigma Press.

[103] **Kullback, S.** (1967), *Information Theory and Statistics*, Dover, Nueva York.

[104] ———————— (1981), Kullback Information, En: Johnson, N.L., Kotz, S., *Encyclopedia of Statistical Sciences*, John Willey & Sons, 421-425.

[105] **Kullback, S., Leibler, R.A.** (1951), On Information and Sufficiency, *Ann. Math.-Statistics,* **22**, 79-86.

[106] **Lauro, N.C., Palumbo, F.** (1998), New Approaches to Principal Components Analysis of Interval Data, En: *NTTS'98, New Techniques & Technologies for Statistics*, Estudio Idea Ed. , Nápoles, 295-300.

[107] **Lauro, N.C., Verde, R., Palumbo, F.** (2000), Factorial Methods with Cohesion Constraints on Symbolic Objects, En: H.A.L. Kiers, J.P. Rasson, P.J.F Groenen, M. Shader (Eds.), *Data Analysis, Classification and Related Methods*, Springer Verlag, Heidelberg, 381-386.

[108] **Lerman, I. C., Da Costa, F. P.** (1995), *Methodological Developments in Decision Trees. An Application to Protein Secondary Structure Prediction*, Grupo de Matemática Aplicada, Fac. de Ciências, Univ. do Porto.

[109] **Maher, P.E., St. Clair, D.** (1993), Uncertain Reasoning in an ID3 Machine Learning Framework, *IEEE'93, 7-12.*

[110] **Manago, M.** (1991), KATE: Intégration de Techniques Symboliques et Numériques en Apprentissage, En: Kodratof, Y. Diday, E. Eds., *Induction Symbolique et Numérique à partir des Données, 125-149.*

[111] **Manton, K.G., Woodbury, M.A., Tolley, H.D.** (1996), *Statistical Applications Using Fuzzy Sets,* Willey Series in Probability and Mathematical Statistics, Wiley-Interscience, John Wiley & Sons.

[112] **Messenger, R., Mandel, L.** (1972), A Modal Search Technique for Predictive Nominal Scale Multivariate Analysis, *J. American Statistical Association*, **67**, 768-772.

[113] **Michalski, R.S.** (1969), On the Quasi-minimal Solution of the General Coverin Problem, *Proc. of the fifth International Symposium on Information Processing,* Bled, Yugoslavia, 125-128.

[114] **Michalski, R.S.** (1973), Aqual1-Computer Implementation of a Variable_valued Logic System VL1 and Examples of its Application to Pattern Recognition, *Proc. of the first International Joint Conference on Pattern Recognition,* Washington D.C., 3-17.

[115] **Michalski, R.S.** (1983), A Theory and Methodology of Inductive Learning, En: Michalski, R.S., Carbonell, J.G. y Mitchel, T.M. Eds., *An Artificial Intelligence Approach,* Tioga, Palo Alto, California.

[116] **Michalski, R.S., Larson, J.** (1983), *Incremental Generation of VL_1 Hypotheses: The Underlying Methodology and the Description of the program AQ11,* Technical report ISG 83-5, Urbana, University of Illinois, Computer Science Department.

[117] **Michalski, R.S.,Mozetic, I., Hong, J., Lavrac, N.** (1986), The Multipurpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains, *Proc. of the Fifth National Conference on Artificial Intelligence,* Morgan Kaufmann, Philadelphia.

[118] **Mingers, J.** (1987), Expert Systems - Rule Induction with Statistical Data, *Journal of the Operational Research Society,* **38**, 39-47.

[119] ———————— (1989), An Empirical Comparison of Selection Measures for Decision-Tree Induction, *Machine Learning,* **4**, 2, 319-342.

[120] **Mola, F.**, **Siciliano, R.** (1992), A Two Predictive Splitting Algorithm in Binary Segmentation, En: Dodge, Y., Whittaker, J. Eds., *Computational Statistics,* vol.1, Physica-Verlag, 179-184.

[121] **Morgan, J.**, **Sonquist, J.** (1963), Problems in the Analysis of Survey Data and a Proposal, *J. American Statistical Association,* **58**, 415-434.

[122] **Morineau, A.** (1998), *Software Requierements Specifications / Interface Requirements Specification for Workbench, SODAS Project* (20821 DG34/D-3/300536), *v02.*

[123] ———————— (2000), The SODAS Software Package, En: Bock, H.H., Diday, E. Eds., 2000a, 386-391.

[124] **Morineau, A., Leprince, V.** (1999), *SODAS Software User manual, SODAS Project* (20821 DG34/D-3/300536) v02.

[125] **Muenier, M.** (1997), *Software Quality Program Plan, SODAS Project* (20821 DG34/D-3/300536) v02.

[126] **Niblett, T., Bratko, I.** (1986), *Learning Decision Rules in Noisy Domains,* Technical Report, Turing Institute, Glasgow.

[127] **Noirhomme-Fraiture, M., Rouard, M** (1997), *Style Guide, SODAS Project* (20821 DG34/D-3/300536), *v01.*

[128] ———————— (2000), Visualizing and Editing Symbolic Objects. En: Bock, H.H., Diday, E. Eds., 2000a, 125-138.

[129] **Nuñez, M.** (1991), The Use of Background Knowledge in Decision Tree Induction, *Machine Learning,* **6**, 231-250.

[130] **Peng, X.P, Kandel, A., Wang, P.** (1991), Concepts, Rules, and Fuzzy Reasoning: A Factor Space Approach, *IEEE Transactions on System, Man, and Cybernetics,* **21**, 1, 194-205.

[131] **Périnel, E.** (1996), *Segmentation et Analyse de Données Symboliques. Application á des Données Probabilistes Imprécises*, Thèse de doctorat, U.F.R. Mathématiques de la Décision, Université de Paris-IX Dauphine, Paris.

[132] ———————— (1999), Construire un Arbre de Discrimination binarire à partir de Données Imprécises, *Revue de Statistique Appliquée,* **47** (1), 5-30.

[133] **Périnel, E., Lechevallier, Y.** (2000), Symbolic Discriminant Rules. En: Bock, H.H., Diday, E. Eds., 2000a, 244-265.

[134] **Polaillon, G.** (2000), Pyramidal Classification for Interval Data using Galois Lattice Reduction, En: Bock, H.H., Diday, E. Eds., 2000a, 324-341.

[135] **Polaillon, G., Diday, E.** (1997), *Gallois Lattices of Symbolic Objects*, rapport **9631**, CEREMADE, Université Paris IX-Dauphine.

[136] **Prade, H.** (1985), A Computational Approach to Approximate and Plausible Reasoning with Applications to Expert Systems, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **PAMI-7**, 3, 260-283.

[137] **Quinlan, J.R.** (1979), Discovering Rules by Induction from large collections of Examples: a case study, En: Michie, D.(Ed.), *Expert Systems in the Micro Electronic Age,* Edimburgh University Press.

[138] ———————— (1986a), Induction of Decision Trees, *Machine Learning 1, 81-106. Artificial Intelligence Approach, II.* California, Kaufmann, 149-166.

[139] ———————— (1986b), Symplifying Decision Trees, *MIT AI Memo no.* **930**.

[140] ———————— (1988), Decision Trees and Multivalued Attributes, En: Hayes, J.E., Michie, D. y Richards, J., Eds., *Machine Intelligence,* Oxford University Press.

[141] ———————— (1990), Probabilistic Decision Trees, En: Kodratoff,Y., Michalski,R. Ed. *Machine Learning, an Artificial Intelligence Approach, III*, California, Morgan Kaufmann, 140-152.

[142] **Ramdami, M.** (1994), *Système d'Induction Formelle à base de Connaissances Imprécises*, Thèse, Université Paris VI-Jussieu.

[143] **Rives, J.** (1990), FID3: Fuzzy Induction Decision Tree, *IEEE'90.*

[144] **Rouard, M., Noirhomme-Fraiture, M., Bisdorff, R.** (1998), Utilisation de l'Etoile Zoom en Exploration de Données Statistiques, *KESDA'98*, Eurostat, Luxemburgo.

[145] **Rounds, E.M.** (1980), A Combined Nonparametric Approach to Feature Selection and Binary Decision Tree Design, *Pattern Recognition,* ***12***, 313-317.

[146] **Ruspini, E.H.** (1969), A New Approach to Clustering*, Information and Control,* **15**, 22-32.

[147] ———————— (1990), *Approximate Reasoning: Past, Present, Future*, Technical Note no. 1465, Artificial Intelligence Center, SRI International, Menlo Park, California.

[148] **Safavian, S.R., Landgrebe, D.** (1991), A Survey of Decision Tree Classifier Methodology, *IEEE Transactions on Systems, Man, and Cybernetics*, **21**, 3, 660-674.

[149] **Séchet, E.** (1995), *Une Application de la Logique Floue en Acoustique: L'Évaluation de l'Influence des Conditions Météorologiques sur la Propagation du Son*, Séminaire du Lise-Ceremade, Université de Paris-IX Dauphine.

[150] **Selby, R.W., Porter, A.A.** (1988), Learning from Examples: Generation and Evaluation of Decision Trees for Software Resource Analysis, *IEEE Transactions on Software Engineering*, **14**, 12 1743-1756.

[151] **Sethi, I.K.** (1995), Neural Implementation of Tree Classifiers, *IEEE Transactions on System, Man and Cybernetics*, **25**, 8, 1243-1249.

[152] **Shafer, G.** (1976), *A Mathematical Theory of Evidence,* Princeton University Press.

[153] **Shannon, C.E.** (1948), A Mathematical Theory of Communication, *Bell System Technical Journal*, **27**, 379-423, 623-656.

[154] **Shlien, S.** (1990), Multiple Binary Decision Tree Classifiers, *Pattern Recognition*, **23**, 7, 757-763.

[155] **Shlien, S.** (1992), Nonparametric Classification using Matched Binary Decision Trees, *Pattern Recognition Letters*, **13**, 83-87.

[156] **Smadhi, S.** (1994), Towards Extraction Method of Knowledge founded by Symbolic Objects, En: Diday, E. et al., Eds. (1994), 430-437.

[157] **Snedecor, G.W., Cochran, W.G.** (1980), *Statistical Methods*, 7^{th} Ed., Iowa State University Press.

[158] **Sokal, R., Fohlf, F.** (1981), *Biometry*, Freeman, San Francisco.

[159] **Spangler, S., Fayyad, U.M., Uthurusamy, R.** (1988), Induction of Decision Trees from Inconclusive Data, 146-150.

[160] **Stéphan, V.** (1996), Description de Classes par des Assertions, *École d'Été Sept. 1996*, Lise-Ceremade, Université de Paris-IX Dauphine.

[161] **Stéphan, V., Hébrail, G., Lechevallier, Y.** (2000): Generation of Symbolic Objects from Relational Data Bases. En: Bock, H.H., Diday, E. Eds., 2000a, 78-105.

[162] **Titterington, D.M., Smith, A.F.M., Makov, U.E.** (1985), *Statistical Analysis of Finite Mixture Distributions,* Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics, John Wiley & Sons.

[163] **Trillas, E., Ansina, C., Terricabras, J.M.** (1995), *Introducción a la Lógica Borrosa*, Ariel, Barcelona.

[164] **Trillas, E., Sanchís, C.** (1979), Sobre Entropías de Conjuntos Borrosos deducidas de Métricas, *Estadística Española,* **82** y **83**.

[165] **Verde, R.** (1995), Fuzzy Binary Segmentation, Dipartimento di Matematica e Stadistica, Università di Napoli "Federico II".

[166] **Verde, R., Carvalho, F.A.T., Lechevallier, Y.** (2000), A Dynamical Clustering Algorithm for Multi-nominal Data, En: H.A.L. Kiers, J.P. Rasson, P.J.F, Groenen, M. Shader (Eds.), *Data Analysis, Classification and Related Methods*, Springer Verlag, Heidelberg, 387-393.

[167] **Weber, S.** (1983), A General Concept on Fuzzy Connectives, Negations and Implicantios based on T-norms and T-conorms, *Fuzzy Sets and Systems*, **11**, 115-134.

[168] **Wu,** (1993), Inductive learning

[169] **Yager, R.R.** (1980), On the Measure of Fuzziness and Negation II. Lattices, *Information and Control*, **44**, 236-260.

[170] **Yuan, Y., Shaw, M.J.** (1995), Induction of Fuzzy Decision Trees, *Fuzzy Sets and Systems*, **69**, 2, 125-139.

[171] **Zadeh, L.A.** (1965), Fuzzy Sets, *Information Control*, **8**, 338-353.

[172] ———————— (1968), Probability Measures of Fuzzy Events, *J. Mathematical Analysis and Applications*, **23**, 421-427.

[173] ———————— (1978), Fuzzy Set as a basis for a Theory of Possibility, *Fuzzy Sets and Systems*, **1**, 1, 3-28.

[174] **Zeidler, J., Scholosser, M.** (1994), Fuzzy Handling of Continuous-Valued Attributes in Decision Trees, *41-46*.

Análisis de Segmentación en el Análisis de Datos Simbólicos

María del Carmen Bravo Llatas

Fe de erratas (enero 2002)

Leyenda: P. : página, p.: párrafo, L.: línea.

P. 8, L. 6 la referencia correcta es Bock y Diday (Bock y Diday, 2000a).

P. 18, en el punto 2., L. 3, se debe añadir el pie de página a la palabra distribución: En esta Memoria, la identificación de las cifras decimales en los números se especifica por el punto decimal en lugar de por la coma.

P. 33, en (1.33) y (1.34) se debe sustituir y por z.

P. 41, el identificador (1.43) debe sustituirse por (1.44).

P. 44, L. 14, debe sustituirse $[d_j \mathcal{R} d'_j]$ por $[d_j \mathcal{R}_j d'_j]$.

P. 45, en (1.54) debe sustituirse $[d_j \mathcal{R} d'_j]$ por $[d_j \mathcal{R}_j d'_j]$. En (1.55), debe sustituirse qp por p.

P. 57, en la definición 1.22, al texto las *aserciones a y b son equivalentes*, debe añadirse *(en E)*. Además, en (1.88) debe sustituirse ω por e.

P. 58, L. 6, se debe eliminar la palabra también, y se debe sustituir de dicha intención por la descripción de dicha intención.

P. 59, p. penúltimo, los textos $1mujer$ y $1hombre$ deben sustituirse por $mujer1$ y $hombre1$, respectivamente.

P. 60, L. 11, al texto "puede ser" debe añadirse ", por ejemplo".

P. 61, L. penúltima, debe sustituirse (??) por (1.98).

P. 62, L. 7, debe sustituirse *edad* por *altura*.

P. 62, L. 11, la referencia correcta es Dempster (Dempster, 1967, 1968).

P. 65, en (1.106) debe sustituirse $[longitud \leq 25]$ por $[longitud \leq 20]$.

P. 72. El segundo párrafo debe ser sustituido por: "Esta proposición se verifica

en el caso particular de que los conjuntos de descripciones sean univariantes $\mathcal{D}'= \mathcal{Y}$ y $\mathcal{D} = \mathcal{P}(\mathcal{Y})$ o $\mathcal{D} = \mathcal{Y}$, y la relación de dominio $\mathcal{R}$ sea la de pertenencia."

P. 74, nota de pie de página: el tercer término de la igualdad a_1 debe ser $\wedge_{i=1,..,p}[Y_i\mathcal{R}_i d_i]$, a_2 es $a_2 = [Y\mathcal{R}'d'] = \wedge_{i=1,..,p}[Y_i\mathcal{R}'_i d'_i]$. En la última línea, el segundo término $\mathcal{R}_i$ se debe sustituir por $\mathcal{R}'_i$. Además, debe añadirse al final de la nota: con $c(.)$ una función definida para pares de niveles de relación.

P. 78, L. 10 en la secuencia de referencias debe añadirse: Brito, 1991.

P. 82, L. 3 debe sustituirse $[X^Z|X^Y]$ por $[X^Y|X^Z]$.

P. 112, la fórmula identificada por (2.36) debe sustituirse por: $U(q_1, ..., q_s) = \frac{1}{s}\sum_{i=1,..,s} \max_{e\in E}(q_i(e))$.

P. 112, en (2.37) el último c_1 debe ser c_s.

P. 136, L. 5, debe sustituirse función de combinación de niveles de relación por función $g(.)$ de combinación de niveles de relación.

P. 138, p. 4, se debe añadir: Antecedentes de aserciones ponderadas pueden encontrarse en Diday (1990, 1991) con los objetos simbólicos modales exteriores, que son ilustrados en (1.100a). En el caso de ponderaciones no booleanas, la generalización (3.8) tiene sentido si se proporciona una definición de suma de aserciones ponderadas (más adelante, en (4.115) se proporciona una definición).

P. 139, p. 4., L. 3, normal debe sustituirse por partición tradicional.

P. 161, debe añadirse un último párrafo: Se define la función g(.) de combinación de niveles de relación en $\mathcal{B}$ (de (3.2)) como la función producto.

P. 162, la condición 2. debe comenzar por: $\min\{Card(Ext_\Omega(\beta_r \wedge b \wedge \mu_r)), Card(Ext_\Omega(\beta_r \wedge b^c \wedge \mu_r))\} < \nu$. En la última línea, se debe sustituir $Ext_\Omega(\beta_k \wedge \mu_k)$ por $Card(Ext_\Omega(\beta_k \wedge \mu_k))$.

P. 162, la condición 5. debe eliminarse.

P. 165, L. penúltima, se debe sustituir (4.11) por (4.12).

P. 171, en (4.34) el segundo término de la equivalencia debe sustitutirse por $Card(Ext_\Omega(\beta_k \wedge [M = i])) < \tau, i \in S^k$.

P. 172, se debe añadir al final de la página: se define la función g(.) de combinación de niveles de relación en $\mathcal{B}$ (de (3.2)) como la función producto.

P. 173, L. 3 desde abajo, debe sustituirse (1.93) por (1.94).

P. 178, L. 6, se debe sustituir elememos por elementos.

P. 196, se debe incluir al final del primer apartado la frase: Se tiene $\sum_{k=1,...,K} w_k^i = 1$, $\forall i = 1, ..., m$. Y, al final del segundo apartado: Se tiene $\sum_{i=1,...,m} wa_i^k = 1$, $\forall k = 1, ..., K$.

P. 209, al final del primer apartado debe añadirse el párrafo: El caso particular de $R_2(.)$ en (4.115) con q_ω obtenida a partir de la función auxiliar de (4.108) supone la definición de la función suma en (3.8) como:

$$\begin{aligned} R(\omega) &= \sum_{k=1,...,K} \beta_k \wedge \mu_k(\omega)\alpha_k = \sum_{k=1,...,K} \beta_k \wedge \mu_k(\omega)[Z \sim q_k] \quad (4.115b) \\ &= [Z \sim \sum_{k=1,...,K} \beta_k \wedge \mu_k(\omega) q_k] \end{aligned}$$

P. 212, L.2 desde abajo, se debe sustituir Gelfand por Belfand.

P. 224, L. 5 desde abajo, se debe sustituir (4.43) por (4.44).

P. 230, pie de página, L. 2, se debe sustituir 1993 por 1993a.

P. 233, L. 3, después del punto se debe añadir: Para descripciones b monoevaluadas (intervalos de rango nulo), se establece la relación de pertenencia a las descripciones de las ramas.

P. 251, tabla 4.4, se debe sustituir muy pequeño por menor que 0.01.

P. 252, L. 9, se debe sustituir 9 por 8.

P. 264, en las representaciones de los nodos decisionales, la forma rigurosa de especificar las relaciones para los predictores es mediante el símbolo $\sim$, en lugar del símbolo $=$. Esto hace referencia a todas las variables especificadas salvo a la variable estrato *cae*.

P. 267, en la representación del sector restauración, la forma rigurosa de especificar las relaciones para los predictores es mediante el símbolo $\sim$, en lugar del

símbolo =. Esto hace referencia a todas las variables especificadas.

P. 270, en las representaciones de los nodos decisionales, la forma rigurosa de especificar las relaciones para los predictores es mediante el símbolo $\sim$, en lugar del símbolo =. Esto hace referencia a todas las variables especificadas salvo a la variable estrato *nace*.

P. 271, en la representación del sector *minería*, la forma rigurosa de especificar las relaciones para los predictores es mediante el símbolo $\sim$, en lugar del símbolo =. Esto hace referencia a todas las variables especificadas.

P. 268, L. penúltima, debe sustituirse tercero por cuarto.

P. 270, tabla 4.5, se debe sustituir muy pequeño por menor que 0.01.

P. 285, L. 4 desde el final, se debe sustituir 1999e por 1998.

P. 298, L. 1 debe sustituirse consolidación datos por consolidación de datos.

P. 352, la referencia Bravo, 1999e debe ser Bravo, 1998.

P. 353, se debe añadir la referencia:

Brito, P. (1991), *Analyse de Données Symboliques. Pyramides d'Héritage*, Thèse de doctorat, U.F.R. Mathématiques de la Décision, Université de Paris-IX Dauphine, Paris.

P. 355, se deben añadir las referencias:

Clark, P., Nibblet, T. (1989), The CN2 Induction Algorithm. *Machine Learning*, **3**, 4, 261-284.

Dempster, A.P. (1967), Upper and Lower Probabilities induced by a Multivalued Mapping, *Annals of Mathematical Statistics,* **38**, 325-339.

Dempster, A.P. (1968), A Generalisation of Bayesian Inference, *Journal of the Royal Statistics Society, Series B,* **30**, 205-247.

P. 359, referencia [108], debe sustituirse **F.P.** por **J.F.P..**