

Gabriel Fiol Roig

Sistema Inteligente Autónomo

para el Aprendizaje Incremental
de Árboles de Decisión



Gabriel Fiol Roig

Sistema Inteligente Autónomo

para el Aprendizaje Incremental
de Árboles de Decisión



Editor en jefe

Prof. Dra. Antonella Carvalho de Oliveira

Editor ejecutivo

Natalia Oliveira Scheffer

Asistente editorial

Flávia Roberta Barão

Bibliotecario

Janaina Ramos

Diseño gráfico

Nataly Evilin Gayde

Thamires Camili Gayde

Vilmar Linhares de Lara Junior

Imágenes de portada

iStock

Edición de arte

Yago Raphael Massuqueto Rocha

2025 por Atena Editora

Copyright © 2025 Atena Editora

Copyright del texto © 2025, el autor

Copyright © 2025, Atena Editora

Los derechos de esta edición han sido cedidos a Atena Editora por el autor.

Publicación en acceso abierto de Atena Editora



El contenido íntegro de este libro está sujeto a la licencia Creative Commons Reconocimiento 4.0 Internacional (CC BY 4.0).

Atena Editora mantiene un firme compromiso con la integridad editorial en todas las etapas del proceso de publicación, garantizando el estricto cumplimiento de las normas éticas y académicas. Adopta políticas para prevenir y combatir prácticas como el plagio, la manipulación o falsificación de datos y resultados, así como cualquier interferencia indebida de intereses financieros o institucionales. Cualquier sospecha de mala conducta científica se trata con la máxima seriedad y se investigará de acuerdo con las normas más estrictas de rigor académico, transparencia y ética.

El contenido de la obra y sus datos, en términos de forma, corrección y fiabilidad, son de exclusiva responsabilidad del autor y no representan necesariamente la posición oficial de Atena Editora. Se permite descargar, compartir, adaptar y reutilizar esta obra para cualquier propósito, siempre que se atribuya la autoría y se haga referencia al editor, de acuerdo con los términos de la Licencia Creative Commons Reconocimiento 4.0 Internacional (CC BY 4.0).

Los artículos nacionales fueron sometidos a una revisión ciega por pares por parte de miembros del Consejo Editorial de la editorial, mientras que los internacionales fueron evaluados por árbitros externos. Todos fueron aprobados para su publicación con arreglo a criterios de neutralidad e imparcialidad académicas.

Sistema inteligente autónomo para el aprendizaje incremental de árboles de decisión

Autor: Gabriel Fiol Roig
Revisión: El autor
Diagramación: Nataly Evilin Gayde
Portada: Yago Raphael Massuqueto Rocha
Indexación: Amanda Kelly da Costa Veiga

Datos de catalogación en publicación internacional (CIP)

R741	Roig, Gabriel Fiol Sistema inteligente autónomo para el aprendizaje incremental de árboles de decisión / Gabriel Fiol Roig. – Ponta Grossa - PR: Atena, 2025.
	Formato: PDF Requisitos del sistema: Adobe Acrobat Reader Modo de acceso: World Wide Web Incluye bibliografía ISBN 978-65-258-3009-4 DOI: https://doi.org/10.22533/at.ed.094250406
	1. Aprendizaje automático. I. Roig, Gabriel Fiol. II. Título.
	CDD 006.31
Preparado por Bibliotecario Janaina Ramos – CRB-8/9166	

Editorial Atena

Ponta Grossa - Paraná - Brasil
+55 (42) 3323-5493
+55 (42) 99955-2866
www.atenaeditora.com.br
contato@atenaeditora.com.br

DECLARACIÓN DEL AUTOR

A efectos de la presente declaración, el término "autor" se utiliza de forma neutra, sin distinción de género ni de número, a menos que se indique lo contrario. Asimismo, el término "obra" se refiere a cualquier versión o formato de creación literaria, incluidos, entre otros, artículos, libros electrónicos, contenidos en línea, de acceso abierto, impresos y comercializados, independientemente del número de títulos o volúmenes. El autor de esta obra declara, a todos los efectos, que 1. no tiene ningún interés comercial que constituya un conflicto de intereses en relación con la publicación; 2. ha participado activamente en la elaboración del trabajo; 3. el contenido está libre de datos y/o resultados fraudulentos, se ha informado debidamente de todas las fuentes de financiación y se han citado y referenciado correctamente los datos e interpretaciones procedentes de otras investigaciones; 4. no tiene ningún interés comercial que constituya un conflicto de intereses en relación con la publicación. Autoriza plenamente la edición y publicación, incluyendo los registros legales, la producción visual y gráfica, así como el lanzamiento y la difusión, de acuerdo con los criterios de Atena Editora; 5. declara ser consciente de que la publicación será de acceso abierto, y podrá ser compartida, almacenada y puesta a disposición en repositorios digitales, de acuerdo con los términos de la Licencia Internacional Creative Commons Reconocimiento 4.0 (CC BY 4.0). 6. se responsabiliza plenamente del contenido del trabajo, incluida la originalidad, la veracidad de la información, las opiniones expresadas y cualquier implicación legal derivada de la publicación.

DECLARACIÓN DEL EDITOR

Atena Editora declara, a todos los efectos legales, que: 1. esta publicación está bajo licencia Creative Commons Attribution 4.0 International Licence (CC BY 4.0), que permite copiar, distribuir, exhibir, ejecutar, adaptar y crear obras derivadas para cualquier propósito, incluso comercial, siempre que se dé el debido crédito al autor o autores y a la editorial. Esta licencia sustituye la lógica de cesión exclusiva de los derechos de autor prevista en la Ley 9610/98, aplicando los principios del acceso abierto; 2. Los autores conservan íntegramente los derechos de autor y se les anima a difundir la obra en repositorios institucionales y plataformas digitales, siempre con la debida atribución de autoría y referencia a la editorial, de acuerdo con los términos de CC BY 4.0; 3. La editorial se reserva el derecho de poner a disposición la publicación en su sitio web, app y otras plataformas, así como de vender ejemplares impresos o digitales, cuando proceda. En el caso de comercialización directa (a través de librerías, distribuidores o plataformas colaboradoras), la cesión de los derechos de autor se realizará de acuerdo con las condiciones establecidas en un contrato específico entre las partes; 4. De acuerdo con la Ley General de Protección de Datos (LGPD), la editorial no cede, comercializa ni autoriza el uso de los datos personales de los autores para fines que no estén directamente relacionados con la difusión de esta obra y su proceso editorial.

Consejo Editorial

Ciencias Exactas y de la Tierra e Ingeniería

Prof^a Dr^a Alana Maria Cerqueira de Oliveira – Instituto Federal do Acre

Prof. Dr. Carlos Eduardo Sanches de Andrade – Universidade Federal de Goiás

Prof^a Dr^a Cristina Aledi Felsemburgh – Universidade Federal do Oeste do Pará

Prof. Dr. Diogo Peixoto Cordova – Universidade Federal do Pampa, Campus Caçapava do Sul

Prof. Dr. Douglas Gonçalves da Silva – Universidade Estadual do Sudoeste da Bahia

Prof. Dr. Eloi Rufato Junior – Universidade Tecnológica Federal do Paraná

Prof^a Dr^a Érica de Melo Azevedo – Instituto Federal do Rio de Janeiro

Prof. Dr. Fabrício Menezes Ramos – Instituto Federal do Pará

Prof. Dr. Fabrício Moraes de Almeida – Universidade Federal de Rondônia

Prof^a Dr^a Glécilla Colombelli de Souza Nunes – Universidade Estadual de Maringá

Prof. Dr. Hauster Maximiler Campos de Paula – Universidade Federal de Viçosa

Prof^a Dr^a Iara Margolis Ribeiro – Universidade Federal de Pernambuco

Prof^a Dr^a Jéssica Barbosa da Silva do Nascimento – Universidade Estadual de Santa Cruz

Prof^a Dr^a Jéssica Verger Nardeli – Universidade Estadual Paulista Júlio de Mesquita Filho

Prof. Dr. Juliano Carlo Rufino de Freitas – Universidade Federal de Campina Grande

Prof. Dr. Leonardo França da Silva – Universidade Federal de Viçosa

Prof^a Dr^a Luciana do Nascimento Mendes – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Prof. Dr. Marcelo Marques – Universidade Estadual de Maringá

Prof. Dr. Marco Aurélio Kistemann Junior – Universidade Federal de Juiz de Fora

Prof. Dr. Marcos Vinicius Winckler Caldeira – Universidade Federal do Espírito Santo

Prof^a Dr^a Maria Iaponeide Fernandes Macêdo – Universidade do Estado do Rio de Janeiro

Prof^a Dr^a Maria José de Holanda Leite – Universidade Federal de Alagoas

Prof^a Dr^a Mariana Natale Fiorelli Fabiche – Universidade Estadual de Maringá

Prof^a Dr^a Natiéli Piovesan – Instituto Federal do Rio Grande do Norte

Prof^a Dr^a Neiva Maria de Almeida – Universidade Federal da Paraíba

Prof^a Dr^a Priscila Natasha Kinas – Universidade do Estado de Santa Catarina

Prof^a Dr^a Priscila Tessmer Scaglioni – Universidade Federal de Pelotas

Prof. Dr. Rafael Pacheco dos Santos – Universidade do Estado de Santa Catarina

Prof. Dr. Ramiro Picoli Nippes – Universidade Estadual de Maringá

Prof^a Dr^a Regina Célia da Silva Barros Allil – Universidade Federal do Rio de Janeiro

Prof. Dr. Sidney Gonçalo de Lima – Universidade Federal do Piauí

A María Magdalena, mi hija, por el sentido que das a mi vida. A mis padres, Jaime y Magdalena, cuyo esfuerzo y determinación me han colmado de valor.

Siempre habrá un lugar para vosotros en mi corazón.

1. INTRODUCCIÓN.....	3
2. MODELO DE UN SISTEMA AUTÓNOMO DE APRENDIZAJE INDUCTO-DEDUCTIVO A PARTIR DE EJEMPLOS.....	7
3. EL APRENDIZAJE FEEDBACK-1	11
4. EL ENTORNO	13
4.1 Tablas Objeto-Atributo completamente especificadas	13
4.2 Tablas Objeto-Atributo incompletamente especificadas.....	15
5. EL DISPOSITIVO DE ADQUISICIÓN DE CONOCIMIENTO (D.A.C)....	22
5.1 Inferencia inductiva sobre Tablas Objeto-Atributo: Bases de atributos.....	22
5.2 Tablas Objeto Atributo como conjuntos de reglas.....	26
5.3 Inferencia inductiva sobre conjuntos de reglas.....	28
6. LA BASE DE CONOCIMIENTOS -HIPÓTESIS/B.C.-	31
6.1 Árboles de Decisión para el aprendizaje	31
6.2 Aprendizaje de Árboles de Decisión.....	33
6.3 Ejemplos	37
6.4 Conclusiones	72
7. EL APRENDIZAJE FEEDBACK-2.....	75
7.1 Las experiencias como consecuencia de la toma de decisiones.....	75
7.2 Implantación de experiencias: algunos casos prácticos.....	78
7.3 Conclusiones del aprendizaje Feedback-2	93
REFERENCIAS	96

PRÓLOGO

Mis inicios en la investigación en Inteligencia Artificial tuvieron lugar en el año 1985 y se centraron en el tema de la inducción a partir de ejemplos clasificados. De ahí surgió mi proyecto de tesis doctoral. La investigación en el tema ya tenía un dilatado recorrido en aquel momento, con un amplio marco conceptual desarrollado. Desde entonces mi dedicación a la investigación en el campo ha sido continua, sin desatender otras áreas por las que me he sentido atraído.

El contenido del libro es totalmente original y se basa en los casi cuarenta años de investigación en el campo. Las referencias bibliográficas del libro provienen en su mayoría de aportaciones propias durante mi dilatada trayectoria, progresos que han ido abriendo y despejando caminos hacia el objetivo final: «comprender el significado del concepto de aprendizaje inductivo autónomo». Ello ha supuesto una gran experiencia, de la que me siento agradecido por las numerosas influencias recibidas por aportaciones de otros autores, cuya mención no aparece en el apartado de las referencias por dos razones principales: para centrar la atención del autor en la perspectiva desde la que se concibe y proyecta la tarea del aprendizaje y por la extensa cantidad de trabajos que se habrían mencionado en las referencias.

A pesar de la gran influencia actual de la inducción a partir de ejemplos en algunas de las áreas relacionadas con el tratamiento de datos en Inteligencia Artificial, no existe todavía una íntegra clarividencia acerca de algunos de los principales conceptos en el campo. Tal es el caso de las populares nociones de «razonamiento» y «aprendizaje», frecuentemente confundidas. Lo dicho provoca, por una parte, un vacío formal sobre el que edificar proyectos conceptualmente sólidos, y por otra una falta de consenso entre expertos y profesores, lo cual se traslada a los estudiantes.

Este libro pretende ofrecer al lector una perspectiva formal del aprendizaje autónomo a partir de ejemplos, a través del diseño de un modelo de aprendizaje inducto-deductivo incremental. Para ello, en los capítulos 1, 2 y 3 se desarrolla un marco conceptual formal de las nociones que sustentan un proyecto de aprendizaje, entre las que destacan los conceptos de «razonamiento», «aprendizaje» y «modelo de aprendizaje». Los posteriores capítulos –4, 5, 6 y 7- describen las características de las componentes de un hipotético modelo de aprendizaje autónomo.

Entre algunas de las **aportaciones** del modelo de aprendizaje que se presenta en el libro cabe destacar:

- i. La posibilidad de considerar **conocimiento incompletamente especificado** en los ejemplos que activan el proceso inductivo y su tratamiento sin necesidad de filtrarlos.
- ii. La capacidad de un **aprendizaje** «sin fin», es decir, **ilimitado** en el tiempo. De esta forma, el conocimiento cualitativo de la hipótesis de aprendizaje progresá –es entrenada- continuamente, enriqueciendo «siempre» sus cualidades descriptivas.

iii. La consideración de **experiencias** durante todo el proceso de aprendizaje. Estas proporcionan una guía en el progreso de la hipótesis de aprendizaje y pueden provenir del sistema, también llamadas experiencias internas, o del usuario experto, estas últimas conocidas como experiencias externas.

iv. **Generalidad** del **proceso inductivo** encargado de generar la hipótesis de aprendizaje. El diseño del modelo de aprendizaje no se limita a un único procedimiento, si no que permite incluir diferentes métodos inductivos que permitan mejorar la hipótesis.

v. Gran capacidad de **autonomía** del **modelo**. Únicamente las experiencias no programables originadas por el usuario experto pueden limitar los márgenes de la dependencia externa del modelo.

Es de destacar los numerosos **puntos de investigación abiertos** que se derivan del trabajo, tanto a nivel teórico como práctico. En particular, destacan:

- La posibilidad de extender el proyecto con otros tipos de hipótesis inductivas diferentes a las que se describen en el libro.
- El tratamiento de atributos mediante mecanismos de codificación binaria o discretización de sus valores. Esta cuestión incide sobre aspectos tales como la existencia de superabundancia de información de la hipótesis y la eficiencia del procedimiento inductivo.
- La creación de un sistema de pesos sobre los atributos a través de los cuales vienen descritos los ejemplos, de manera que el propio sistema sea capaz, a través de las experiencias, ajustar automáticamente dichos pesos con objeto de mejorar la hipótesis.
- Creación de un catálogo estándar de experiencias internas y externas, con objeto de facilitar el diseño del dispositivo encargado de su generación.
- La solución de problemas complejos a través de auténticas jerarquías de hipótesis. En ocasiones la complejidad de los problemas exige un diseño –generalmente top-down– de la solución a través de jerarquías de hipótesis con dependencias establecidas.
- Desarrollo de una plataforma software para el desarrollo práctico de aplicaciones basadas en el modelo de aprendizaje inductivo que se presenta en el libro.

Podría añadirse a la lista anterior una nutrida cantidad de puntos de investigación más específicos con los que continuar edificando el modelo de aprendizaje que se presenta.

El texto está orientado tanto a investigadores con experiencia en el campo, como a profesores y alumnos de grado o master con una formación previa en inteligencia artificial.

1. INTRODUCCIÓN

En ciertos universos complejos se considera que el ambiente adopta, en cada momento, una determinada **situación** [12], la cual puede verse como una especie de «estado complejo», difícil de manejar en el ámbito de un **agente basado en metas –ABM-**.

Una situación describe las características del ambiente en un momento considerado, haciendo uso para ello de una cantidad ingente de conocimiento –**base de conocimientos**-. La cantidad de situaciones es, en general, muy elevada, y en muchos problemas resulta imposible de predecir.

La solución de los problemas se obtiene a partir de transiciones entre situaciones a través de la toma de decisiones –acciones- hasta dar con una situación objetivo.

No todo el conocimiento relativo a una situación está explicitado –es decir, el agente no tiene acceso directo al mismo- en la base de conocimientos. Esta es una cuestión muy importante, pues para llevar a cabo de forma segura y adecuada la transición entre situaciones, el agente posiblemente deberá descubrir conocimiento no explícito en su base de conocimientos –esto es, conocimiento que, a pesar de no estar explicitado, está implícito en La base de conocimientos-. El proceso de llevar a cabo la explicitación de conocimiento se conoce como **razonamiento**.

La figura 1.1 ilustra el razonamiento como un proceso de transición entre situaciones.

Razonamiento

Los agentes que razonan constituyen uno de los tipos más «completos» de agentes en cuanto a capacidades cognoscitivas –sólo superados por los agentes que aprenden, pues éstos precisan de la capacidad de razonar para aprender.

El razonamiento constituye una de las más potentes herramientas de explotación del conocimiento sobre las que se sustenta la inteligencia humana.

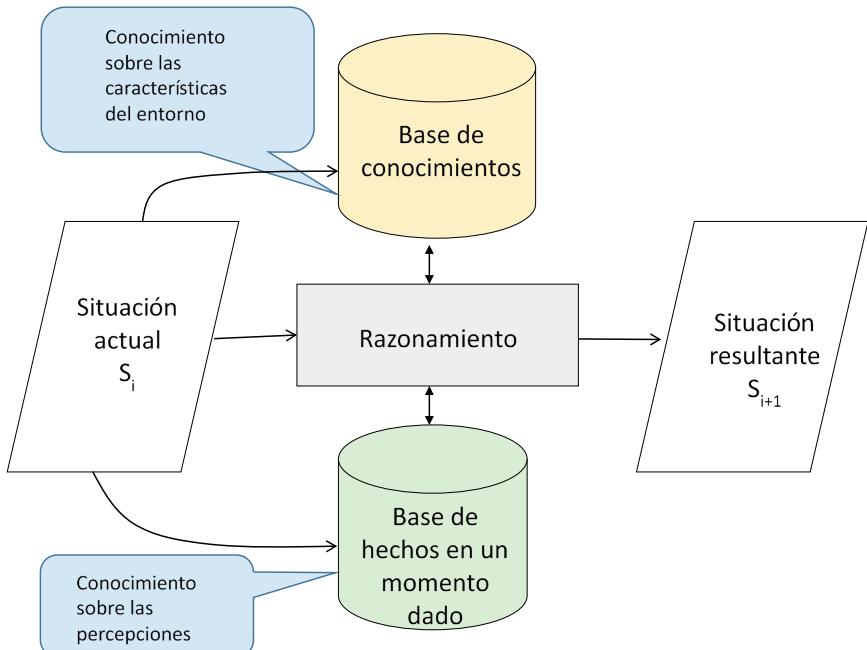


Figura 1.1. Razonamiento como proceso de transición entre situaciones.

El razonamiento siempre se lleva a cabo sobre un conocimiento considerado. No hay razonamiento sin conocimiento. Así pues, el razonamiento va directamente asociado a una representación particular del conocimiento.

El razonamiento es un **proceso inferencial** basado en algún lenguaje –el lenguaje usado para representar el conocimiento–.

Un proceso inferencial pretende obtener conocimientos implícitos en una base de conocimientos a partir de los conocimientos explícitos en la misma.

Para poder implantar un proceso de razonamiento mediante un lenguaje, éste debe permitir:

- Representar conocimientos acerca de un entorno en el cual el agente razonará –*Base de conocimientos –B.C.–*.
- Ofrecer unas «normas» que permitan obtener conocimientos nuevos –inferidos– expresados en el lenguaje dado, a partir de conocimientos existentes representados en el mismo lenguaje.

Resumiendo, el **razonamiento** es un proceso mediante el cual un sistema explora una base de conocimientos, con objeto de alcanzar unos objetivos o conclusiones mediante la extracción/generación –adquisición– de conocimiento relevante sobre la situación actual del mundo –B.C..

Así pues, un agente que razona explora un universo de conocimientos –la **B.C.**– con objeto de mantener actualizada la descripción –**situación**– del mundo –la propia **B.C.**– de

manera que le permita decidir las acciones a tomar.

Durante un proceso de razonamiento se infieren expresiones del lenguaje –esto es, conocimientos nuevos- que no formaban parte de la base de conocimientos original y que se añaden a la misma.

Los **agentes que razonan** de este modo también se conocen como **agentes basados en conocimiento –ABC-**.

La figura 1.2 ilustra la relación entre un proceso razonamiento, el lenguaje de la representación del conocimiento, el mundo real y la base de conocimientos o mundo virtual.

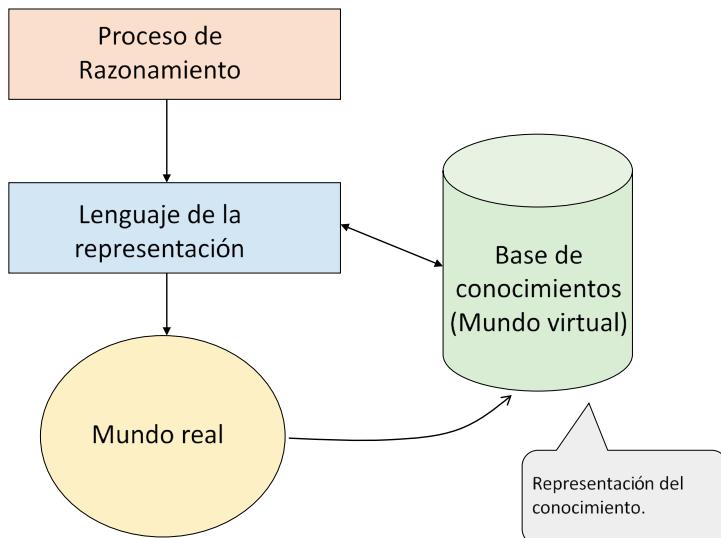


Figura 1.2. Relación entre representación del conocimiento, lenguaje de la representación, mundo real y razonamiento.

De acuerdo con la figura 1.2, un **proceso de razonamiento** –o simplemente razonamiento- se basa en un **lenguaje de representación del conocimiento**, el cual se usa para representar **el mundo real** –base de conocimientos- y para llevar a cabo las inferencias del **razonamiento** con objeto de alcanzar los objetivos del agente.

Principales tipos de razonamiento.

- Razonamiento **deductivo**.

Proceso a través del cual se pretende alcanzar conocimientos específicos sobre alguna entidad a partir de conocimientos generales sobre la misma y sobre un entorno al cual dicha entidad pertenece. El más importante es el *razonamiento lógico*. También está el razonamiento lógico con incertezas.

- Razonamiento **inductivo**, también llamado **adquisición inductiva de conocimiento**.

Proceso basado en obtener conclusiones o representaciones generales/abstractas –reglas, árboles de decisión, grafos, etc- sobre alguna entidad a partir de representaciones específicas –partículas o elementos concretos de conocimiento- sobre dicha entidad y sobre el entorno al cual dicha entidad pertenece. Uno de los tipos de razonamiento inductivo más estudiados, es el *razonamiento a partir de ejemplos clasificados*.

- Razonamiento **inducto-deductivo**.

Proceso basado en obtener conclusiones o representaciones generales –reglas, árboles de decisión, grafos, etc- sobre alguna entidad a partir de representaciones generales y específicas –esto es, partículas o elementos concretos de conocimiento y a su vez representaciones abstractas- de dicha entidad y sobre el entorno al cual dicha entidad pertenece.

Es el tipo de razonamiento más complejo, que sirve de soporte a numerosos sistemas de **aprendizaje**.

Aprendizaje

Proceso mediante el cual un sistema mejora –perfecciona- su función con el tiempo. El aprendizaje de un agente es la capacidad del mismo para converger, a lo largo del tiempo, hacia sus objetivos.

Para que un agente pueda aprender con el tiempo, debe ser capaz de razonar sobre el entorno en todo momento y juntamente con sus experiencias y posiblemente las de algún usuario experto externo, ser capaz de extraer las pertinentes consecuencias –**aprender**-.

Por tanto, el aprendizaje de un agente se basa en las **experiencias** resultantes de su toma de decisiones y en las continuas **observaciones del entorno**.

Este trabajo se centra en el desarrollo de un modelo de aprendizaje autónomo a partir de ejemplos y de las experiencias del agente y posiblemente las de algún usuario experto externo a lo largo del proceso de aprendizaje, cuyo proceder no tiene limitaciones temporales, lo que significa que se trata de un proceso permanentemente abierto a una mejora de la función del agente. En cuanto a la explotación del conocimiento, cabe decir que el modelo que se propone se basa en un razonamiento inducto-deductivo, pues actúa a la vez sobre estructuras de conocimiento específicas y estructuras abstractas con objeto de «mejorar» las propias estructuras abstractas.

2. MODELO DE UN SISTEMA AUTÓNOMO DE APRENDIZAJE INDUCTO-DEDUCTIVO A PARTIR DE EJEMPLOS

Los agentes que aprenden de forma autónoma constituyen el máximo exponente dentro de la categoría de los agentes inteligentes. Tales agentes se caracterizan por su capacidad de mejorar su función a lo largo del tiempo a través de la experiencia –si ésta existe- acumulada y de la observación de nuevos elementos de conocimiento del entorno.

El aprendizaje se concibe como un proceso de **actualización** de una **hipótesis** de acuerdo con nuevas evidencias acerca del entorno y/o las experiencias, de manera que la **hipótesis actualizada** sea la que mejor se ajuste a las experiencias y a las nuevas evidencias. Existen diferentes tipos de aprendizaje, entre los que destacan, entre otros, el aprendizaje por rutina, el aprendizaje a partir de instrucciones, el aprendizaje por analogía y el aprendizaje inductivo. Este último es quizás el más estudiado y tradicionalmente ha sido categorizado principalmente en aprendizaje a partir de ejemplos y aprendizaje a partir de observaciones. Otros autores, no obstante, realizan una taxonomía del mismo en aprendizaje supervisado, aprendizaje no supervisado, aprendizaje semi supervisado y aprendizaje por refuerzo.

El presente trabajo se enmarca en el aprendizaje inducto-deductivo –A.I.D.- autónomo a partir de ejemplos. La dinámica del A.I.D. se basa en la mejora progresiva de la función del agente a medida que van surgiendo nuevos ejemplos y nuevas experiencias sobre la hipótesis. La figura 2.1 ilustra un modelo de un tal sistema de aprendizaje [1, 3].

- **Entorno:** contiene los ejemplos acerca del ambiente. Éstos constituyen porciones específicas/concretas de conocimiento sobre algún concepto. Tales conocimientos vienen descritos extensionalmente.
- **Dispositivo de Adquisición de Conocimiento –D.A.C.-:** es una etapa procedural basada en un razonamiento inducto-deductivo. Se encarga de adquirir los conocimientos relevantes provenientes del Entorno y del *Feedback*. El conocimiento adquirido es abstracto –general- y constituye la hipótesis o base de conocimientos actual.
- **Hipótesis o Base de conocimientos –H.B.C.-:** constituye lo que el agente conoce acerca del ambiente –descripción del ambiente-. Tiene un formato abstracto –intensional-, lo cual facilita el razonamiento deductivo del agente en el logro de sus objetivos. Se va actualizando a lo largo del tiempo. Figura 2.1. Modelo sencillo de un sistema de aprendizaje.

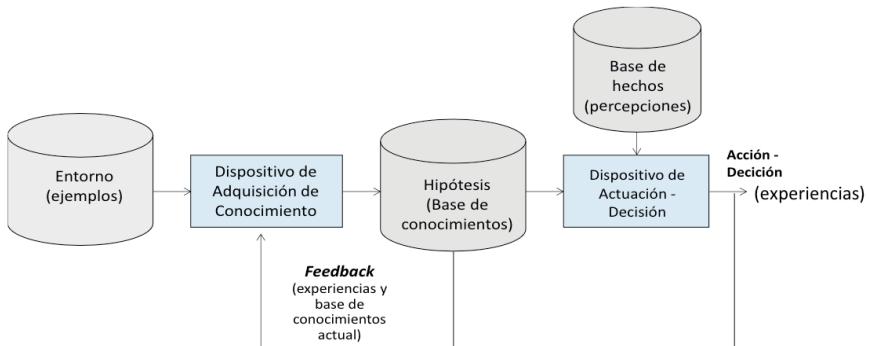


Figura 2.1. Modelo sencillo de un sistema de aprendizaje.

- **Base de hechos –percepciones-**: almacena las percepciones del agente en cada instante con objeto de utilizarlas como «hechos» en el razonamiento deductivo sobre la Hipótesis/B.C.. Se trata de un conocimiento extensional.
- **Dispositivo de Actuación-Decisión**: a partir de las percepciones y de lo que el agente conoce acerca del ambiente –Hipótesis/B.C.-, este dispositivo se encarga de razonar de forma deductiva con objeto de tomar una acción/decisión sobre algún aspecto particular del mundo.
- **Acción-Decisión**: dispositivo encargado de ejecutar las acciones/decisiones con objeto de alcanzar una situación objetivo.
- **Feedback**: retroalimentación basada en las experiencias sobre las acciones/decisiones ejecutadas de acuerdo con la Hipótesis/B.C., con objeto de actualizar/optimizar la propia Hipótesis/B.C.. En general, se trata de conocimiento abstracto.

Como se observa en el modelo de la figura 2.1, el aprendizaje constituye un proceso retroalimentado por la Hipótesis/B.C. o base de conocimientos del agente y por las experiencias de la etapa de Actuación-Decisión.

Se trata de un proceso cíclico, donde cada ciclo constituye un «paso/etapa» del proceso de aprendizaje. En cada ciclo se mejora, si realmente es mejorable –pero en ningún caso se empeora-, la calidad de la base de conocimientos.

Por tanto, un sistema de aprendizaje a partir de ejemplos tiene un carácter incremental. La mejora de la hipótesis es gradual, de manera que en cada ciclo del bucle se incrementa su funcionalidad.

Algunas cuestiones fundamentales al respecto son:

1. La *Hipótesis/B.C.* del proceso de aprendizaje debe ajustarse a criterios bien definidos que dependen de la naturaleza del problema.
2. La *Hipótesis/B.C.* resultante de la actualización de la *Hipótesis/B.C.* actual en cada ciclo de aprendizaje, debe suponer una mejora real de la *Hipótesis/B.C.*

actual. A más conocimiento mejor es la hipótesis. En cualquier caso, la Hipótesis/B.C. resultante de la actualización nunca debe ser peor que la hipótesis actual – nótense que puede darse el caso donde las experiencias y/o los nuevos ejemplos del *Entorno* sean irrelevantes en la mejora de la Hipótesis/B.C. actual, de acuerdo con las características de mejora del criterio considerado-.

3. En cada ciclo de aprendizaje debe considerarse «todo» el conocimiento proporcionado por el *Entorno* hasta el momento. Se trata de una cuestión fundamental, pues de otra forma sólo podría obtenerse una «aproximación» a la Hipótesis/B.C. establecida por el criterio considerado.

4. El *Feedback* de la figura 2.1 puede categorizarse en dos tipos de retroalimentación, tal como se muestra en la figura 2.2: *Feedback-1* y *Feedback-2*. No obstante sólo una de ellas puede llevarse a cabo en un instante dado.

- El ***Feedback-1*** se llevará a cabo cuando el ciclo de aprendizaje requiera la actualización de la B.C. debido, únicamente, a la aparición de nuevos ejemplos desde el entorno, que deben ser procesados. No se tienen en cuenta experiencias en este tipo de feedback. Se trata de una forma de retroalimentación más sencilla, en la que el proceso de aprendizaje automático se guía únicamente a partir del conocimiento que le proporciona el *Entorno*.
- El ***Feedback-2*** se requerirá una vez ejecutado el dispositivo de *Actuación-Decisión* y su función consiste en proporcionar al *Dispositivo de Adquisición de Conocimiento*, el conocimiento sobre las experiencias obtenidas en la toma de decisiones en base a la Hipótesis/B.C. del ciclo considerado. De esta forma, el *DAC* procederá a la actualización de la Hipótesis/B.C. de acuerdo con dichas experiencias y, si se da el caso, también tendrá en cuenta la aparición de nuevos ejemplos desde el *Entorno*.

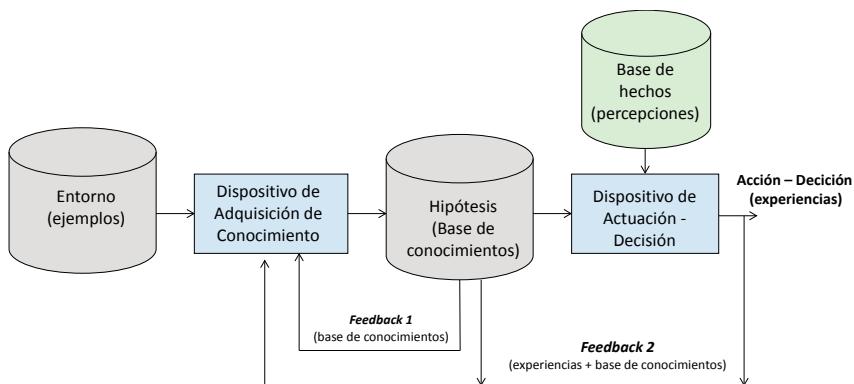


Figura 2.2. Retroalimentación de un sistema de aprendizaje.

Este trabajo se centrará primeramente en el aprendizaje con retroalimentación de tipo *Feedback-1* o simplemente **aprendizaje Feedback-1**. Dicho modelo constituye una forma de aprendizaje incremental cuyos componentes se ilustran en la figura 2.3.

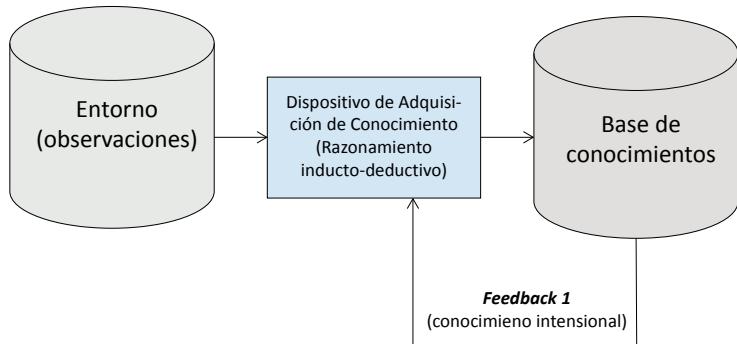


Figura 2.3. Aprendizaje *Feedback-1*.

Posteriormente se discutirán los principales aspectos del **aprendizaje Feedback-2**, especialmente por lo que respecta a los principales tipos de experiencias y a la puesta en práctica de las mismas.

3. EL APRENDIZAJE FEEDBACK-1

Se trata de un modelo de aprendizaje incremental a partir de nuevos ejemplos de conocimiento del entorno. La cantidad de nuevos ejemplos que provocan la ejecución de un ciclo de aprendizaje es arbitraria, con al menos un ejemplo por ciclo, y debe ser asumible computacionalmente –en otro caso el aprendizaje se realizará a partir de cantidades asumibles de ejemplos.

En cada ciclo de aprendizaje, la *Hipótesis/B.C.* debe actualizarse de acuerdo con algún criterio bien definido.

La ***Hipótesis/B.C.*** actualizada en un ciclo de aprendizaje debe preservar las siguientes **propiedades**:

- a. Debe cubrir «todos» los elementos de conocimiento –ejemplos- aparecidos hasta el momento.
- b. Debe disponer de toda la información necesaria sobre los ejemplos para así poder aplicar con todas las garantías el criterio utilizado para su actualización en cada uno de los ciclos de aprendizaje.

¿Por qué resulta interesante este tipo de aprendizaje incremental? ¿Qué ventajas proporciona?

- i. Es una forma natural de aprendizaje. Se apoya en el hecho de que «a partir de una hipótesis y nuevas evidencias, es muy posible que la hipótesis sea mejorable, pero nunca es posible empeorarla».
- ii. Es un aprendizaje acumulativo, es decir, cubre todos los elementos del entorno que han aparecido durante el proceso.
- iii. No siempre están disponibles todos los ejemplos del *Entorno* desde el inicio del proceso de aprendizaje y, en numerosas ocasiones, el experto ni siquiera conoce su número. De esta forma, el sistema actualiza la *Hipótesis/B.C.* a medida que van apareciendo nuevos ejemplos.
- iv. También puede aplicarse en el caso de que todos los elementos del entorno estén disponibles desde el inicio.
- v. Evoluciona a lo largo del tiempo, mejorando siempre que sea posible, la *Hipótesis/B.C.* en cada ciclo.
- vi. No existen límites en cuanto al número de ejemplos del *Entorno* a tratar a lo largo del tiempo.
- vii. Tampoco existen límites en cuanto al número de ciclos necesarios para completar definitivamente la tarea del aprendizaje. Mientras surjan nuevos ejemplos, la tarea de aprendizaje se ejecutará.
- viii. La complejidad computacional en cada ciclo de aprendizaje es asumible, siempre y cuando el número de atributos que describen los ejemplos sea computacionalmente «razonable».

ix. Permite aprender a partir de ejemplos con información incompleta. De hecho, no es necesario indicarle al sistema si los ejemplos contienen o no información incompleta, él mismo los identifica. Tampoco es necesario ningún sistema de filtros con objeto de realizar un pretratamiento de ejemplos incompletamente especificados.

x. El dispositivo de *Adquisición de Conocimiento* se encarga de mejorar la *Hipótesis/B.C.* a partir de la propia *Hipótesis/B.C.* –actual- y nuevos ejemplos. Para ello se basa en un razonamiento de tipo inducto-deductivo sobre estructuras específicas –los ejemplos- y abstractas –la *Hipótesis/B.C.*-. Como se describió en la sección 1, este tipo de razonamiento constituye un proceso basado en obtener conclusiones o representaciones generales –reglas, árboles de decisión, grafos, etc.- sobre alguna/s entidad/es –llamadas conceptos- a partir de representaciones **generales** y **específicas** de dicha/s entidad/es. Se trata de un proceso de extracción de conocimiento cualitativo y constituye el tipo de razonamiento más complejo.

En las siguientes secciones se describen las características de cada componente del modelo de aprendizaje Feedback-1.

4. EL ENTORNO

El entorno, también conocido como *Tabla Objeto-Atributo*, abreviado OAT, constituye el conocimiento de entrada del *Dispositivo de Adquisición de Conocimiento* a partir de ejemplos [3, 5, 9]. Se trata de conocimiento específico, expresado mediante porciones concretas, llamadas también ejemplos acerca de algunos conceptos. En otros trabajos [2] las OATs se conocen como *Sistemas de Información*. Las OATs descritas en esta sección han sido formuladas únicamente para atributos con dominios discretos. En [11] se presenta una extensión para el tratamiento de OATs con atributos continuos.

En esta sección se describen formalmente los conceptos básicos relacionados con las OATs. En la subsección 4.1 se presenta el concepto de OAT completamente especificada, en la que no existen ejemplos incompletamente especificados o vagos. Se trata del caso más sencillo. Mientras que la subsección 4.2 se dedica a OATs incompletamente especificadas. Es el caso más complejo, que permite además unificar ambos tipos de OAT en un único concepto.

4.1 Tablas Objeto-Atributo completamente especificadas

Definición 1 (Tablas Objeto-Atributo completamente especificadas). Una OAT completamente especificada o simplemente OAT, es una 7-tupla, $OAT = \langle D, R, V, F, C, G, f \rangle$, donde:

$D = \{d_1, d_2, \dots, d_m\}$ es un conjunto de ejemplos o porciones concretas de conocimiento.

$R = \{r_1, r_2, \dots, r_n\}$ es un conjunto de cualidades o atributos en términos de los cuales se describen los ejemplos de D .

$V = \{V_1, V_2, \dots, V_n\}$ es un conjunto formado por los conjuntos de valores que adoptan los atributos r_i , $i=1\dots n$, en la OAT, siendo V_i el conjunto de valores que puede adoptar el atributo r_i en la OAT. Un tal conjunto V_i se llama el dominio de valores o simplemente dominio del atributo r_i .

$F = \{f_1, f_2, \dots, f_n\}$ es un conjunto de funciones que definen los valores que adoptan los elementos de D respecto a las cualidades r_i , $i = 1, 2, \dots, n$. Esto es, $f_i : D \times \{r_i\} \rightarrow V_i$ son las funciones especificadoras de los valores que adopta cada ejemplo respecto a cada uno de los atributos.

C es un conjunto de w conceptos, $C = \{C_1, C_2, \dots, C_w\}$.

f es una función que asigna a cada elemento de D sus correspondientes conceptos, esto es, $f : D \rightarrow (C)$, donde $\Pi(C)$ denota el conjunto de partes de C .

Finalmente, $G = \{g_1, g_2, \dots, g_n\}$ es un conjunto de funciones que definen, para cada valor del dominio de los atributos, el/los conceptos asociados al valor; esto es, $g_i : V_i \rightarrow \Pi(C)$, de manera que si $t_p \in V_i$, $g_i(t_p) = \bigcup_{j=1\dots m} f(d_j)$, tal que $f_i(d_j, r_i) = t_p$.

Gráficamente, una OAT puede ilustrarse mediante la tabla de la figura 4.1, donde $t_i^k \in V_k$, $1 \leq k \leq n$, $1 \leq i \leq m$, es el valor del atributo r_k asociado al ejemplo d_i a través de la función f_k ; $\{C_i, C_j\} \in \Pi(C)$, es el elemento del conjunto de partes de C asociado al elemento d_i a través de la función f , esto es, $f(d_i) = \{C_i, C_j\}$. Obsérvese como cada fila de la OAT contiene la descripción completa de un ejemplo o elemento de D , la cual puede representarse a través de una $n+2$ tupla tal como $(d_i, t_i^1, t_i^2, \dots, t_i^n, \{C_i, C_j\})$.

D\R	r_1	r_2	...	r_k	...	r_n	$\Pi(C)$
d_1	t_1^1	t_1^2	...			t_1^n	$\{C_a, C_b, C_c\}$
d_2	t_2^1	t_2^2	...			t_2^n	$\{C_p\}$
...
d_i	t_i^1	t_i^2	...			t_i^n	$\{C_i, C_j\}$
...
d_m	t_m^1	t_m^2	...			t_m^n	$\{C_q\}$

Figura 4.1. Ilustración gráfica de una OAT.

Sea $R_x = \{r_a, r_b, \dots, r_c\}$, $R_x \subseteq R$, un subconjunto de atributos de R . La parte de una $n+2$ tupla de la OAT formada únicamente por valores de los atributos de R_x se llama una **tupla atributo-valor** de R_x . Así por ejemplo, a partir de la $n+2$ tupla $(d_i, t_i^1, t_i^2, \dots, t_i^n, \{C_i, C_j\})$, la tupla atributo-valor de R_x es $(t_i^a, t_i^b, \dots, t_i^n)$.

La incompletitud del conocimiento sobre los valores de los atributos de una OAT definirá las características de la propia OAT. Así, si los valores de todos los atributos de R son conocidos con total exactitud, diremos que la OAT en cuestión está completamente especificada. Si por el contrario para algún elemento $d_i \in D$ el valor de algún atributo de R no viene completamente especificado, diremos que la OAT está incompletamente especificada.

Ejemplo_1.

Considérese la OAT de la figura 4.2, donde, de acuerdo con la definición 1 tenemos que:

D\R	r ₁	r ₂	r ₃	r ₄	$\Pi(C)$
d ₁	0	rojo	1	0	{C _a }
d ₂	0	verde	0	1	{C _b }
d ₃	1	rojo	0	1	{C _b }
d ₄	0	azul	0	0	{C _d }
d ₅	1	azul	2	0	{C _a }
d ₆	0	rojo	2	0	{C _a }
d ₇	1	azul	1	1	{C _a }

Figura 4.2. Ilustración gráfica de la OAT del ejemplo_1.

$$D = \{d_1, d_2, d_3, d_4, d_5, d_6, d_7\}.$$

$$R = \{r_1, r_2, r_3, r_4\}.$$

$$V = \{V_1, V_2, V_3, V_4\}, \text{ con:}$$

$$V_1 = \{0, 1\},$$

$$V_2 = \{\text{rojo, verde, azul}\}, \quad V_3 = \{0, 1, 2\}, \quad V_4 = \{0, 1\}.$$

$$F = \{f_1, f_2, f_3, f_4\}, \text{ con:}$$

$$f_1 : D \times \{r_1\} \rightarrow V_1, \quad f_2 : D \times \{r_2\} \rightarrow V_2,$$

$$f_3 : D \times \{r_3\} \rightarrow V_3,$$

$$f_4 : D \times \{r_4\} \rightarrow V_4.$$

Por ejemplo, $f_1(d_1, r_1) = 0$, $f_1(d_1, r_2) = \text{rojo}$, $f_1(d_2, r_2) = \text{verde}$, $f_2(d_5, r_2) = \text{azul}$, $f_4(d_3, r_4) = 1$.

$$C = \{C_a, C_b, C_d\}.$$

f: D → $\Pi(C)$ produce las siguientes asignaciones:

$$f(d_1) = \{C_a\}, \quad f(d_2) = \{C_b\}, \quad f(d_3) = \{C_b\}, \quad f(d_4) = \{C_d\}, \quad f(d_5) = \{C_a\}, \quad f(d_6) = \{C_d\}, \quad f(d_7) = \{C_a\}.$$

Finalmente, G = {g₁, g₂, g₃, g₄}, con:

$$g_1 : V_1 \rightarrow \Pi(C), \quad g_2 : V_2 \rightarrow \Pi(C), \quad g_3 : V_3 \rightarrow \Pi(C), \quad g_4 : V_4 \rightarrow \Pi(C), \text{ de manera que:}$$

$$g_1(0) = \{C_a, C_b, C_d\}, \quad g_1(1) = \{C_a, C_b\};$$

$$g_2(\text{rojo}) = \{C_a, C_b, C_d\}, \quad g_2(\text{verde}) = \{C_b\}, \quad g_2(\text{azul}) = \{C_a, C_d\};$$

$$g_3(0) = \{C_b, C_d\}, \quad g_3(1) = \{C_a\}, \quad g_3(2) = \{C_a, C_d\};$$

$$g_4(0) = \{C_a, C_d\}, \quad g_4(1) = \{C_a, C_b\};$$

Sea R_x = {r₁, r₃} un subconjunto de atributos de la OAT de la figura 4.2. Entonces (r₁=0, r₃=1), (r₁=0, r₃=0), (r₁=1, r₃=0), (r₁=1, r₃=2), (r₁=0, r₃=2), (r₁=1, r₃=1), son todas las posibles tuplas atributo-valor de R_x.

4.2 Tablas Objeto-Atributo incompletamente especificadas

Las características que determinan el grado de incompletitud del conocimiento proporcionado por una OAT, vienen determinadas por la precisión con la que se describen los valores de los dominios de los atributos. Así, si r_j ∈ R es un atributo de una OAT cuyo

dominio discreto es V_j , son múltiples las formas de describir de manera incompleta/vaga el valor de r_j en una situación dada.

La manera más imprecisa de describir el valor de r_j en una situación específica consiste, evidentemente, en no discernir ninguna posibilidad acerca del valor real de r_j en la situación considerada. Si por el contrario el valor real de r_j es descrito a través de la consideración de varias posibilidades o valores de su dominio, entonces nos encontramos ante un tipo de incompletitud de conocimiento menos vaga que la que acaba de considerarse, puesto que la cantidad de conocimiento del que se dispone sobre el valor del atributo es mayor [10]. En el otro extremo, nos encontramos con el caso en que el valor de r_i sea totalmente conocido, esto es, corresponda a un único valor de su dominio, V_i , en cuyo caso no existe ningún tipo de vaguedad o imprecisión en la descripción de r_i .

A continuación se formaliza el concepto de conocimiento incompleto sobre los valores de un atributo de la OAT. Con objeto de facilitar la discusión y a la vez por causas de extensión del trabajo, sólo se considerarán atributos con dominios discretos [10]. En [11] se investiga el caso de incompletitud con dominios de valores continuos.

Considérese una OAT dada, y sea $r_j \in R$ un atributo cuyo dominio de valores es discreto.

Definición 2 (c_valores). El valor de un atributo r_j en una situación específica d_i puede representarse mediante un subconjunto $V_{di,rj}$, $V_{di,rj} = \{v_{j1}^i, v_{j2}^i, \dots, v_{jp}^i\}$, siendo $v_{j1}^i, v_{j2}^i, \dots, v_{jp}^i$ los diferentes valores simples adoptados por r_j en la situación d_i . Si el número de valores de $V_{di,rj}$ es mayor que 1, es decir $\#V_{di,rj} > 1$ -donde el símbolo $\#$ representa el cardinal de $V_{di,rj}$ -, entonces se dice que r_j es un *atributo incompletamente especificado/vago*. Si $\#V_{di,rj} = 1$ entonces r_j es un *atributo completamente especificado*. $\#V_{di,rj}$ nunca puede ser menor que 1. Si $V_{di,rj}$, $V_{di,rj} = \{v_{j1}^i, v_{j2}^i, \dots, v_{jp}^i\}$, es un valor vago del atributo r_j -es decir, $\#V_{di,rj} > 1$ - en la situación específica d_i , entonces $V_{di,rj}$ también se representará mediante el símbolo $*_{j1,j2,\dots,jp}$.

El valor $V_{di,rj}$ puede interpretarse de diferentes formas según el caso en cuestión. Así por ejemplo, podría interpretarse como la disyunción lógica de los valores simples que contiene; en otras palabras, si $V_{di,rj} = \{v_{ji}, v_{jk}, \dots, v_{js}\}$ entonces se dice que el atributo r_j adopta uno de los valores incluídos en $V_{di,rj}$, lo cual puede escribirse también como $V_{di,rj} = v_{ji} \text{ OR } v_{jk} \text{ OR } \dots \text{ OR } v_{js}$, siendo OR la función lógica del mismo nombre. Nótese que en el caso que los atributos adopten un único valor en toda situación real específica, entonces la OR debe reemplazarse por una OR-exclusiva –OREX-. Un tal valor $V_{di,rj}$ se denomina un *c_valor*. También podría darse el caso de una interpretación conjuntiva de los valores simples, esto es, $V_{di,rj} = v_{ji} \text{ AND } v_{jk} \text{ AND } \dots \text{ AND } v_{js}$, etc.

Según la definición 2, observemos que si un atributo r_j es vago en alguna situación específica d_i , entonces r_j es definitivamente vago -es decir, r_j se considera simplemente un atributo vago-. Si no existe ninguna situación específica d_i en la que r_j sea vago, entonces r_j es definitivamente un atributo completamente especificado.

La definición 2 supone el punto de partida en la formalización del concepto de conocimiento incompleto respecto de los valores de los atributos de una OAT. Obsérvese, sin embargo, que nada se ha dicho acerca de las características de los valores simples, v_{ji} , de los atributos, por tanto, éstos pueden considerarse arbitrarios, y así también los c_valores, los cuales resultarán de gran utilidad para representar aquellas situaciones en las que no se conozca con precisión el valor de los atributos asociados a los elementos de D.

A partir de aquí, el valor de un atributo para un elemento considerado se interpretará como un c_valor, de acuerdo con la definición 2.

Definición 3 (c_dominio de un atributo). El c_dominio, V_i , del atributo r_i , abarca los valores que puede tomar dicho atributo en todas las situaciones específicas observadas hasta el momento actual. Esto significa que:

- Como V_i está formado por c_valores de r_i , entonces V_i es un conjunto de subconjuntos, ya que, según la definición 2, cada valor viene representado por un conjunto.
- Como V_i está formado por los valores de r_i observados hasta «el momento actual», entonces V_i puede evolucionar con el tiempo, ya que posiblemente se observen nuevos valores de r_i en el futuro.

Definición 4 (peso de los c_valores). Sea $V_{di,rj}$ el c_valor de r_j en una situación determinada. Se define el peso de $V_{di,rj}$, representado por $w(V_{di,rj})$, como la cantidad de valores susceptibles de ser adoptados por r_j en la situación considerada; esto es, $w(V_{di,rj}) = \#V_{di,rj}$.

De acuerdo con la definición 4, el c_valor $V_{di,rj}$ de un atributo r_j es tanto más vago cuanto mayor sea su peso.

Definición 5 (nueva representación para los c_valores vagos). Sea $V_{di,rj} = \{v_{ji}, v_{jk}, \dots, v_{jp}\}$ un c_valor vago del dominio V_j –esto es, con $w(V_{di,rj}) > 1$ – en una situación específica. Entonces $V_{di,rj}$ se representará también mediante el símbolo $*_{i,k,\dots,p}^j$.

La figura 4.3 ilustra gráficamente una OAT donde se dan situaciones de vaguedad en la descripción de los valores de algunos de los atributos de R.

D R	r_1	r_2	...	r_k	...	r_n	$\Pi(C)$
d_1	$\{t_1^1\}$	$\{t_1^2\}$...			$\{t_1^n\}$	$\{C_a, C_b, C_c\}$
d_2	$\{t_2^1\}$	$\{t_{1,3}^2\}$...			$\{t_2^n\}$	$\{C_p\}$
.
d_i	$\{t_{2,4,6}^1\}$	$\{t_i^2\}$...			$\{t_i^n\}$	$\{C_i, C_j\}$
.							.
d_m	$\{t_m^1\}$	$\{t_m^2\}$...			$\{t_m^n\}$	$\{C_q\}$

Figura 4.3. OAT con c_valores vagos.

Obsérvese como $t_{2,4,6}^1$ es un valor vago asociado al elemento d_i respecto al atributo r_1 , y $t_{1,3}^2$ un valor vago asociado a d_2 respecto al atributo r_2 . Obsérvese también como los c_valores completamente especificados de los atributos se representan por elementos de los dominios encerrados entre paréntesis, tal como se describe en la definición 2. Por tanto, la diferencia entre las OATs de las figuras 4.1 y 4.3 radica en que ahora los valores de los atributos son conjuntos de valores –c_valores-.

Tanto en las OATs completamente especificadas como en OATs incompletamente especificadas, todo ejemplo $d_j \in D$ puede interpretarse como una regla antecedente-consecuente, donde el antecedente está formado por los pares atributo-valor de d_j y el consecuente por los conceptos asociados al ejemplo. Así, el ejemplo d_i de la figura 4.3 podría representarse como:

$$(r_1 = t_{2,4,6}^1) \wedge (r_2 = t_i^2) \wedge \dots \wedge (r_n = t_i^n) \rightarrow \{C_i, C_j\}$$

Definición 7 (Tablas Objeto-Atributo completa e incompletamente especificadas). Esta definición extrapola el concepto de OAT completamente especificada de la definición 1, al de OAT incompletamente especificada, con la intención de unificar dichos conceptos en ambos tipos de OATs. Recuérdese que los conceptos **CIS_OAT** y **OAT** son sinónimos. Ahora, una Tabla Objeto-Atributo, completa o incompletamente especificada, abreviada **OAT**, se define como:

$OAT = < D, R, W, V, V', F, H, Q, C, f >$, donde:

$D = \{d_1, d_2, \dots, d_m\}$ es un conjunto de ejemplos o porciones concretas de conocimiento.

$R = \{r_1, r_2, \dots, r_n\}$ es un conjunto de cualidades o atributos en términos de los cuales se describen los ejemplos de D .

$W = \{W_1, W_2, \dots, W_n\}$ es un conjunto de dominios de valores simples de los atributos, $r_i, r_i \in R, i=1 \dots n$, siendo, $W_j, W_j \in W$, el dominio de valores simples del atributo r_j en la OAT. Un dominio simple está formado por todos los posibles valores simples de los atributos.

$V = \{V_1, V_2, \dots, V_n\}$ es un conjunto formado por los dominios de cada atributo, $r_i, r_i \in R$, $i=1 \dots n$, siendo, $V_j = \{v_{1,rj}, v_{2,rj}, \dots, v_{p,rj}\}$, $V_j \in V$, el dominio del atributo r_j en la OAT. Nótese que los valores de un dominio V_j son todos diferentes y adoptan la forma $v_{k,rj}$, $v_{k,rj} \in V_j$. Segundo la definición 2, cada valor $v_{k,rj} \in V_j$ es un subconjunto de valores simples de W_j .

C es un conjunto de w conceptos, $C = \{C_1, C_2, \dots, C_w\}$, de forma que cada C_i representa un concepto.

f es una función que asigna a cada elemento de D los conceptos correspondientes, es decir, $f: D \rightarrow \Pi(C)$, donde $\Pi(C)$ denota el conjunto de partes de C .

V' , $V' = \{V'_1, V'_2, \dots, V'_n, V'_{1,2}, \dots, V'_{i,j,\dots,k}, \dots, V'_{1,2,\dots,n}\}$, representa el conjunto de dominios de todas las tuplas de valores de atributos de la OAT correspondientes a todos los subconjuntos de atributos de R . Es decir, $V'_{i,j,\dots,k}$, con $i=1 \dots n, j=1 \dots n, \dots k=1 \dots n, i \neq j \neq \dots \neq k$, representa el dominio de las tuplas de valores de atributos de la OAT correspondientes al subconjunto de atributos $\{r_i, r_j, \dots, r_k\}$. Nótese cómo el conjunto V' constituye una extensión de V a todos los subconjuntos de atributos de la OAT. El conjunto V' contiene $2^n - 1$ elementos - es decir, dominios de tuplas de valores de atributos.

$F = \{f_1, f_2, \dots, f_n\}$, $f_i: D \times \{r_i\} \rightarrow V_i$, $i=1 \dots n$, es un conjunto de funciones que definen los valores que adoptan los elementos de D para cada atributo r_i , $i=1, 2, \dots, n$.

$H = \{h_1, h_2, \dots, h_n\}$ es un conjunto de funciones, una para cada atributo r_j de la OAT, tal que h_j , $j=1 \dots n$, asocia el subconjunto correspondiente de conceptos a cada valor simple v_{jk} , $v_{jk} \in V_{w,rj}$, $V_{w,rj} = \{v_{j1}, v_{j2}, \dots, v_{jp}\}$, $v_{w,rj} \in V_j$. Es decir, $h_j: W_j \rightarrow \Pi(C)$, de modo que $h_j(v_{jk}) = f(d_1) \cup f(d_2) \cup \dots \cup f(d_m)$ para todo d_s , $s=1 \dots m$, tal que $v_{jk} \in f_j(d_s, r_j)$.

$Q = \{q_1, q_2, \dots, q_n\}$ es un conjunto de funciones, una por cada atributo r_i de la OAT. Sea $v_{k,rj} = \{v_{j1}, v_{j2}, \dots, v_{jp}\}$, $v_{k,rj} \in V_j$, un valor del atributo r_j , donde v_{jk} , $k=1 \dots p$, son sus valores simples. Las funciones del conjunto Q toman la forma $q_j: V_j \rightarrow \Pi(\Pi(C))$, de modo que cada q_j , $j=1 \dots n$, asocia el conjunto correspondiente de subconjuntos de conceptos de C a cada valor diferente $v_{k,rj}$, $v_{k,rj} \in V_j$, siendo $q_j(v_{k,rj}) = \{h_j(v_{j1}), h_j(v_{j2}), \dots, h_j(v_{jp})\}$. Nótese que la expresión $\{h_j(v_{j1}), h_j(v_{j2}), \dots, h_j(v_{jp})\}$ representa un conjunto de subconjuntos de conceptos, uno para cada valor simple, v_{jk} , $v_{jk} \in V_{k,rj}$.

No será necesario distinguir si una OAT está completamente o incompletamente especificada, puesto que toda OAT completamente especificada puede describirse mediante una OAT incompletamente especificada. A partir de aquí, se considerará que el término OAT se refiere a una OAT.

Ejemplo_2.

Considérese la OAT de la figura 4.4, la cual contiene ejemplos incompletamente especificados. De acuerdo con la definición 7 tenemos: $OAT = < D, R, W, V, V', F, H, Q, C, f >$, donde:

$$D = \{d_1, d_2, d_3, d_4, d_5, d_6, d_7\}.$$

$$R = \{r_1, r_2, r_3, r_4\}.$$

$$C = \{ C_a, C_b, C_d \}.$$

Los valores que adoptan los atributos respecto a cada ejemplo en la OAT se conocen como *c_valores* y se encierran entre paréntesis, puesto que un *c_valor* se considera un conjunto de valores. Obsérvese como el elemento d_3 adopta, respecto al atributo r_1 , el *c_valor* $V_{d3,r1} = \{^{*1}_{0,1}\}$, el cual también puede representarse como $V_{d3,r1} = \{0, 1\}$. Dicho *c_valor* es vago, pues $\# V_{d3,r1} > 1$. Lo mismo ocurre con los *c_valores* $V_{d6,r3} = \{0, 1, 2\}$ y $V_{d4,r4} = \{0, 1\}$. El resto de *c_valores* de la OAT están completamente especificados.

La columna derecha de la tabla, representada por el símbolo $\Pi(C)$, representa los conceptos de *C* asociados a cada elemento $d_i \in D$.

D\R	r_1	r_2	r_3	r_4	$\Pi(C)$
d_1	{0}	{rojo}	{1}	{0}	{ C_a }
d_2	{0}	{verde}	{0}	{1}	{ C_b }
d_3	$\{^{*1}_{0,1}\}$	{rojo}	{0}	{1}	{ C_b }
d_4	{0}	{azul}	{0}	$\{^{*4}_{0,1}\}$	{ C_d }
d_5	{1}	{azul}	{2}	{0}	{ C_a }
d_6	{0}	{rojo}	$\{^{*3}_{0,1,2}\}$	{1}	{ C_b }
d_7	{1}	{azul}	{1}	{1}	{ C_a }

Figura 4.4. OAT del ejemplo_2.

$V = \{ V_1, V_2, V_3, V_4 \}$, representa el conjunto de *c_dominios* de los atributos de *R*, siendo V_i el *c_dominio* del atributo r_i . Así pues, tenemos que:

$$V_1 = \{ \{0\}, \{1\}, \{^{*1}_{0,1}\} \} = \{ \{0\}, \{1\}, \{0, 1\} \},$$

$$V_2 = \{ \{rojo\}, \{verde\}, \{azul\} \},$$

$$V_3 = \{ \{0\}, \{1\}, \{2\}, \{^{*3}_{0,1,2}\} \} = \{ \{0\}, \{1\}, \{2\}, \{0, 1, 2\} \},$$

$$V_4 = \{ \{0\}, \{1\}, \{^{*4}_{0,1}\} \} = \{ \{0\}, \{1\}, \{0, 1\} \}$$

$W = \{W_1, W_2, \dots, W_n\}$ es un conjunto de dominios de valores simples de los atributos, $r_i, r_i \in R, i=1 \dots n$, siendo, $W_j, W_j \in W$, el dominio de valores simples del atributo r_j en la OAT. Por lo tanto:

$$W_1 = \{0, 1\},$$

$$W_2 = \{rojo, verde, azul\},$$

$$W_3 = \{0, 1, 2\},$$

$$W_4 = \{0, 1\}.$$

$V' = \{V'_1, V'_2, V'_3, V'_4, V'_{1,2}, V'_{1,3}, V'_{1,4}, V'_{2,3}, V'_{2,4}, V'_{3,4}, V'_{1,2,3}, V'_{1,2,4}, V'_{1,3,4}, V'_{2,3,4}, V'_{1,2,3,4}\}$, representa el conjunto de dominios de tuplas atributo-valor de la OAT para todos los subconjuntos de atributos de *R*.

Algunos ejemplos de dichos dominios son:

$$V'_1 = \{<0>, <1>, <{}^*_0, 1>\},$$

$$V'_{1,2} = \{<0, \text{rojo}>, <0, \text{verde}>, <{}^*_0, 1, \text{rojo}>, <0, \text{azul}>, <1, \text{azul}>\},$$

$$V'_{2,3,4} = \{<\text{rojo}, 1, 0>, <\text{verde}, 0, 1>, <\text{rojo}, 0, 1>, <\text{azul}, 0, {}^*_0, 1>, <\text{azul}, 2, 0>, <\text{rojo},$$

$${}^*_0, 1, 2>, <\text{azul}, 1, 1>\}.$$

$F = \{f_1, f_2, \dots, f_n\}$, $f_i: D \times \{r_i\} \rightarrow V_i$, $i=1 \dots n$, es un conjunto de funciones que definen los valores que adoptan los elementos de D para cada atributo r_i , $i=1, 2, \dots, n$.

Ejemplos:

$$f_1(d_1, r_1) = \{0\}, f_1(d_2, r_1) = \{0\}, f_1(d_3, r_1) = \{{}^*_0, 1\};$$

$$f_2(d_1, r_2) = \{\text{rojo}\}; f_2(d_2, r_2) = \{\text{verde}\}; f_2(d_3, r_2) = \{\text{rojo}\}$$

$H = \{h_1, h_2, \dots, h_n\}$ es un conjunto de funciones, una para cada atributo r_j de la OAT, tal que h_j , $j=1..n$, asocia el subconjunto correspondiente de conceptos a cada valor simple v_{jk} , $v_{jk} \in V_{w,rj}$, $V_{w,rj} = \{v_{j1}, v_{j2}, \dots, v_{jp}\}$, $v_{w,rj} \in V_j$. Es decir, $h_j: W_j \rightarrow \Pi(C)$, de modo que $h_j(v_{jk}) = f(d_1) \cup f(d_2) \cup \dots \cup f(d_m)$ para todo d_s , $s=1..m$, tal que $v_{jk} \in f_j(d_s, r_j)$.

Ejemplos:

Considérese el atributo r_1 . Se tiene que $W_1 = \{0, 1\}$.

$$\text{Entonces: } h_1(0) = \{C_a\} \cup \{C_b\} \cup \{C_c\} \cup \{C_d\} \cup \{C_e\} = \{C_a, C_b, C_d\};$$

$$h_1(1) = \{C_b\} \cup \{C_a\} \cup \{C_a\} = \{C_a, C_b\}.$$

Considérese ahora el atributo r_3 . Obsérvese como $W_3 = \{0, 1, 2\}$.

$$\text{Entonces: } h_3(0) = \{C_b\} \cup \{C_b\} \cup \{C_d\} \cup \{C_b\} = \{C_b, C_d\};$$

$$h_3(1) = \{C_a\} \cup \{C_b\} \cup \{C_a\} = \{C_a, C_b\};$$

$$h_3(2) = \{C_a\} \cup \{C_b\} = \{C_a, C_b\}.$$

$Q = \{q_1, q_2, \dots, q_n\}$ es un conjunto de funciones $q_j: V_j \rightarrow \Pi(\Pi(C))$, una para cada atributo r_j de la OAT, de manera que q_j , $j=1..n$, asocia el correspondiente conjunto de subconjuntos de conceptos de C a cada valor diferente $v_{k,rj} = \{v_{j1}, v_{j2}, \dots, v_{jp}\}$, $v_{k,rj} \in V_j$, siendo $q_j(v_{k,rj}) = \{h_j(v_{j1}), h_j(v_{j2}), \dots, h_j(v_{jp})\}$.

Ejemplos:

Considérese el dominio $V_1 = \{\{0\}, \{1\}, \{0, 1\}\}$ del atributo r_1 .

$$\text{Entonces: } q_1(\{0\}) = \{h_1(0)\} = \{C_a, C_b, C_d\};$$

$$q_1(\{1\}) = \{h_1(1)\} = \{C_a, C_b\};$$

$$q_1(\{0, 1\}) = \{h_1(0), h_1(1)\} = \{\{C_a, C_b, C_d\}, \{C_a, C_b\}\}.$$

Considérese a continuación el dominio $V_3 = \{\{0\}, \{1\}, \{2\}, \{0, 1, 2\}\}$ del atributo r_3 .

$$\text{Entonces: } q_3(\{0\}) = \{h_3(0)\} = \{C_b, C_d\};$$

$$q_3(\{1\}) = \{h_3(1)\} = \{C_a, C_b\};$$

$$q_3(\{2\}) = \{h_3(2)\} = \{C_a, C_b\};$$

$$q_3(\{0, 1, 2\}) = \{h_3(0), h_3(1), h_3(2)\} = \{\{C_b, C_d\}, \{C_a, C_b\}, \{C_a, C_b\}\}.$$

Por razones de claridad en la presentación de conceptos, la OAT descrita en esta sección ha sido formulada únicamente para atributos con dominios discretos. Una extensión para el caso de la OAT con atributos continuos se presenta en [11].

5. EL DISPOSITIVO DE ADQUISICIÓN DE CONOCIMIENTO (D.A.C)

El D.A.C. se encarga de generar/actualizar la *Hipótesis/BC* en cada ciclo de aprendizaje del sistema, de manera que la hipótesis resultante sea la que más se ajuste al criterio establecido. Se trata de una etapa fundamental del modelo de aprendizaje adoptado.

Recordemos el modelo del sistema de aprendizaje *Feedback-1* de la figura 2.3. La clave para actualizar la BC pasa por definir un método de adquisición de conocimiento que, a partir del razonamiento sobre nuevos ejemplos del *Entorno* y sobre la *Hipótesis/BC*, permita mejorar la funcionalidad de esta última de acuerdo con la norma de calidad impuesta a la propia BC a través del correspondiente criterio.

Dificultades aparentes: el hecho de razonar sobre dos tipos de conocimiento radicalmente distintos, siendo uno muy específico y el otro abstracto, supone, a priori, una elevada complejidad. No obstante, hemos diseñado un método relativamente sencillo para dicha tarea.

Considerando que la BC viene expresada con formato árbol de decisión, entonces puede ser interpretada como un conjunto de reglas antecedente-consecuente, de manera que cada rama del árbol desde la raíz a un nodo hoja constituye una regla.

A su vez, cada ejemplo del *Entorno* puede también ser visto como una regla antecedente-consecuente, tal como se describe al final de la sección 5.2.

Entonces, **actualizar** la *Hipótesis/BC* consiste en «modificarla» coherentemente de acuerdo con las reglas representadas por los nuevos ejemplos del *Entorno* que figuran en el ciclo actual de aprendizaje.

Un concepto fundamental para llevar a cabo la mencionada tarea de actualización de la BC es el de **base de atributos** o simplemente **base**. Una base de atributos es un subconjunto adecuado de atributos de la OAT en términos del cual se creará la BC actualizada. La sección 6.1 describe las características del concepto mencionado.

5.1 Inferencia inductiva sobre Tablas Objeto-Atributo: Bases de atributos

La selección de un subconjunto «adecuado» de atributos, R_x , $R_x \subseteq R$, para describir intensionalmente los subconjuntos o conceptos de C en una OAT, constituye una de las principales etapas del proceso de inferencia inductivo. El término «**adecuado**» significa que **los atributos de R_x deben permitir describir los conceptos de C de acuerdo con las características del criterio considerado**. Tales características, independientemente de la naturaleza del problema particular, deben **garantizar en cualquiera de los casos una descripción nítida de los conceptos**, esto es, ausente de confusiones o contradicciones. Lo dicho supone que todo criterio debe imponer, entre otras posibles consideraciones, la

ausencia de contradicciones en la descripción de los conceptos. Todo subconjunto de atributos R_x , $R_x \subseteq R$, que garantiza una descripción nítida de los conceptos de C, recibe el nombre de **base de atributos** de R respecto a C. Por tanto, gran parte de los esfuerzos computacionales de un proceso inductivo se centrarán en la búsqueda de alguna base de atributos «adecuada», siempre de acuerdo con el criterio considerado.

Documentación completa sobre el concepto de base de atributos para OAT's completamente especificadas puede encontrarse en [3], [5], [9]. A continuación se ofrece una definición del concepto de **base de atributos** a partir de la perspectiva unificada de OATs completamente especificadas y OATs incompletamente especificadas [10].

Definición 8 (pares de c_valores parcialmente comunes). Considérese una OAT dada. Sean $v_{k,rj}, v_{s,rj} \in V_j$, $k \neq s$, dos c_valores del dominio de un mismo atributo r_j . Si se cumple que $V_{k,rj} \cap V_{s,rj} \neq \emptyset$, entonces $v_{k,rj}$ y $v_{s,rj}$ se dice que son un par de c_valores parcialmente comunes. Nótese que los c_valores parcialmente comunes se definen en el mismo dominio.

Definición 9 (c_valores individuales contradictorios). Considérese una OAT dada. Sea $r_j \in R$ un atributo de R con dominio V_j . Sea $v_{k,rj} = \{v_{j1}, v_{j2}, \dots, v_{jw}\}$ un c_valor en el dominio V_j , $v_{k,rj} \in V_j$. Se dice que $v_{k,rj}$ es un c_valor contradictorio si y solo si existe algún $d_r \in D$ con $f_j(d_r, r_j) \cap v_{k,rj} \neq \emptyset$, tal que $f_j(d_r) \subset h_j(v_{js})$, $s=1..w$, para cualquier valor simple $v_{js} \in V_{k,rj}$, $v_{js} \notin f_j(d_r, r_j) \cap v_{k,rj}$ –Obsérvese que el símbolo \subset se refiere a la inclusión estricta de subconjuntos –.

Definición 10 (pares de c_valores contradictorios). Considérese una OAT dada. Sea $r_j \in R$ un atributo de R con dominio V_j . Sean $v_{k,rj} = \{v_{j1}, v_{j2}, \dots, v_{jw}\}$ y $v_{p,rj} = \{v_{j1}, v_{j2}, \dots, v_{js}\}$ dos c_valores del dominio V_j , $v_{k,rj} \in V_j$, $v_{p,rj} \in V_j$, tal que $v_{k,rj} \cap v_{p,rj} \neq \emptyset$. Sean $d_a \in D$ y $d_b \in D$, $a \neq b$, tales que $f_j(d_a, r_j) = v_{k,rj}$ y $f_j(d_b, r_j) = v_{p,rj}$. Si se cumple que $f_j(d_a) \neq f_j(d_b)$ entonces el par $(v_{k,rj}, v_{p,rj})$ se dice que es un par de c_valores contradictorios. En el caso de que $v_{k,rj} = v_{p,rj}$, entonces $v_{k,rj}/v_{p,rj}$ son valores individuales contradictorios -ver definición 5-.

Obsérvese como un par de c_valores contradictorios no garantizan una descripción nítida de los correspondientes conceptos de C. A su vez, un mismo c_valor individual del dominio de un atributo resulta contradictorio si no permite distinguir sin ningún tipo de confusión los conceptos asociados al c_valor en la OAT.

Definición 11 (pares de tuplas atributo-c_valor contradictorias). Considérese una OAT dada. Sean $e_a = <V_{dv,ri}, V_{dv,rj}, \dots, V_{dv,rk}>$, $e_a \in V'_{i,j,\dots,k}$, $V_{dv,rs} \in V_s$, $s=i,j,\dots,k$, y $e_b = <V_{dw,ri}, V_{dw,rj}, \dots, V_{dw,rk}>$, $e_b \in V'_{i,j,\dots,k}$, $V_{dw,rp} \in V_p$, $p=i,j,\dots,k$, $v,w=1..m$, un par de dos tuplas atributo valor definidas sobre el mismo subconjunto de atributos $R_x = \{r_i, r_j, \dots, r_k\}$, $R_x \subseteq R$. Se dice que el par (e_a, e_b) es un par de tuplas contradictorias con respecto al subconjunto de atributos R_x , si y sólo si $[(V_{dv,ri}, V_{dw,ri}), (V_{dv,rj}, V_{dw,rj}), \dots, (V_{dv,rk}, V_{dw,rk})]$ son pares de c_valores parcialmente comunes y $(f_j(d_v) \neq f_j(d_w))$.

La siguiente definición 12 describe el concepto de base de atributos, fundamental en el proceso inductivo, pues pone de relieve las características que el conocimiento sobre los atributos debe satisfacer con objeto de ser considerado «adecuado» para describir intensionalmente los conceptos de la OAT.

Definición 12 (Base de atributos). Considérese una determinada OAT. Sea $R_x = \{r_1, r_2, \dots, r_k\} \subseteq R$, un subconjunto de atributos de la OAT. Se dice que R_x es una base de atributos con respecto a C si, y sólo si, no existe ningún par de tuplas contradictorias en R_x .

El concepto de base de atributos resulta fundamental, puesto que toda base de conocimientos viene expresada en términos de una base de atributos.

D\R	r_1	r_2	r_3	r_4	$\Pi(C)$
d_1	{0}	{rojo}	{1}	{0}	{C _a }
d_2	{0}	{verde}	{0}	{1}	{C _b }
d_3	{* _{0,1} }	{rojo}	{0}	{1}	{C _b }
d_4	{0}	{azul}	{0}	{* _{0,1} }	{C _d }
d_5	{1}	{azul}	{2}	{0}	{C _a }
d_6	{0}	{rojo}	{* _{0,2} }	{1}	{C _b }
d_7	{1}	{azul}	{1}	{1}	{C _a }

Figura 5.1. OAT.

Ejemplo_1.

Con objeto de aclarar los conceptos de esta sección, vamos a aplicarlos a la OAT de la figura 5.1. El objetivo central del ejemplo será la comprensión del concepto de *base de atributos*.

- **Considérese el atributo r_1 .**

- De acuerdo con la figura, los *c_valores* asociados a los elementos $d_i \in D$ respecto del atributo r_1 son:

$$V_{d_1,r_1} = \{0\}; V_{d_2,r_1} = \{0\}; V_{d_3,r_1} = \{{*}_{0,1}\}; V_{d_4,r_1} = \{0\}; V_{d_5,r_1} = \{1\}; V_{d_6,r_1} = \{0\}; V_{d_7,r_1} = \{1\};$$

- El *cDominio* V_1 del atributo r_1 es: $V_1 = \{\{0\}, \{1\}, \{{*}_{0,1}\}\} = \{\{0\}, \{1\}, \{0,1\}\}$

Recuérdese que el *c_valor* $\{{*}_{0,1}\}$ equivale a {0, 1}.

- Según la **definición 8**, los pares de *c_valores* de V_1 , *parcialmente comunes* respecto del atributo r_1 son:

$(\{0\}, \{0\}), (\{1\}, \{{*}_{0,1}\}), (\{0\}, \{{*}_{0,1}\})$ y $(\{1\}, \{{*}_{0,1}\})$, puesto que $\{0\} \cap \{0\} = \{0\}$, $\{0\} \cap \{{*}_{0,1}\} = \{0\}$, $\{1\} \cap \{1\} = \{1\}$ y $\{1\} \cap \{{*}_{0,1}\} = \{1\}$.

- De acuerdo con la **definición 9**, a continuación se describen los valores individuales contradictorios para algunos atributos:

Respecto del atributo r_1 se tienen:

$\{0\} \in V_1$ es un *c_valor individual contradictorio*, puesto que $h_1(0) = \{C_a, C_b, C_d\}$. Para comprobarlo primero escogemos un elemento $d_i \in D$, $i=1 \dots 7$, y veamos si a través de dicho elemento se satisfacen las condiciones de la definición 9. Si escogemos por ejemplo el elemento $d_1 \in D$, se tiene que $f_1(d_1, r_1) \cap \{0\} \neq \emptyset$ y $f(d_1) = \{C_a\}$. Como $f(d_1) \subset h_1(0)$

se concluye que $\{0\}$ es un *c_valor individual contradictorio*.

$\{1\} \in V'_1$ es un *c_valor individual contradictorio*, puesto que $h_1(1) = \{C_a, C_b\}$. Si escogemos por ejemplo el elemento $d_3 \in D$, se tiene que $f_1(d_3, r_1) \cap \{1\} \neq \emptyset$ y $f(d_3) = \{C_b\}$. Como $f(d_3) \subset h_1(1)$ se concluye que el *c_valor* $\{1\}$ es un *c_valor contradictorio*.

$\{\ast^1_{0,1}\} \in V'_1$ es un *c_valor individual contradictorio*. Si por ejemplo escogemos el elemento $d_3 \in D$, se tiene que $f_1(d_3, r_1) \cap \{\ast^1_{0,1}\} = f_1(d_3, r_1) \cap \{0, 1\} = \{\ast^1_{0,1}\} \cap \{0, 1\} = \{0, 1\} \cap \{0, 1\} = \{0, 1\} \neq \emptyset$ y $f(d_3) = \{C_b\}$. Ahora probamos si existe algún valor simple, v_{11} , $v_{11} \in f_1(d_3, r_1) \cap \{\ast^1_{0,1}\}$, tal que $f(d_3) \subset h_1(v_{11})$. Si seleccionamos por ejemplo el valor simple, 1 , $1 \in f_1(d_3, r_1) \cap \{\ast^1_{0,1}\}$, vemos que $f(d_3) \subset h_1(1)$ -obsérvese como $h_1(1) = \{C_a, C_b\}$. Por tanto, se concluye que el *c_valor* $\{\ast^1_{0,1}\}$ es un *c_valor contradictorio*.

- Los **pares de c_valores contradictorios** –definición 10- respecto de r_1 son:

Analicemos primero el par $(\{0\}, \{\ast^1_{0,1}\})$, $\{0\}, \{\ast^1_{0,1}\} \in V'_1$, y veamos si satisface las condiciones de la **definición 10**:

i. Se trata de un *par de c_valores parcialmente comunes*, pues $\{0\} \cap \{\ast^1_{0,1}\} = \{0\}$ – definición 8.

ii. Considérense los elementos $d_1, d_3 \in D$, con $f_1(d_1, r_1) = \{0\}$ y $f_1(d_3, r_1) = \{\ast^1_{0,1}\}$.

iii. Observando la OAT vemos que $f(d_1) = \{C_a\}$ y $f(d_3) = \{C_b\}$.

iv. Por tanto, se concluye que $(\{0\}, \{\ast^1_{0,1}\})$ es un par de *c_valores contradictorios*.

Analicemos a continuación el par $(\{1\}, \{\ast^1_{0,1}\})$, $\{1\}, \{\ast^1_{0,1}\} \in V'_1$.

i. Se trata de un *par de c_valores parcialmente comunes*, pues $\{1\} \cap \{\ast^1_{0,1}\} = \{1\}$.

ii. Considérense los elementos $d_3, d_5 \in D$, con $f_1(d_3, r_1) = \{\ast^1_{0,1}\}$ y $f_1(d_5, r_1) = \{1\}$.

iii. Observando la OAT vemos que $f(d_3) = \{C_b\}$ y $f(d_5) = \{C_a\}$.

Por tanto, se concluye que $(\{1\}, \{\ast^1_{0,1}\})$ es un par de *c_valores contradictorios*.

Puede observarse que el par $(\{0\}, \{1\})$ no es un par de *c_valores contradictorios*, pues $\{0\}$ y $\{1\}$ no son valores parcialmente comunes.

- Veamos la **definición 11**, la cual se centra en *tuplas* –y no en *c_valores*- *atributo-c_valor* de la OAT.

Considérese el atributo r_1 , donde $V'_1 = \{\langle\{0\}\rangle, \langle\{1\}\rangle, \langle\{\ast^1_{0,1}\}\rangle\}$. Obsérvese como las tuplas de *c_valores* se encierran mediante los símbolos $<, >$.

Considére el par de tuplas siguiente:

$e_a = \langle V_{d_1, r_1} \rangle = \langle\{0\}\rangle$ y $e_b = \langle V_{d_2, r_1} \rangle = \langle\{0\}\rangle$. De la OAT se observa que:

i. $(\{0\}, \{0\})$ es un par de valores parcialmente comunes.

ii. $f(d_1) = \{C_a\}$ y $f(d_2) = \{C_b\}$. Por tanto, $f(d_1) \neq f(d_2)$.

iii. Al satisfacerse las dos condiciones anteriores, se concluye por la definición 11 que (e_a, e_b) es un par de tuplas de la OAT contradictorias con respecto a r_1 .

Considérese a continuación el par de tuplas de la OAT del ejemplo:

$e_a = \langle V_{d1,r1} \rangle = \langle \{0\} \rangle$ y $e_b = \langle V_{d3,r1} \rangle = \langle \{^*_{0,1}\} \rangle$. De la OAT se observa que:

i. $\{0\}$ y $\{^*_{0,1}\}$ son un par de *c_valores parcialmente comunes* –definición 8.

ii. $f(d_1) = \{C_a\}$ y $f(d_3) = \{C_b\}$.

iii. Al satisfacerse las dos condiciones anteriores, se concluye por la definición 11 que (e_a, e_b) es un par de tuplas atributo-c_valor de la OAT contradictorias con respecto a r_1 .

Por tanto, e_a y e_b son un par de tuplas atributo-c_valor contradictorias.

Siguiendo el mismo razonamiento, se puede comprobar que las tuplas $e_a = \langle \{1\} \rangle$, $e_b = \langle \{^*_{0,1}\} \rangle$ son también contradictorias.

Considérese a continuación el subconjunto de atributos $\{r_2, r_3\}$.

El dominio $V'_{2,3}$ de tuplas atributo-c_valor de la OAT es, $V'_{2,3} = \{\langle \{\text{rojo}\}, \{0\} \rangle, \langle \{\text{rojo}\}, \{1\} \rangle, \langle \{\text{rojo}\}, \{2\} \rangle, \langle \{\text{verde}\}, \{0\} \rangle, \langle \{\text{azul}\}, \{0\} \rangle, \langle \{\text{azul}\}, \{1\} \rangle, \langle \{\text{azul}\}, \{2\} \rangle\}$.

• Considérense las siguientes tuplas de la OAT de la figura 5.1:

$e_a = \langle V_{d3,r2}, V_{d3,r3} \rangle = \langle \{\text{rojo}\}, \{0\} \rangle$, $e_b = \langle V_{d6,r2}, V_{d6,r3} \rangle = \langle \{\text{rojo}\}, \{^*_{0,2}\} \rangle$. Observando en la OAT las filas correspondientes a las tuplas consideradas vemos que:

i. $V_{d3,r2} \cap V_{d6,r2} = \{\text{rojo}\} \cap \{\text{rojo}\} = \{\text{rojo}\}$ y $V_{d3,r3} \cap V_{d6,r3} = \{0\} \cap \{0,1\} = \{0\}$. Por tanto, se satisface la primera condición de la definición 11.

ii. $f(d_3) = f(d_6) = \{C_b\}$. Por tanto, no se satisface la segunda condición de la definición 11.

Finalmente, se concluye que las tuplas e_a y e_b no son un par de tuplas atributo-c_valor contradictorias.

Si se analiza el resto de pares de tuplas correspondientes al subconjunto de atributos $\{r_1, r_2\}$ observaremos como no existe ningún otro par de tuplas contradictorias. Por tanto, de acuerdo con la definición 12, se concluye que $\{r_1, r_2\}$ es una base de atributos.

Se invita al lector a buscar todas las bases de la OAT del ejemplo.

5.2 Tablas Objeto Atributo como conjuntos de reglas

Hasta ahora se ha considerado las OATs como conjuntos de elementos específicos de conocimiento llamados ejemplos. En esta sección se extiende el concepto de OAT a un conjunto de descripciones abstractas de conocimiento expresadas en términos de reglas de producción o reglas antecedente-consecuente, de manera que ello permita su tratamiento mediante procesos inductivos.

De manera general, las reglas de producción constituyen una de las formas más comunes de expresar la base de conocimientos de un agente reactivo [12], [18]. El antecedente de las reglas está compuesto por conjunciones de pares atributo-valor, mientras que el consecuente consiste en una sola acción –también llamada concepto-, que

puede ser disparada bajo el supuesto de la verdad de su antecedente. Una base de reglas es un conjunto de reglas.

Las reglas de producción que se considerarán consistirán en las ramas de un árbol de decisión, de manera que cada regla corresponde a una rama. La figura 5.2.a ilustra un árbol de decisión y la 5.2.b su conjunto de reglas.

Obsérvese como un árbol de decisión cubre el dominio de valores de todos los atributos.

Definición 13 (reglas completas / full rules). Sea AT el conjunto de atributos a través de los que se describen todos los antecedentes de las reglas de una base de reglas de producción, BC. Una regla completa –full rule- es aquella en cuyo antecedente participan todos los atributos de AT –también conocidos como atributos de la BC-.

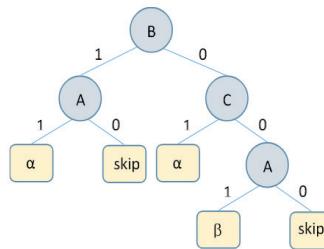


Figura 5.2.a. Árbol de decisión.

$$(B=1) \wedge (A=1) \rightarrow \alpha$$

$$(B=1) \wedge (A=0) \rightarrow \text{skip}$$

$$(B=0) \wedge (C=1) \rightarrow \alpha$$

$$(B=0) \wedge (C=0) \wedge (A=1) \rightarrow \beta$$

$$(B=0) \wedge (C=0) \wedge (A=0) \rightarrow \text{skip}$$

Figura 5.2.b. Conjunto de reglas del árbol.

Por ejemplo, las reglas $(B=1) \wedge (A=1) \rightarrow \alpha$ y $(B=1) \wedge (A=0) \rightarrow \text{skip}$ de la figura 5.2.b no son reglas completas, puesto que su antecedente no contiene todos los atributos de AT que son $AT=\{A,B,C\}$. Mientras, la regla $(B=0) \wedge (C=0) \wedge (A=1) \rightarrow \beta$ sí es una regla completa.

Una regla que no es completa puede convertirse en una regla completa añadiendo a su antecedente aquellos pares atributo-valor cuyos atributos, $r_i, r_i \in AT$, no aparecen en la regla y cuyo valor de dichos atributos se corresponderá con el $c_{\text{valor}}^{*}_{j,k,\dots,p}$, siendo $\{j, k, \dots, p\}$ el dominio de valores simples de r_i , esto es $\{j, k, \dots, p\} \in W_i$ –véase la definición 7.

Por ejemplo la regla $(B=1) \wedge (A=1) \rightarrow \alpha$ puede convertirse en la siguiente regla completa: $(B=1) \wedge (A=1) \wedge (C=*_{0,1}) \rightarrow \alpha$. Obsérvese como el dominio de valores simples del atributo C, W_C , es el conjunto $W_C = \{0, 1\}$.

Una OAT puede considerarse como un conjunto de reglas completas. Las OATs surgidas del proceso de conversión de reglas incompletas en reglas completas son OATs

incompletamente especificadas y pueden ser tratadas por los correspondientes mecanismos de inferencia [10, 18].

Definición 14 (OAT de reglas). Se define una **OAT de reglas** como una estructura formada por reglas de producción de manera que todas las reglas son completas.

5.3 Inferencia inductiva sobre conjuntos de reglas

La cuestión objetivo en este punto es «cómo razonar sobre una OAT expresada como un conjunto de reglas» con objeto de «inferir otro conjunto de reglas objetivo».

La inferencia más común sobre un conjunto de reglas es la deductiva, caracterizada por la búsqueda de conclusiones específicas a partir de premisas que representan conocimientos más generales -por ejemplo, reglas. Mientras, la inferencia inductiva representa un razonamiento sobre elementos concretos de conocimiento con objeto de obtener conclusiones generales. Así las cosas, parece ser que ninguno de los tipos de inferencia «deductiva/ inductiva» se ajusta a la tarea que realmente se propone en este punto, pues lo que se pretende es obtener, a partir de una estructura de conocimiento general –conjunto de reglas- otra estructura de conocimiento general –otro conjunto de reglas-. En esta sección se describe un sencillo procedimiento de inferencia inductiva, ***inductive_infer***, que permite llevar a cabo el objetivo considerado.

Sean $R = \{r_1, r_2, \dots, r_n\}$ un conjunto de atributos, $V = \{V_1, V_2, \dots, V_n\}$ el conjunto de c_{dominios} de los atributos de R y $C = \{C_1, C_2, \dots, C_k\}$ un conjunto de k conceptos.

Considérese una OAT de reglas definida sobre **todos los atributos** de R y los conceptos de C . Obsérvese como todas las reglas de la OAT deben ser **completas**.

El procedimiento ***inductive_infer***[18], se ejecuta sobre una OAT de reglas completas y se basa en los dos siguientes pasos generales:

1. Obtener una *base de atributos* –véase la definición 12-, base_p de la OAT de reglas de acuerdo con algún criterio preestablecido.

2. A partir de la base de atributos obtenida, base_p , obtener una estructura intensional –por ejemplo, un árbol de decisión u otro tipo de estructura- de acuerdo con algún criterio preestablecido.

A veces puede ocurrir que la base de atributos del paso 1 de *inductive_infer* dependa de las características de la estructura intensional del paso 2, con lo cual no es posible realizar los dos anteriores pasos por separado, si no de forma unificada. Veamos algunos ejemplos.

Ejemplo_2.

Considérese la BC de la figura 5.3, cuyo universo es cerrado, es decir no puede darse ninguna otra situación de conocimiento que no sea el expresado mediante las reglas. Con objeto de simplificar la discusión, se omitirán los paréntesis en las ilustraciones de las bases de reglas.

$$\begin{aligned}
 (A=1) \wedge (B=1) &\rightarrow a \\
 (B=0) \wedge (C=1) &\rightarrow a \\
 (A=1) \wedge (B=0) \wedge (C=0) &\rightarrow b
 \end{aligned}$$

Figura 5.3. Base de conocimientos.

El conjunto, R, de atributos de la BC es $R = \{A, B, C\}$, los c_dominios son $V_A = \{\{1\}\}$, $V_B = \{\{0\}, \{1\}\}$, $V_C = \{\{0\}, \{1\}\}$ y $C = \{\alpha, b\}$.

Para que *inductive_infer* pueda actuar sobre la BC, ésta primero debe convertirse en una OAT de reglas completas, de acuerdo con la sección 5.2. Así, se obtiene la BC de la figura 5.4, cuyas reglas son todas completas.

$$\begin{aligned}
 (A=1) \wedge (B=1) \wedge (C = *_{0,1}^C) &\rightarrow a \\
 (A = *_1^A) \wedge (B=0) \wedge (C=1) &\rightarrow a \\
 (A=1) \wedge (B=0) \wedge (C=0) &\rightarrow b
 \end{aligned}$$

Figura 5.4. Base de conocimientos con reglas completas.

A continuación se aplica *inductive_infer*. Supóngase el criterio que determina las características del árbol de decisión objetivo, establece que debe obtenerse un árbol de decisión con un menor número de nodos a partir de una base de atributos con un mínimo número de atributos. Así:

1. Primero se obtiene una base con un mínimo número de atributos. En caso de que existan varias bases con la dicha propiedad, cualquiera de ellas resulta adecuada. El subconjunto de atributos $\{B, C\}$ constituye una base que satisface el mencionado criterio.

2. A continuación, a partir de la base obtenida en el paso anterior debe obtenerse un árbol de decisión con un mínimo número de nodos. En caso de existir varios árboles con una mínima cantidad de nodos, cualquiera de ellos sirve como candidato. La figura 5.5 ilustra un tal árbol.

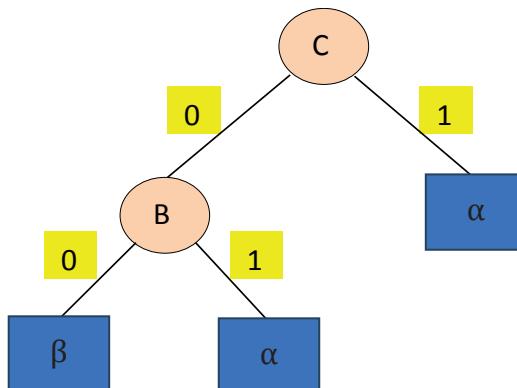


Figura 5.5. Árbol de decisión con un mínimo número de nodos.

En [6, 3] se describe una aproximación del uso de la inferencia inductiva sobre OATs completamente especificadas con objeto de mejorar la calidad de la descripción de los conceptos de una OAT. En [18] se realiza una profundización sobre la misma cuestión considerando bases de reglas completa o incompletamente especificadas y con prioridades establecidas sobre las reglas.

6. LA BASE DE CONOCIMIENTOS -HIPÓTESIS/B.C.-

El modelo utilizado en este trabajo para representar la Hipótesis o B.C. es la estructura **árbol de decisión**. Un árbol de decisión es un conjunto de reglas organizadas jerárquicamente que cubre un conjunto de ejemplos conocido como **OAT** el cual constituye el **dominio de entrada de datos**. El hecho de constituir un dominio significa que aquellos ejemplos que no figuran/aparecen en la propia OAT, se considera que no pueden aparecer, es decir, no existen –lo dicho se conoce como **hipótesis de dominio cerrado**. Esto se considera así porque la OAT se interpreta como un modelo válido –útil-, cuyo árbol de decisión inducido debe servir para clasificar correctamente «cualquier» elemento del mismo –esto es, de la OAT-.

No obstante, sí se considera que **la OAT puede evolucionar con el tiempo**, es decir, pueden aparecer nuevos ejemplos en posteriores momentos de tiempo. En tal caso, el concepto de OAT ya no se considera como un modelo válido «para siempre», si no que su validez queda limitada hasta el instante en que aparezcan nuevos ejemplos. Tales modelos en los que se considera la hipótesis de dominio cerrado pero que pueden evolucionar a lo largo del tiempo se conocen como **modelos/dominios evolutivos**. La cuestión que surge aquí es ¿Cómo enfocar la inducción de árboles de decisión cuando la OAT evoluciona con el tiempo?

6.1 Árboles de Decisión para el aprendizaje

Como la hipótesis/árbol de decisión evoluciona a medida que lo hace la OAT – es decir, a medida que aparecen nuevos ejemplos-, entonces para que ésta pueda ser actualizada en cada ciclo de aprendizaje, esto es, cuando aparezcan nuevos ejemplos del dominio de la OAT, es necesario conocer cuáles de tales ejemplos ya han aparecido en ciclos de aprendizaje anteriores y cuáles de ellos no han aparecido anteriormente. De esta forma, los ejemplos «no aparecidos» se etiquetarán mediante el símbolo **n.a.**. Los árboles de decisión que se generan bajo esta consideración se conocen como **árboles de decisión para el aprendizaje –ADA-**. Un ADA contiene todo el historial de los ejemplos que han aparecido y también de aquellos que no han aparecido hasta el momento actual.

En esta sección, el concepto de «ejemplo» se refiere al conjunto de pares atributo-valor a través de los cuales viene descrito el ejemplo así como a los conceptos asociados al mismo. En otras palabras, en una OAT con n atributos, un ejemplo se representará como una $n+1$ tupla $(t_i^1, t_i^2, \dots, t_i^n, \{C_i, C_j\})$, siendo t_i^k los pares atributo valor y $\{C_i, C_j\}$ los conceptos asociados al ejemplo.

A modo de ejemplo ilustrativo de las características de un ADA, sin entrar en cuestiones más profundas sobre su uso en un procedimiento de aprendizaje, considérese el conjunto de tres ejemplos de la figura 6.1.

Un árbol de decisión para el aprendizaje inferido a partir de los ejemplos de la figura 6.1 debe cubrir, por una parte, dichos ejemplos y por otra parte el resto de ejemplos del dominio que no aparecen en la figura 6.1, distinguiéndolos mediante el símbolo *n.a.*. La figura 6.2 muestra un ejemplo de un tal árbol de decisión para el aprendizaje.

$$(A=0) \wedge (B=0) \wedge (C=0) \rightarrow C1$$

$$(A=0) \wedge (B=1) \wedge (C=0) \rightarrow C2$$

$$(A=1) \wedge (B=0) \wedge (C=1) \rightarrow C3$$

Figura 6.1. Conjunto de ejemplos.

Obsérvese como los dominios de los atributos A, B, C de la figura 6.1 son binarios.

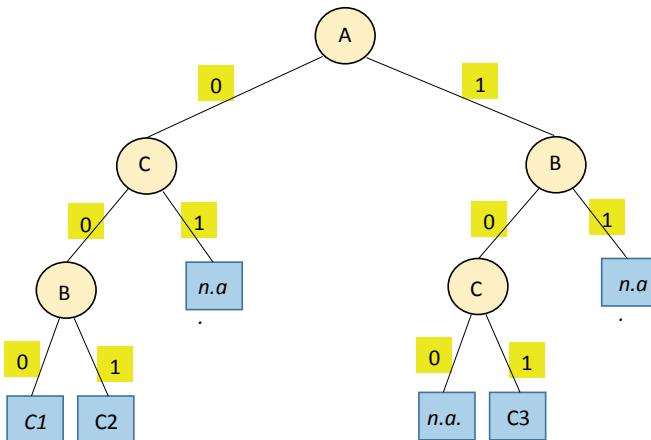


Figura 6.2. Árbol de decisión para el aprendizaje -ADA-.

Como se ha comentado al principio de esta sección, la actualización de un ADA debido a la aparición de **nuevos ejemplos** en un **ciclo de aprendizaje**, supone conocer la relación de estos ejemplos nuevos con los ejemplos ya aparecidos en ciclos de aprendizaje anteriores, los cuales se encuentran descritos en el ADA. Lo dicho se manifiesta a través de las siguientes cuestiones:

- ¿Han aparecido anteriormente los nuevos ejemplos que aparecen en el ciclo de aprendizaje actual? En caso afirmativo ya vienen descritos –cubiertos- por el ADA. En tal caso, habrá que obtener el conocimiento necesario de dichos ejemplos a partir del ADA con objeto de proceder a actualizar el propio ADA.
- En caso de no haber aparecido anteriormente, hay que conocer si causan confusiones con ejemplos ya aparecidos anteriormente. Dos ejemplos causan confusión si cada par atributo-valor de los ejemplos tiene valores comunes en ambos ejemplos, pero los conceptos asociados a los ejemplos son diferentes. En otras palabras, si $e_a = (t_1^1, t_1^2, \dots, t_1^n, \{C_s, \dots, C_p\})$ y $e_b = (t_k^1, t_k^2, \dots, t_k^n, \{C_p, \dots, C_q\})$ son los ejemplos, entonces $t_1^1 \cap t_k^1 \neq \emptyset, t_1^2 \cap t_k^2 \neq \emptyset, \dots, t_1^n \cap t_k^n \neq \emptyset$ y $\{C_s, \dots, C_p\} \neq \{C_p, \dots, C_q\}$.

En caso afirmativo deberá aplicarse un proceso de reescritura de la OAT con objeto de eliminar todas las confusiones. En caso negativo habrá que obtener el conocimiento necesario de los ejemplos descritos en el ADA con objeto de proceder a actualizar el propio ADA.

iii. ¿Cómo relacionar el ADA actual con los nuevos ejemplos con objeto de obtener un ADA actualizado de acuerdo al criterio considerado?

Las dos primeras cuestiones implican que debe conocerse la historia pasada de los ejemplos que aparecen en cada paso/ciclo de aprendizaje inducto-deductivo del proceso de aprendizaje. Tal conocimiento está contenido en el ADA actual.

Recuérdese que el ADA de un ciclo de aprendizaje considerado, sólo se utiliza como estructura soporte de todos los ejemplos que han sido proporcionados y de los que no lo han sido hasta el ciclo considerado, por parte del entorno. Por tanto, un ADA no constituye la estructura árbol de decisión adecuada para clasificar los ejemplos, pues incluye todos los ejemplos etiquetados con el símbolo n.a.. Para solventar esta cuestión, el ADA obtenido en cada ciclo de aprendizaje deberá convertirse en un árbol de decisión apropiado, que llamaremos **árbol de decisión final** – abreviado **ADF**. A partir de un ADA puede obtenerse fácilmente un ADF, el cual no contendrá ramas etiquetadas con el símbolo n.a.. El ADF es, de hecho, el árbol de decisión objetivo.

Otra cuestión relacionada con el aprendizaje se refiere a la calidad del **ADF**. Es imprescindible que el **proceso de aprendizaje converja** hacia un árbol que satisfaga plenamente el criterio considerado. Esto es, que el ADF generado en cada ciclo de aprendizaje converja hacia el hipotético árbol de decisión objetivo final –excepto en aquellos ciclos donde los nuevos ejemplos que aparecen resulten irrelevantes de acuerdo con el criterio considerado para la mejora del árbol, donde puede ocurrir que el ADF generado en el ciclo actual no sea mejor ni peor que el ADF del anterior ciclo-.

6.2 Aprendizaje de Árboles de Decisión

El proceso inferencial encargado de actualizar el ADA y el ADF en cada ciclo de aprendizaje de un modelo Feedback-1 se basa en los dos siguientes elementos de conocimiento:

- Los nuevos ejemplos de la OAT que aparecen en el ciclo de aprendizaje actual.
- El ADA/hipótesis generado en el ciclo de aprendizaje anterior.

Por tanto, dicho proceso debe razonar adecuadamente sobre un conjunto de instancias específicas de conocimiento –los nuevos ejemplos- y sobre una instancia abstracta –el ADA-, de ahí su consideración como razonamiento **inducto-deductivo**. El resultado será un ADA actualizado a partir de los dos elementos de conocimiento mencionados.

En relación a la cuestión de la complejidad computacional del proceso inducto-deductivo incremental, cabe mencionar que ésta será menor que la de un proceso

únicamente inductivo que actúa sólo sobre una OAT que contenga, en cada ciclo de aprendizaje, todos los ejemplos aparecidos hasta el momento y que todos ellos se procesarán en el correspondiente ciclo de aprendizaje, pues el número de reglas del ADA actual será, posiblemente y de forma sensible, menor que el número de ejemplos de la OAT.

En esta sección se presenta un algoritmo de adquisición de conocimiento como punto central de un sistema de aprendizaje autónomo incremental. Dicho algoritmo toma como entradas un ADA/hipótesis actual y una OAT que contiene los nuevos ejemplos del ciclo de aprendizaje en cuestión –se trata de los ejemplos proporcionados por el entorno-, descritos mediante reglas de producción, y actualiza el ADA de acuerdo con los ejemplos de la OAT.

Características del método:

- i. Cada uno de los ciclos de aprendizaje se basa en el procesamiento del ADA actual –esto es, en todos los ciclos de aprendizaje procesados hasta el ciclo actual-, el cual cubre todos los ejemplos aparecidos hasta el momento, y de los ejemplos proporcionados por el entorno en el ciclo dado. El ADA, al ser una estructura abstracta, contiene reglas en lugar de ejemplos. Por tanto, el número de reglas a procesar será con toda seguridad –bastante- menor que el número de ejemplos de una OAT equivalente.
- ii. En el **primer ciclo/paso de aprendizaje**, se procesa, por una parte, el primer conjunto de ejemplos proporcionados por el entorno, descritos en un formato OAT de reglas -abreviado OAT-. Por otra parte, junto con los ejemplos de la OAT, también deberá procesarse el ADA actual de este primer ciclo, que cubre todos los ejemplos del dominio de la OAT que no aparecen en la propia OAT de este primer ciclo. La descripción de tales ejemplos se obtiene a partir de los dominios de los atributos de la OAT y todos ellos tienen asociado el concepto *n.a.*, pues son ejemplos que todavía no han aparecido. Aunque la generación del ADA actual de este ciclo pueda parecer un proceso complejo y engoroso, en realidad su desarrollo es muy simple.
- iii. El método permite obtener un ADA ajustado al criterio utilizado, sin limitaciones.
- iv. Si la OAT contiene atributos con dominios multivaluados enormemente vastos, posiblemente se den situaciones donde el número de reglas a procesar sea muy numeroso. Existen diferentes opciones para tratar tales situaciones:
 - Binarizar o discretizar los dominios de los atributos afectados con objeto de reducirlos.
 - Procesar la OAT de forma aproximada, aplicando algún criterio que garantice que los árboles de decisión obtenidos serán razonables.

Ejemplo. Considérese una OAT (actual) constituida inicialmente por los siguientes tres ejemplos expresados en forma de reglas:

$$d_1 : (A=0) \wedge (B=0) \wedge (C=0) \rightarrow C_1$$

$$d_2 : (A=0) \wedge (B=1) \wedge (C=0) \rightarrow C_2$$

$$d_3 : (A=1) \wedge (B=0) \wedge (C=1) \rightarrow C_3$$

Cada ejemplo de la OAT viene descrito mediante un conjunto de tres atributos, $R = \{A, B, C\}$, con c_{dominios} binarios, $V_A = V_B = V_C = \{0, 1\}$. El conjunto de conceptos es, $C = \{C_1, C_2, C_3\}$.

Como inicialmente todavía no se ha creado ningún ADA que cubra ejemplos de la OAT, entonces el árbol de decisión actual del primer ciclo debe cubrir todos los ejemplos del dominio de la OAT que no aparecen en la propia OAT, los cuales se asociarán al concepto *n.a.* Un ejemplo de un tal árbol se ilustra en la figura 6.3. Existen obviamente otros posibles árboles de decisión en esta primera etapa; cualquiera de ellos resulta aceptable.

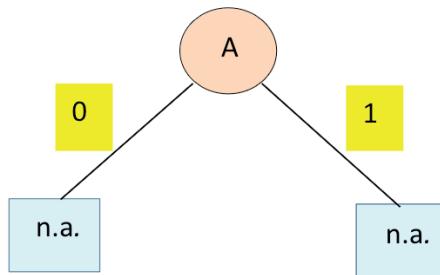


Figura 6.3. ADA del ciclo inicial.

A continuación se unen en un único grupo las reglas de producción correspondientes a los ejemplos de la OAT y al árbol de decisión. Tales reglas se muestran en la figura 6.4. El conjunto está formado por cinco reglas. Sobre este conjunto de reglas debe aplicarse el proceso inductivo descrito en [10, 18] con objeto de crear un ADA actualizado. No obstante, para poder aplicar el mencionado proceso inductivo, todas las reglas deben ser reglas completas –full rules-. Sólo las dos últimas reglas no son reglas completas; convertirlas es fácil, la figura 6.5 muestra el resultado.

$$(A=0) \wedge (B=0) \wedge (C=0) \rightarrow C_1,$$

$$(A=0) \wedge (B=1) \wedge (C=0) \rightarrow C_2$$

$$(A=1) \wedge (B=0) \wedge (C=1) \rightarrow C_3$$

$$(A=0) \rightarrow \text{n.a.}$$

$$(A=1) \rightarrow \text{n.a.}$$

Figura 6.4. Conjunto de reglas del primer ciclo.

Puesto que el ADA debe actualizarse de acuerdo con los ejemplos de la OAT, entonces las reglas se han dispuesto de acuerdo con las prioridades, situando en la parte superior del conjunto las reglas concernientes a los ejemplos de la OAT y debajo

las reglas completas concernientes al ADA, tal como se ilustra en la figura 6.5. Ahora ya puede aplicarse el procedimiento inductivo descrito en [10, 18] al conjunto de reglas. Un posible resultado de la aplicación del procedimiento inductivo es el ADA que se muestra en la figura 6.2. Dependiendo del criterio que se aplique, existen óbviamente otros posibles ADAs resultantes.

$$\begin{aligned}
 & (A=0) \wedge (B=0) \wedge (C=0) \rightarrow C_1 \\
 & (A=0) \wedge (B=1) \wedge (C=0) \rightarrow C_2 \\
 & (A=1) \wedge (B=0) \wedge (C=1) \rightarrow C_3 \\
 & (A=0) \wedge (B=^{*B}_{0,1}) \wedge (C=^{*C}_{0,1}) \rightarrow \text{n.a.} \\
 & (A=1) \wedge (B=^{*B}_{0,1}) \wedge (C=^{*C}_{0,1}) \rightarrow \text{n.a.}
 \end{aligned}$$

Figura 6.5. Conjunto de reglas completas del primer ciclo.

A continuación se describe un algoritmo de aprendizaje con todos los detalles necesarios para actualizar el ADA actual en un ciclo arbitrario del proceso de aprendizaje, de acuerdo con los nuevos ejemplos que aparecen en el ciclo actual proporcionados por la OAT –abreviado OAT actual–.

ALGORITMO: INCREMENTAL KNOWLEDGE ACQUISITION(*E, OAT_actual, ADA_actual*)

Inicio

Sea **E** el conjunto de nuevos ejemplos que aparecen en el ciclo de aprendizaje actual y sea **OAT_actual** la descripción de tales ejemplos mediante los correspondientes atributos y conceptos. (1)

si existen contradicciones entre los elementos de **E**, **entonces** (2)

restablecer la **OAT_actual** con objeto de eliminar las contradicciones y reescribir el **ADA_actual** de acuerdo con los términos adoptados en el restablecimiento de la **OAT_actual**. (3)

fin si;

Sea **ADA_actual** el árbol de decisión para el aprendizaje actual; (4)

Actualizar, a partir de la **OAT_actual** y del **ADA_actual**, el conjunto de atributos, el conjunto de conceptos y los dominios de cada atributo; (5)

Obtener el conjunto de reglas completas del **ADA_actual** de acuerdo con el conjunto de atributos y sus dominios en el propio **ADA_actual**; (6)

Completar, si procede, el conjunto de reglas completas de la **OAT_actual** y del **ADA_actual** de acuerdo con el conjunto de atributos y dominios actualizados – punto (5)–, de manera que las reglas resultantes sean completas; (7)

Mirar si hay contradicciones entre el conjunto de reglas de la **OAT_actual** completada en el paso anterior y el conjunto de reglas del **ADA_actual** completado en el paso anterior –esto son los ejemplos que ya han aparecido en los ciclos anteriores y se describen a través del **ADA_actual**; (8)

si no hay contradicciones **entonces** {**actualizar el ADA_actual**} (9)

i. Unificar los conjuntos de reglas completas de *OAT_actual* y *ADA_actual*, de manera que las reglas de *OAT_actual* tengan mayor prioridad; (10)

ii. Aplicar el procedimiento inductivo, *inductive_infer* al conjunto de reglas resultante y obtener el árbol de decisión actualizado, *ADA_actualizado*; (11)

Sino (12)

restablecer la *OAT_actual* y el *ADA_actual* con objeto de eliminar las contradicciones; (13)

Volver al punto (9); (14)

fin si;

fin

6.3 Ejemplos

A continuación, se ilustra la aplicación del anterior algoritmo a través de una serie de ejemplos sencillos ilustrativos. Con objeto de facilitar la comprensión del algoritmo evitando hacer un uso extensivo en su resolución, se omitirá el caso de existencia de contradicciones entre los ejemplos del conjunto *E* –punto (1) del algoritmo- y entre los ejemplos de la *OAT_actual* y del *ADA_actual* –punto (9) del algoritmo-.

Cada ejecución del algoritmo constituye un ciclo del proceso de aprendizaje.

Con objeto de facilitar la discusión, se simplificará la descripción del valor de incertezza, $r_i^{*_{a...b}}$, de los atributos de la siguiente forma: en la asignación, $r_i = r_i^{*_{a...b}}$, se omitirá a partir de este punto el superíndice, *i*, de $r_i^{*_{a...b}}$; así se escribirá simplemente $r_i = r_i^{*_{a...b}}$. Además, para evitar una escritura excesivamente formal y facilitar el desarrollo de las soluciones, los *c_valores* de los ejemplos de las OAT se escribirán sin paréntesis, por ejemplo, el *c_valor*, {0}, se escribirá simplemente como 0.

Recuérdese que los conceptos *OAT* y *CIS_OAT* se consideran sinónimos.

Obsérvese como cada llamada al algoritmo recibe tres parámetros: (*E*, *OAT_actual*, *ADA_actual*). La *OAT_actual* contiene la descripción de los ejemplos de *E* en términos de atributos y la asignación de los correspondientes conceptos a cada ejemplo.

Los dominios de los atributos que se consideran en el algoritmo son dominios de valores simples, representados por el símbolo W_i en la definición 7. Por tanto, cualquier mención del concepto «dominio» en los ejemplos, se referirá al «dominio simple».

Ejemplo_1.

Ciclo_1 –inicio del proceso de aprendizaje-. Se ejecuta por primera vez el algoritmo.

- Se presentan los nuevos ejemplos del ciclo en cuestión. Estos ejemplos constituyen el conjunto, *E*, del algoritmo. (1)

Sea $E = \{d_1, d_2, d_3, d_4\}$, donde los d_i se describen según la figura 6.6. La descripción de los ejemplos en términos de valores de atributos juntamente con los conceptos descritos por cada ejemplo, se conoce, como se menciona en el algoritmo de aprendizaje, como OAT_actual. A partir de la OAT_actual deben determinarse los atributos presentes –cuyo número puede variar en ciclos posteriores- y los dominios de dichos atributos –que también pueden variar en posteriores ciclos–.

La OAT_actual de este ciclo se muestra también en la figura 6.6.

D/R	r_1	r_2	C
d_1	0	0	c_1
d_2	0	1	c_2
d_3	0	0	c_1
d_4	1	0	c_1

Figura 6.6. OAT_actual del Ciclo_1.

A partir de los ejemplos de la OAT_actual se observan el conjunto de atributos actual, **R**, el conjunto de conceptos, **C**, y los dominios de valores simples de cada atributo, W_{ri} , $i=1, 2$. Así, queda:

$$E = \{d_1, d_2, d_3, d_4\}$$

$$R = \{r_1, r_2\}$$

$$C = \{c_1, c_2\}$$

$$W_{r1} = \{0, 1\} \quad W_{r2} = \{0, 1\}$$

- A continuación se determina si existen contradicciones entre los ejemplos de E. (2)

A partir de la OAT_actual puede observarse como **no existen contradicciones** entre los ejemplos de E.

- A continuación, se describe el **ADA_actual**, el cual se pasa, junto con el conjunto E y la OAT_actual, como parámetro al algoritmo. (4)

El ADA_actual del Ciclo_1 de cualquier problema es estándar y puede configurarse con un nodo raíz, cuya etiqueta puede ser cualquiera de los atributos del conjunto R de la OAT_actual, y tantas ramas desde la raíz como valores tenga el dominio del atributo que se usa para etiquetar la propia raíz. El árbol ilustrado en la figura 6.7 constituye un ADA_actual. Existen otros posibles árboles que pueden usarse como ADA_actual, cualquiera de ellos resulta válido.

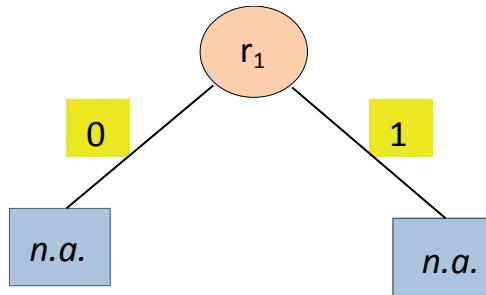


Figura 6.7. *ADA_actual* del ciclo 1.

Las reglas del *ADA_actual* del árbol de la figura 6.7 se ilustran en la figura 6.8.

$$(r_i=0) \rightarrow \text{n.a.}$$

$$(r_i=1) \rightarrow \text{n.a.}$$

Figura 6.8. Reglas del *ADA_actual* de la figura 6.7.

El árbol de la figura 6.7 sólo contiene un atributo, cuyo dominio es binario, y en el que aparece un único concepto. Es decir:

$$R' = \{r_i\} \quad C' = \{\text{n.a.}\} \quad W'_{r1} = \{0, 1\}$$

Obsérvese como los conjuntos de atributos, conceptos y dominios del *ADA_actual*, se han representado con el mismo símbolo que en el caso de la *OAT_actual* pero seguidos de un apóstrofe, esto es R' , C' y W'_{ri} respectivamente.

- A continuación deben **actualizarse**, a partir de la *OAT_actual* y del *ADA_actual*, el conjunto de atributos, los dominios de los atributos y el conjunto de conceptos, de la siguiente forma: (5)

Los conjuntos de atributos, dominios y el conjunto de conceptos actualizados se representarán respectivamente por R^{act} , W_{ri}^{act} y C^{act} , de manera que:

$$R^{\text{act}} = R \cup R' \quad C^{\text{act}} = C \cup C' \quad W_{ri}^{\text{act}} = W_{ri} \cup W'_{ri}$$

Así, queda:

$$R^{\text{act}} = \{r_1, r_2\}$$

$$C^{\text{act}} = \{c_1, c_2, \text{n.a.}\}$$

$$W_{r1}^{\text{act}} = \{0, 1\} \quad W_{r2}^{\text{act}} = \{0, 1\}$$

Recuérdese que el símbolo *n.a.*, utilizado como concepto del *ADA_actual*, sólo se usa para identificar los ejemplos que no han aparecido en ninguno de los ciclos anteriores al ciclo actual. A efectos prácticos, dicho símbolo no puede causar ningún tipo de confusión con los ejemplos que aparecen en la *OAT_actual* del presente ciclo.

- A continuación, se ejecuta la siguiente acción del algoritmo:

«Obtener el conjunto de reglas completas del *ADA_actual* de acuerdo con el conjunto de atributos y sus dominios en el propio *ADA_actual*» (6)

El conjunto de reglas completas del ADA_actual se basa en la extensión del conjunto de reglas del ADA_actual –figura 6.8- de acuerdo con el conjunto de atributos, el conjunto de conceptos y los conjuntos de dominios de los atributos del propio ADA_actual, esto es, se basa en R' , C' y W_{ri}^* respectivamente. En la figura 6.9 se ilustran las reglas completas de la figura 6.8.

$$(r_1=0) \rightarrow \text{n.a.}$$

$$(r_1=1) \rightarrow \text{n.a.}$$

Figura 6.9. Conjunto de reglas completas del ADA_actual de la figura 6.7.

Obsérvese como en este ejemplo particular las reglas de las figuras 6.8 y 6.9 son idénticas.

- A continuación, se ejecuta la siguiente acción del algoritmo:

«Completar, si procede, el conjunto de reglas completas de la OAT_actual y del ADA_actual de acuerdo con el conjunto de atributos y dominios actualizados, de manera que las reglas resultantes sean completas» **(7)**

En este punto debe completarse/actualizarse, si procede, los conjuntos de reglas de las figuras 6.6 y 6.9 de acuerdo con los conjuntos de atributos, conceptos y dominios de los atributos actualizados, esto es, R^{act} , C^{act} , W_{ri}^{act} , $i=1..2$, respectivamente.

El recubrimiento de las reglas de la figura 6.9 puede representarse a través del mapa de la figura 6.10.a. Obsérvese como en dicho recubrimiento sólo participan los atributos, conceptos y dominios de R' , C' , W_{ri}^* respectivamente. Lo que debe hacerse en este punto es extender dicho recubrimiento a un nuevo recubrimiento en el que participen los conjuntos de atributos, conceptos y dominios actualizados –punto (5)-. La figura 6.10.b ilustra un tal recubrimiento extendido.

r_1	0	1
r_2	n.a.	n.a.

Figura 6.10.a. Recubrimiento del ADA_actual.

r_2 / r_1	0	1
0	n.a.	n.a.
1	n.a.	n.a.

Figura 6.10.b. Recubrimiento extendido.

A partir de la figura 6.10.b resulta fácil completar/actualizar el conjunto de reglas completas del ADA_actual. La figura 6.11 ilustra un tal conjunto.

$$(r_1=0) \wedge (r_2=^*_{0,1}) \rightarrow \text{n.a.}$$

$$(r_1=1) \wedge (r_2=^*_{0,1}) \rightarrow \text{n.a.}$$

Figura 6.11. Conjunto actualizado de reglas completas del ADA_actual de la figura 6.7.

Obsérvese como en todas las reglas de la figura 6.11 aparecen los dos atributos de conjunto actualizado, $R^{\text{act}} = \{r_1, r_2\}$.

Respecto a la OAT_actual –figura 6.6-, no hay nada que completar, puesto que los conjuntos de atributos y dominios, R y W_i , $i = 1..2$, coinciden con los conjuntos R^{act} y, W_i^{act} , $i = 1..2$, respectivamente.

- A continuación se ejecuta la siguiente acción del algoritmo:

«Mirar si hay contradicciones entre el conjunto de reglas de la **OAT_actual** completado en el paso anterior y el conjunto de reglas del **ADA_actual** completado en el paso anterior» (8)

En este punto se observa si hay contradicciones entre los ejemplos/reglas completas de la OAT_actual –figura 6.6- y las reglas completas del ADA_actual –figura 6.11-. Recuérdese que el concepto n.a. nunca causa contradicción con ninguno de los conceptos del conjunto, C , de la OAT_actual. Por tanto se puede afirmar que no existen contradicciones entre las reglas de la OAT_actual y las reglas del ADA_actual.

- A continuación se ejecuta la siguiente acción del algoritmo:

«**si** no hay contradicciones **entonces** {*actualizar el ADA_actual*}» (9)

Como no hay contradicciones, entonces el algoritmo continúa con la primera acción en el dominio de la sentencia condicional.

- A continuación se ejecuta la siguiente acción del algoritmo:

i. Unificar los conjuntos de reglas completas de *OAT_actual* y *ADA_actual*, de manera que las reglas de *OAT_actual* tengan mayor prioridad; (10)

En este punto se trata de unificar en un solo conjunto las reglas de las figuras 6.6 y 6.11. La figura 6.12 muestra dicha unificación.

$$(r_1=0) \wedge (r_2=0) \rightarrow c_1$$

$$(r_1=0) \wedge (r_2=1) \rightarrow c_2$$

$$(r_1=0) \wedge (r_2=0) \rightarrow c_1$$

$$(r_1=1) \wedge (r_2=0) \rightarrow c_1$$

$$(r_1=0) \wedge (r_2=^*_{0,1}) \rightarrow \text{n.a.}$$

$$(r_1=1) \wedge (r_2=^*_{0,1}) \rightarrow \text{n.a.}$$

Figura 6.12. Unificación de las reglas completas de la OAT_actual y del ADA_actual dispuestas según prioridades Top-Down.

Las reglas de la figura 6.12 pueden interpretarse más claramente mediante la tabla de la figura 6.13, que en realidad constituye un mapa de Karnaugh.

$r_2 \setminus r_1$	0	1
0	$c_1, \text{n.a.}$	$c_1, \text{n.a.}$
1	$c_2, \text{n.a.}$	n.a.

Figura 6.13. Mapa completo de las reglas de la figura 6.12.

Cada celda de dicha tabla representa aquellas reglas de la figura 6.12 cuyo antecedente coincide con la numeración de la celda en la tabla, de manera que dentro de la celda se escriben los conceptos descritos por las correspondientes reglas. Dichos conceptos se escriben linealmente en la celda, de izquierda a derecha, de manera que los conceptos de más a la izquierda corresponden a reglas con mayor prioridad. De esta forma, sólo el primer concepto –el de más a la izquierda- será el que realmente corresponde a la celda, tal como se ilustra en la figura 6.14.

$r_2 \setminus r_1$	0	1
0	c_1	c_1
1	c_2	n.a.

Figura 6.14. Mapa sin contradicciones de las reglas de la figura 6.12.

La discusión anterior describe una técnica sencilla sobre cómo pueden eliminarse las contradicciones en un conjunto de reglas con prioridades Top-Down. No obstante, cabe enfatizar que el concepto n.a. no causa confusión con ningún otro concepto.

- A continuación se ejecuta la siguiente acción del algoritmo:
- ii. « Aplicar el procedimiento inductivo, *inductive_infer*, al conjunto de reglas resultante y obtener el árbol de decisión actualizado, *ADA_actualizado* » (11)

El conjunto de reglas resultante es aquel al cual se han aplicado las prioridades, tal como se muestra en la figura 6.14, y por tanto están libres de contradicciones. La figura 6.15 ilustra dicho conjunto de reglas.

$$\begin{aligned}
 & (r_1=0) \wedge (r_2=0) \rightarrow c_1 \\
 & (r_1=0) \wedge (r_2=1) \rightarrow c_2 \\
 & (r_1=1) \wedge (r_2=0) \rightarrow c_1 \\
 & (r_1=1) \wedge (r_2=1) \rightarrow \text{n.a.}
 \end{aligned}$$

Figura 6.15. Conjunto de reglas resultante de la aplicación de las prioridades.

Con objeto de actualizar el *ADA_actual* –Figura 6.7- con los nuevos ejemplos de la *OAT_actual* –Figura 6.6-, se aplica al conjunto de reglas de la figura 6.15 el procedimiento de inferencia, *inductive_infer* [10, 18]. El criterio en el que se basa un tal procedimiento depende de la naturaleza del problema a resolver. En nuestro ejemplo, con objeto de clarificar esta discusión, se ha elegido el criterio basado en la creación de un árbol con un mínimo número de nodos terminales/hoja, en otras palabras, con un mínimo número de reglas. Un tal árbol se ilustra en la figura 6.16.

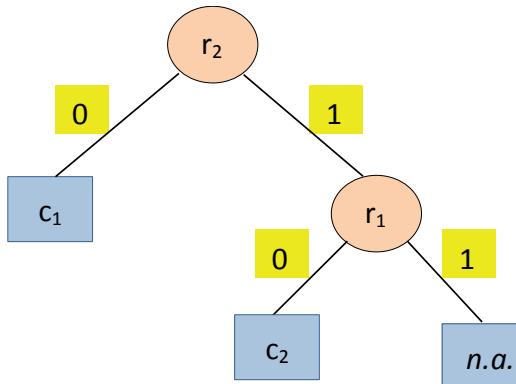


Figura 6.16. Árbol de decisión para el aprendizaje de mínimo número de ramas del Ciclo_1.

El árbol de la figura 6.16 se conoce como el árbol de decisión actualizado, **ADA_actualizado**, del Ciclo_1. Dicho árbol se pasará como parámetro al siguiente ciclo –Ciclo_2-, constituyendo el ADA_actual de dicho ciclo.

Ciclo_2. Se ejecuta por segunda vez el algoritmo –debido a la aparición de nuevos ejemplos-.

- Considérese la aparición de nuevos ejemplos desde el entorno, lo cual provoca una nueva ejecución del algoritmo o **Ciclo_2**. Sea $E = \{d_5, d_6, d_7, d_8\}$ dicho conjunto de ejemplos. (1)

Los ejemplos $d_i \in E$, $i=5..8$, se describen mediante la **OAT_actual** de la figura 6.17. A partir de la OAT_actual deben determinarse los atributos presentes –cuyo número puede variar en ciclos posteriores- y los dominios de dichos atributos –que también pueden variar en posteriores ciclos-.

D/R	r ₁	r ₂	r ₃	C
d_5	0	0	0	c ₁
d_6	0	1	1	c ₂
d_7	0	0	1	c ₁
d_8	1	0	1	c ₁

Figura 6.17. OAT_actual del Ciclo_2.

A partir de los ejemplos de la OAT_actual se observan el conjunto de atributos actual, **R**, el conjunto de conceptos, **C**, y los dominios de cada atributo, W_{ri} , $i=1, 2, 3$ y el conjunto de conceptos. Así, queda:

$$E = \{d_5, d_6, d_7, d_8\}$$

$$R = \{r_1, r_2, r_3\}$$

$$C = \{c_1, c_2\}$$

$$W_{r1} = \{0, 1\} \quad W_{r2} = \{0, 1\} \quad W_{r3} = \{0, 1\}$$

- A continuación se determina si existen contradicciones entre los elementos de E. (2)

Obsérvese como no hay contradicciones entre los ejemplos de la OAT_actual, en la que aparecen **cuatro ejemplos**, los cuales vienen descritos por **tres atributos**, es decir, un atributo más, r_3 , que los ejemplos del ciclo anterior –**CICLO_1**-. Este atributo es binario. Por tanto, el presente ciclo de aprendizaje deberá tener en cuenta la aparición de un nuevo atributo binario. En los próximos pasos se verá cómo debe tratarse esta situación.

- A continuación se describe el ADA_actual, el cual se pasa, junto con el conjunto E, como parámetro al algoritmo. (4)

El **ADA_actual** se ilustra en la figura 6.16.

Las reglas del ADA_actual se ilustran en la figura 6.18.

$$(r_2=0) \rightarrow c_1$$

$$(r_1=0) \wedge (r_2=1) \rightarrow c_2$$

$$(r_1=1) \wedge (r_2=1) \rightarrow \text{n.a.}$$

Figura 6.18. Reglas del ADA_actual –del CICLO_2-.

El árbol de la figura 6.16 contiene dos atributos binarios y tres conceptos. Es decir:

$$R' = \{r_1, r_2\}$$

$$C' = \{c_1, c_2, \text{n.a.}\}$$

$$W'_{r1} = \{0, 1\} \quad W'_{r2} = \{0, 1\}$$

- A continuación se ejecuta el siguiente punto del algoritmo, «Actualizar, a partir de la **OAT_actual** y del **ADA_actual**, el conjunto de atributos, el conjunto de conceptos y los dominios de cada atributo» (5)

Para ello se procede de la siguiente forma:

$$R^{\text{act}} = R \cup R'$$

$$C^{\text{act}} = C \cup C'$$

$$W_{ri}^{\text{act}} = W_{ri} \cup W'_{ri}$$

Así, los **conjuntos actualizados** quedan como:

$$R^{\text{act}} = \{r_1, r_2, r_3\}$$

$$C^{\text{act}} = \{c_1, c_2, \text{n.a.}\}$$

$$W_{r1}^{\text{act}} = \{0, 1\} \quad W_{r2}^{\text{act}} = \{0, 1\} \quad W_{r3}^{\text{act}} = \{0, 1\}$$

- Se procede a la ejecución de la siguiente acción del algoritmo: «Obtener el conjunto de reglas completas del ADA_actual de acuerdo con el conjunto de atributos y sus dominios en el propio ADA_actual» **(6)**

El conjunto de reglas completas del ADA_actual se basa en la extensión de su conjunto de reglas –figura 6.18- de acuerdo con el conjunto de atributos, el conjunto de conceptos y los conjuntos de dominios de los atributos del propio ADA_actual, esto es en R' , C' y W'_r respectivamente. En la figura 6.19 se ilustran las reglas completas de la figura 6.18.

$$(r_1 = *_{0,1}) \wedge (r_2 = 0) \rightarrow c_1$$

$$(r_1 = 0) \wedge (r_2 = 1) \rightarrow c_2$$

$$(r_1 = 1) \wedge (r_2 = 1) \rightarrow \text{n.a.}$$

Figura 6.19. Conjunto de reglas completas del ADA_actual de la figura 6.16.

- A continuación se procede con la siguiente acción del algoritmo.

«Completar, si procede, el conjunto de reglas completas de la **OAT_actual** y del **ADA_actual** de acuerdo con el conjunto de atributos y dominios actualizados, de manera que las reglas resultantes sean completas» **(7)**

El ADA_actual de este ciclo sólo tiene dos atributos, $R' = \{r_1, r_2\}$, mientras que el conjunto actualizado de ejemplos –punto (5) del algoritmo- contiene tres atributos, $R^{\text{act}} = \{r_1, r_2, r_3\}$. Por tanto es necesario ajustar las reglas completas del ADA_actual de acuerdo con el dominio de atributos actualizado. ¿Cómo puede llevarse a cabo dicha tarea?

La figura 6.20.a ilustra las porciones de conocimiento cubiertas por el ADA_actual en el dominio de los atributos de $R' = \{r_1, r_2\}$. La cuestión es, ¿cómo puede extenderse dicho recubrimiento del ADA_actual en un dominio de tres atributos, $R^{\text{act}} = \{r_1, r_2, r_3\}$, cuando se desconoce la relación entre los subconjuntos $\{r_1, r_2\}$ y $\{r_3\}$? La solución se ilustra en la figura 6.20.b.

r_2 / r_1	0	1
0	c_1	c_1
1	c_2	n.a.

Figura 6.20.a. Recubrimiento del ADA_actual.

$r_3 / r_1 r_2$	00	01	11	10
0	c_1	c_2	n.a.	c_1
1	c_1	c_2	n.a.	c_1

Figura 6.20.b. Recubrimiento extendido.

A partir de la figura 6.20.b se obtiene el conjunto actualizado de reglas completas del ADA_actual del Ciclo_2. La figura 6.21 ilustra dicho conjunto.

$$\begin{aligned}
 (r_1 =^*_{0,1}) \wedge (r_2 = 0) \wedge (r_3 =^*_{0,1}) &\rightarrow c_1 \\
 (r_1 = 0) \wedge (r_2 = 1) \wedge (r_3 =^*_{0,1}) &\rightarrow c_2 \\
 (r_1 = 1) \wedge (r_2 = 1) \wedge (r_3 =^*_{0,1}) &\rightarrow \text{n.a.}
 \end{aligned}$$

Figura 6.21. Conjunto actualizado de reglas completas del ADA_actual.

A partir de aquí, las reglas que se considerarán en el presente ciclo para el ADA_actual serán las ilustradas en la figura 6.21.

Respecto a la OAT_actual, no hay necesidad de completarla, pues $R = R^{\text{act}}$ y $W_{ri} = W_{ri}^{\text{act}}$, $i=1..3$. Dicha OAT_actual se ilustra en la figura 6.17.

- A continuación se ejecuta la siguiente acción del algoritmo.

«Mirar si hay contradicciones entre el conjunto de reglas de la **OAT_actual** completado en el paso anterior y el conjunto de reglas del **ADA_actual** completado en el paso anterior» (8)

Una forma práctica para observar la existencia de contradicciones consiste en unificar los conjuntos reglas completas de la OAT_actual –figura 6.17- y del ADA_actual –figura 6.21-. Con objeto de abreviar la discusión, las reglas se disponen en una jerarquía top-down, siendo las de mayor prioridad las correspondientes a la OAT_actual. La figura 6.22 ilustra dicho conjunto de reglas.

$$\begin{aligned}
 (r_1 = 0) \wedge (r_2 = 0) \wedge (r_3 = 0) &\rightarrow c_1 \\
 (r_1 = 0) \wedge (r_2 = 1) \wedge (r_3 = 1) &\rightarrow c_2 \\
 (r_1 = 0) \wedge (r_2 = 0) \wedge (r_3 = 1) &\rightarrow c_1 \\
 (r_1 = 1) \wedge (r_2 = 0) \wedge (r_3 = 1) &\rightarrow c_1 \\
 (r_1 = ^*_{0,1}) \wedge (r_2 = 0) \wedge (r_3 = ^*_{0,1}) &\rightarrow c_1 \\
 (r_1 = 0) \wedge (r_2 = 1) \wedge (r_3 = ^*_{0,1}) &\rightarrow c_2 \\
 (r_1 = 1) \wedge (r_2 = 1) \wedge (r_3 = ^*_{0,1}) &\rightarrow \text{n.a.}
 \end{aligned}$$

Figura 6.22. Conjunto total de reglas del ciclo actual dispuestas según prioridades Top-Down.

El análisis del conjunto de reglas de la figura 6.22 permite establecer que no existen contradicciones.

- A continuación se evalúa el siguiente punto del algoritmo.

«**si** no hay contradicciones **entonces** {*actualizar el ADA_actual*}» (9)

- Al no existir contradicciones –punto (8)- se procede a la ejecución de la siguiente acción en el ámbito de la sentencia condicional.

i. «Unificar las reglas de *OAT_actual* y *ADA_actual*, de manera que las reglas de *OAT_actual* tengan mayor prioridad» (10)

Las reglas de la OAT_actual y del ADA_actual son completas y ya han sido unificadas según las prioridades top-down, tal como se muestra en la figura 6.22.

- Se procede a la ejecución de la siguiente acción del algoritmo.
- ii. «Aplicar el procedimiento inductivo, *inductive_infer*, al conjunto de reglas resultante y obtener el árbol de decisión actualizado, ***ADA_actualizado***»(11)

Para ello primero deben eliminarse las contradicciones causadas por las prioridades –a pesar de que anteriormente ya se determinó que no existían contradicciones, la consideración de la posibilidad de que existan sólo se realiza por motivos de claridad, especialmente por lo que respecta al concepto n.a.- La figura 6.23 ilustra todas las posibles contradicciones producidas entre las reglas. La Figura 6.24 ilustra un mapa libre de contradicciones.

$r_3 \setminus r_1, r_2$	00	01	11	10
0	c_1, c_1	c_2	n.a.	c_1
1	c_1, c_1	c_2, c_2	n.a.	c_1, c_1

Figura 6.23. Mapa de contradicciones completo de las reglas de la figura 6.21.

$r_3 \setminus r_1, r_2$	00	01	11	10
0	c_1	c_2	n.a.	c_1
1	c_1	c_2	n.a.	c_1

Figura 6.24. Mapa sin contradicciones de las reglas de la figura 6.22.

La figura 6.25 ilustra las reglas asociadas al mapa de la figura 6.24.

A continuación se aplica a las reglas de la figura 6.25 el procedimiento *inductive_infer* con objeto de generar un árbol de decisión para el aprendizaje con un mínimo número de ramas. La figura 6.26 ilustra el ***ADA_actualizado*** resultante.

$$\begin{aligned}
 (r_1 =^*_{0,1}) \wedge (r_2 = 0) \wedge (r_3 =^*_{0,1}) &\rightarrow c_1 \\
 (r_1 = 0) \wedge (r_2 = 1) \wedge (r_3 =^*_{0,1}) &\rightarrow c_2 \\
 (r_1 = 1) \wedge (r_2 = 1) \wedge (r_3 =^*_{0,1}) &\rightarrow \text{n.a.}
 \end{aligned}$$

Figura 6.25. Conjunto de reglas de la figura 6.24.

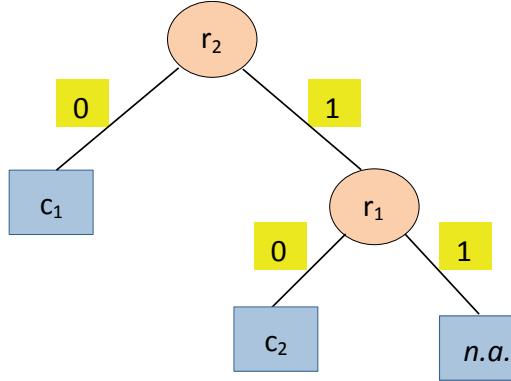


Figura 6.26. ADA_actualizado con un mínimo número de ramas.

Observando el ADA de la figura 6.26 vemos que es idéntico al de la figura 6.16, el cual era el ADA resultante del **ciclo_1**. Esto significa que los nuevos ejemplos que aparecen en el **ciclo_2**, que se usan para actualizar el ADA resultante del **ciclo_1**, realmente no aportan conocimiento significativo para que tal árbol evolucione; en otras palabras, son ejemplos redundantes que no aportan conocimiento nuevo a la base de conocimientos –árbol de decisión- actual.

Es posible y resulta sencillo obtener, cuando se desee, un árbol de decisión final –**ADF**- a partir un ADA considerado. Éste constituye la hipótesis objetivo. Este árbol de decisión final sólo contiene ejemplos que han aparecido, por tanto, no contendrá ningún ejemplo clasificado como *n.a.* Para obtener un tal árbol basta considerar una OAT que contenga todas las reglas completas correspondientes a ejemplos **aparecidos** del *ADA_actualizado* considerado y aplicar a dicha OAT el procedimiento *inductive_infer* guiado por el criterio que se considere para obtener el ADF. Por ejemplo, a partir del ADA de la figura 6.26 se obtiene el conjunto de reglas completas de la figura 6.27, correspondientes únicamente a los ejemplos que han aparecido a lo largo del proceso de aprendizaje. Recuérdese que un ADF es un árbol de **dominio cerrado**, pues considera que los ejemplos no aparecidos –*n.a.*- no tienen opciones de aparecer.

$$\begin{aligned} (r_1 = *_{0,1}) \wedge (r_2 = 0) \wedge (r_3 = *_{0,1}) &\rightarrow c_1 \\ (r_1 = 0) \wedge (r_2 = 1) \wedge (r_3 = *_{0,1}) &\rightarrow c_2 \end{aligned}$$

Figura 6.27. Conjunto de reglas completas de ejemplos del ADA_actualizado de la figura 6.26 que han aparecido.

Aplicando *inductive_infer* al conjunto de reglas de la figura 6.27, y considerando el criterio basado en obtener un árbol con un mínimo número de reglas, se obtiene al **Árbol de decisión final –ADF-** ilustrado en la figura 6.28.

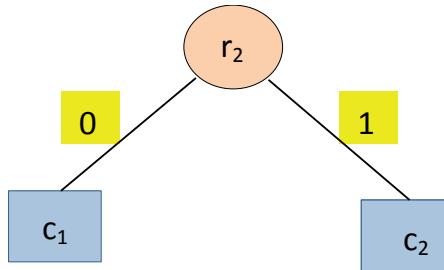


Figura 6.28. ADF con un mínimo número de reglas.

Ejemplo_2.

El siguiente ejemplo constituye otra ilustración del algoritmo del aprendizaje descrito.

Ciclo_1 –inicio del proceso de aprendizaje-. Se ejecuta por primera vez el algoritmo.

- Se presentan los nuevos ejemplos del ciclo en cuestión. Estos ejemplos constituyen el conjunto, $E = \{d_1, d_2, d_3, d_4\}$, del algoritmo. (1)

La **OAT_actual** de este ciclo se muestra en la figura 6.29. Obsérvese como no existen contradicciones entre los ejemplos de la OAT_actual.

A partir de los ejemplos de la OAT_actual se observan el conjunto de atributos, R , el conjunto de conceptos, C , y los dominios de cada atributo, W_{ri} , $i=1, 2, 3$. Así, queda:

$$R = \{r_1, r_2, r_3\},$$

$$C = \{c_1, c_2\}$$

$$W_{r1} = \{0, 1\}$$

$$W_{r2} = \{0, 1\} \quad W_{r3} = \{0, 1\}$$

D/R	r ₁	r ₂	r ₃	C
d ₁	0	0	0	c ₁
d ₂	0	1	1	c ₂
d ₃	1	0	0	c ₁
d ₄	1	1	1	c ₂

Figura 6.29. OAT_actual del Ciclo_1.

- A continuación se determina si existen contradicciones entre los elementos de E. (2)

Tal como se ha comentado en el paso anterior, observando la OAT_actual de la figura 6.29 se determina que no existen contradicciones entre los elementos de E.

- A continuación se describe el ADA_actual del Ciclo_1, el cual se pasa, junto con el conjunto E, como parámetro al algoritmo. (4)

Tal como se mencionó en el anterior ejemplo, el ADA_actual del Ciclo_1 de cualquier problema puede estar formado por un nodo raíz, cuya etiqueta puede ser cualquiera de los atributos del conjunto R de la OAT_actual, y tantas ramas desde la raíz como valores tenga

el dominio del atributo que se usa para etiquetar la propia raíz. El árbol ilustrado en la figura 6.30 constituye un ADA_actual. Existen otros posibles árboles que pueden usarse como ADA_actual, cualquiera de ellos resulta válido.

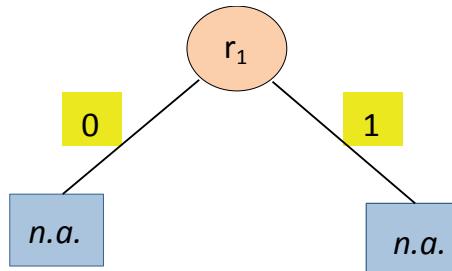


Figura 6.30. *ADA_actual* del Ciclo 1.

Las reglas del *ADA_actual* de la figura 6.30 se ilustran en la figura 6.31.

$$(r_1=0) \rightarrow \text{n.a.}$$

$$(r_1=1) \rightarrow \text{n.a.}$$

Figura 6.31. Reglas del *ADA_actual* del árbol de la figura 6.30.

El árbol de la figura 6.30 sólo contiene un atributo, cuyo dominio es binario, y en el que aparece un único concepto. Es decir:

$$R' = \{r_1\}$$

$$C' = \{\text{n.a.}\}$$

$$W_{r_1} = \{0, 1\}$$

- A continuación se ejecuta la siguiente acción del algoritmo.

«Actualizar, a partir de la **OAT_actual** y del **ADA_actual**, el conjunto de atributos, el conjunto de conceptos y los dominios de cada atributo » (5)

EL resultado es el siguiente:

$$R^{\text{act}} = R \cup R'$$

$$C^{\text{act}} = C \cup C'$$

$$W_{r_1}^{\text{act}} = W_{r_1} \cup W_{r_1}'$$

Así, queda:

$$R^{\text{act}} = \{r_1, r_2, r_3\},$$

$$C^{\text{act}} = \{c_1, c_2, \text{n.a.}\}$$

$$W_{r_1}^{\text{act}} = \{0, 1\} \quad W_{r_2}^{\text{act}} = \{0, 1\} \quad W_{r_3}^{\text{act}} = \{0, 1\}$$

- A continuación se ejecuta la siguiente acción del algoritmo:

«Obtener el conjunto de reglas completas del ADA_actual de acuerdo con el conjunto de atributos y sus dominios en el propio ADA_actual» (6)

El conjunto de reglas completas del ADA_actual se basa en la extensión del conjunto de reglas de la figura 6.31, de acuerdo con el conjunto de atributos, el conjunto de conceptos y los dominios de los atributos del propio ADA_actual, esto es, se basa en R' , C' y W_n respectivamente. En la figura 6.32 se ilustran las reglas completas del árbol de la figura 6.30. Obsérvese como en este caso particular, el conjunto de reglas de la figura 6.32 coincide el de la figura 6.31.

$$(r_1=0) \rightarrow \text{n.a.}$$

$$(r_1=1) \rightarrow \text{n.a.}$$

Figura 6.32. Conjunto de reglas completas del ADA_actual de la figura 6.30.

- A continuación se ejecuta la siguiente acción del algoritmo:

«Completar, si procede, el conjunto de reglas completas de la OAT_actual y del ADA_actual de acuerdo con los conjuntos de atributos y dominios actualizados, de manera que las reglas resultantes sean completas» (7)

En este punto debe completarse, por una parte, el conjunto de reglas de la figura 6.32 de acuerdo con los conjuntos de atributos, conceptos y dominios de los atributos actualizados, esto es, R^{act} , C^{act} , W_{ri}^{act} , $i = 1..3$, respectivamente.

La situación es similar a la del ejemplo 1. Así, a partir de la figura 6.32 se obtiene el conjunto actualizado de reglas completas de la figura 6.33.

$$(r_1=0) \wedge (r_2=^*_{0,1}) \wedge (r_3=^*_{0,1}) \rightarrow \text{n.a.}$$

$$(r_1=1) \wedge (r_2=^*_{0,1}) \wedge (r_3=^*_{0,1}) \rightarrow \text{n.a.}$$

Figura 6.33. Conjunto actualizado de reglas completas del ADA_actual de la figura 6.30.

Obsérvese como en todas las reglas de la figura 6.33 aparecen los tres atributos de conjunto actualizado, $R^{\text{act}} = \{r_1, r_2, r_3\}$, y los dominios actualizados, $W_{r1}^{\text{act}} = \{0, 1\}$, $W_{r2}^{\text{act}} = \{0, 1\}$, $W_{r3}^{\text{act}} = \{0, 1\}$.

Por otra parte, debe completarse, si procede, el conjunto de reglas completas de la OAT_actual –figura 6.29-. Puede observarse como dicha tarea no procede en este caso pues el conjunto de atributos y sus dominios coinciden con los conjuntos de atributos y dominios actualizados..

- A continuación se ejecuta la siguiente acción del algoritmo:

«Mirar si hay contradicciones entre el conjunto de reglas de la OAT_actual completado en el paso anterior y el conjunto de reglas del ADA_actual completado en el paso anterior» (8)

Se puede afirmar que no existen contradicciones entre las reglas completas de la OAT_actual -figura 6.29- y las reglas completas del ADA_actual –figura 6.33-, pues todas las reglas del ADA_actual corresponden a ejemplos que no han aparecido.

- A continuación se ejecuta la siguiente acción del algoritmo:
- «**si** no hay contradicciones **entonces** {“actualizar el *ADA_actual*”}» (9)
- Como no se detectan contradicciones en el paso (8), entonces se procede con la primera sentencia dentro de la condicional.
 - «Unificar los conjuntos de reglas completas de *OAT_actual* y *ADA_actual*, de manera que las reglas de *OAT_actual* tengan mayor prioridad»(10)

$$\begin{aligned}
 & (r_1=0) \wedge (r_2=0) \wedge (r_3=0) \rightarrow c_1 \\
 & (r_1=0) \wedge (r_2=1) \wedge (r_3=1) \rightarrow c_2 \\
 & (r_1=1) \wedge (r_2=0) \wedge (r_3=0) \rightarrow c_1 \\
 & (r_1=1) \wedge (r_2=1) \wedge (r_3=1) \rightarrow c_2 \\
 & (r_1=0) \wedge (r_2=^{*}_{0,1}) \wedge (r_3=^{*}_{0,1}) \rightarrow \text{n.a.} \\
 & (r_1=1) \wedge (r_2=^{*}_{0,1}) \wedge (r_3=^{*}_{0,1}) \rightarrow \text{n.a.}
 \end{aligned}$$

Figura 6.34. Conjunto de reglas completas dispuestas según prioridades Top-Down.

Se trata de unificar en un solo conjunto, las reglas de las figuras 6.29 y 6.33, de manera que las de la figura 6.29 tengan mayor prioridad.

La figura 6.34 muestra el resultado de acuerdo con un sistema Top-Down de prioridades.

Con objeto de aplicar el siguiente paso del algoritmo, el conjunto de reglas con prioridades de la figura 6.34 debe expresarse mediante un conjunto de reglas equivalente pero sin necesidad de establecer prioridades. Veamos la siguiente descripción general del método.

Si representamos las reglas de la figura 6.34 mediante un mapa de Karnaugh, tal como se hizo en el anterior ejemplo 1, observaremos más claramente la interpretación de los conceptos, lo cual facilitará la eliminación de las contradicciones en las reglas de la figura 6.34. Las figuras 6.35.a y 6.35.b ilustran dicha representación.

$r_3 \setminus r_1, r_2$	00	01	11	10
0	$c_1, \text{n.a.}$	n.a.	n.a.	$c_1, \text{n.a.}$
1	n.a.	$c_2, \text{n.a.}$	$c_2, \text{n.a.}$	n.a.

Figura 6.35.a. Mapa de Karnaught completo de las reglas de la figura 6.34.

$r_3 \setminus r_1, r_2$	00	01	11	10
0	c_1	n.a.	n.a.	c_1
1	n.a.	c_2	c_2	n.a.

Figura 6.35.b. Mapa de Karnaught sin contradicciones de la figura 6.34.

Finalmente, a partir del mapa de la figura 6.35.b puede obtenerse el conjunto de reglas libre de prioridades de la figura 6.36.

$$\begin{aligned}
 & (r_1=0) \wedge (r_2=0) \wedge (r_3=0) \rightarrow c_1 \\
 & (r_1=0) \wedge (r_2=1) \wedge (r_3=1) \rightarrow c_2 \\
 & (r_1=1) \wedge (r_2=0) \wedge (r_3=0) \rightarrow c_1 \\
 & (r_1=1) \wedge (r_2=1) \wedge (r_3=1) \rightarrow c_2 \\
 & (r_1=0) \wedge (r_2=0) \wedge (r_3=1) \rightarrow \text{n.a.} \\
 & (r_1=0) \wedge (r_2=1) \wedge (r_3=0) \rightarrow \text{n.a.} \\
 & (r_1=1) \wedge (r_2=1) \wedge (r_3=0) \rightarrow \text{n.a.} \\
 & (r_1=1) \wedge (r_2=0) \wedge (r_3=1) \rightarrow \text{n.a.}
 \end{aligned}$$

Figura 6.36. Conjunto de reglas completas sin prioridades.

- La siguiente acción del algoritmo manifiesta:
 - «Aplicar el procedimiento inductivo, *inductive_infer*, al conjunto de reglas resultante y obtener el árbol de decisión actualizado, *ADA_actualizado*» (11)

El procedimiento *inductive_infer* se aplica a conjuntos de reglas completas sin prioridades establecidas. En la presente situación se aplicará al conjunto de reglas de la figura 6.36.

En este ejemplo, con objeto de clarificar esta discusión, también se ha elegido el criterio basado en la creación de un árbol con un mínimo número de nodos terminales/hoja, en otras palabras, con un mínimo número de reglas. Un tal árbol se ilustra en la figura 6.37.

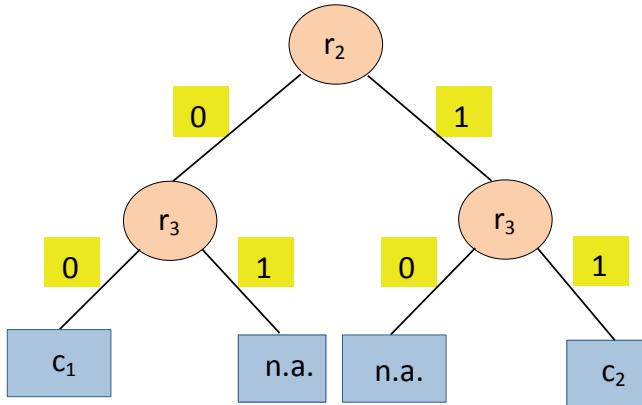


Figura 6.37. ADA_actualizado del Ciclo_1 con mínimo número de ramas.

Ciclo_2. Se ejecuta por segunda vez el algoritmo –debido a la aparición de nuevos ejemplos–.

- Considérese la aparición de nuevos ejemplos desde el entorno, lo cual provoca una nueva ejecución del algoritmo o **Ciclo_2**. Sea $E = \{d_5, d_6\}$ dicho conjunto de ejemplos, ilustrados mediante la **OAT_actual** de la figura 6.38. (1)

D \ R	r₂	r₃	C
d₅	0	1	c ₁
d₆	0	2	c ₁

Figura 6.38. OAT_actual del Ciclo_2.

A partir de la OAT_actual deben determinarse los atributos presentes –cuyo número puede variar en ciclos posteriores-, los dominios de dichos atributos –que también pueden variar en posteriores ciclos- y los conceptos asociados a los ejemplos –que también pueden variar en posteriores ciclos.

Obsérvese como hay **dos ejemplos** nuevos, los cuales vienen descritos por **dos atributos** de R, r₂ y r₃. No obstante, el atributo r₃ adopta el valor ‘2’ para el ejemplo d₆, lo que significa que su dominio ha aumentado respecto del dominio del Ciclo_1 anterior, aspecto que deberá tenerse en cuenta en este Ciclo_2. Sólo existe un único concepto, c₁, asociado a los ejemplos.

El conjunto de atributos, R, y los dominios de cada atributo, W_{ri}, i=2, 3, de la OAT_actual queda como:

$$E = \{d_5, d_6\}$$

$$R = \{r_2, r_3\},$$

$$C = \{c_1\}$$

$$W_{r2} = \{0\}$$

$$W_{r3} = \{1, 2\}$$

- A continuación se analiza la existencia de contradicciones entre los elementos de E. (2)

A partir de la OAT_actual de la figura 6.38 puede observarse como no existen contradicciones entre los nuevos ejemplos de este ciclo.

- A continuación se describe el ADA_actual, el cual se pasa, junto con el conjunto E, como parámetro al algoritmo. (4)

$$(r_2=0) \wedge (r_3=0) \rightarrow c_1$$

$$(r_2=0) \wedge (r_3=1) \rightarrow \text{n.a.}$$

$$(r_2=1) \wedge (r_3=0) \rightarrow \text{n.a.}$$

$$(r_2=1) \wedge (r_3=1) \rightarrow c_2$$

Figura 6.39. Reglas del ADA_actual de la figura 6.37.

El **ADA_actual** se ilustra en la figura 6.37, cuyas reglas se muestran en la figura 6.39.

El árbol de la figura 6.37 contiene dos atributos, r_2 y r_3 , no obstante, el conjunto de atributos, R' , del conjunto de reglas a partir del cual fue generado –figura 6.36- contiene tres atributos, r_1 , r_2 y r_3 , cuyos respectivos dominios son $\{0,1\}$, $\{0,1\}$ y $\{0,1\}$, para W'_{ri} , $i = 1..3$, respectivamente. Por tanto, esquematizando estos datos se tiene, para el ADA_actual:

$$R' = \{r_1, r_2, r_3\}$$

$$C' = \{c_1, c_2, \text{n.a.}\}$$

$$W'_{r1} = \{0, 1\}$$

$$W'_{r2} = \{0, 1\}$$

$$W'_{r3} = \{0, 1\}$$

- A continuación se ejecuta la siguiente acción del algoritmo.

«Actualizar, a partir de la **OAT_actual** y del **ADA_actual**, el conjunto de atributos, el conjunto de conceptos y los dominios de cada atributo» (5)

EL resultado es el siguiente:

$$R^{\text{act}} = R \cup R'$$

$$C^{\text{act}} = C \cup C'$$

$$W'^{\text{act}}_{ri} = W_{ri} \cup W'_{ri}$$

Así, queda:

$$R^{\text{act}} = \{r_1, r_2, r_3\},$$

$$C^{\text{act}} = \{c_1, c_2, \text{n.a.}\}$$

$$W'^{\text{act}}_{r1} = \{0, 1\}$$

$$W'^{\text{act}}_{r2} = \{0, 1\}$$

$$W'^{\text{act}}_{r3} = \{0, 1, 2\}$$

- A continuación se ejecuta la siguiente acción del algoritmo:

«Obtener el conjunto de reglas completas del ADA_actual de acuerdo con el conjunto de atributos y sus dominios en el propio ADA_actual» (6)

La figura 6.40 ilustra el conjunto de reglas completas del ADA_actual según los atributos y el dominio de los mismos a partir del que fue generado.

$$(r_1 = *_{0,1}) \wedge (r_2 = 0) \wedge (r_3 = 0) \rightarrow c_1$$

$$(r_1 = *_{0,1}) \wedge (r_2 = 0) \wedge (r_3 = 1) \rightarrow \text{n.a.}$$

$$(r_1 = *_{0,1}) \wedge (r_2 = 1) \wedge (r_3 = 0) \rightarrow \text{n.a.}$$

$$(r_1 = *_{0,1}) \wedge (r_2 = 1) \wedge (r_3 = 1) \rightarrow c_2$$

Figura 6.40. Reglas completas del ADA_actual.

- A continuación se ejecuta la siguiente acción del algoritmo:

«Completar, si procede, el conjunto de reglas completas de la **OAT_actual** y del **ADA_actual** de acuerdo con el conjunto de atributos y dominios actualizados, de manera que las reglas resultantes sean completas» (7)

En este punto debe completarse, por una parte, el conjunto de reglas del **ADA_actual** –figura 6.40- de acuerdo con los conjuntos de atributos y dominios de los atributos actualizados, esto es, R^{act} y W'^{act}_{ri} , $i = 1..3$, respectivamente.

Como $R' = R^{\text{act}}$ entonces no es necesario actualizar el conjunto de atributos del ADA_actual. Lo mismo ocurre con $W'_i = W_i^{\text{act}}$, $i = 1..2$; no obstante, $W'_{r3} \neq W_{r3}^{\text{act}}$, pues $W'_{r3} = \{0, 1\}$ y $W_{r3}^{\text{act}} = \{0, 1, 2\}$. Por tanto, hay que extender las reglas de la figura 6.40 para que cubran los valores del dominio de W_{r3}^{act} no cubiertos por W'_{r3} , esto es, los valores de $W_{r3}^{\text{act}} - W'_{r3} = \{2\}$. La figura 6.41 muestra el conjunto completado/actualizado de reglas completas del ADA_actualizado.

$$\begin{aligned}(r_1 = *_{0,1}) \wedge (r_2 = 0) \wedge (r_3 = 0) &\rightarrow c_1 \\(r_1 = *_{0,1}) \wedge (r_2 = 0) \wedge (r_3 = 1) &\rightarrow \text{n.a.} \\(r_1 = *_{0,1}) \wedge (r_2 = 1) \wedge (r_3 = 0) &\rightarrow \text{n.a.} \\(r_1 = *_{0,1}) \wedge (r_2 = 1) \wedge (r_3 = 1) &\rightarrow c_2 \\(r_1 = *_{0,1}) \wedge (r_2 = 0) \wedge (r_3 = 2) &\rightarrow \text{n.a.} \\(r_1 = *_{0,1}) \wedge (r_2 = 1) \wedge (r_3 = 2) &\rightarrow \text{n.a.}\end{aligned}$$

Figura 6.41. Conjunto completado de reglas completas del ADA_actual.

Las dos últimas reglas de la figura 6.41 pueden escribirse con una sola regla:

$$(r_1 = *_{0,1}) \wedge (r_2 = *_{0,1}) \wedge (r_3 = 2) \rightarrow \text{n.a.}$$

Cualquiera de las dos formas de escribir las anteriores reglas resulta igualmente útil para el proceso de aprendizaje.

Por otra parte, debe completarse el conjunto de reglas completas de la **OAT_actual** –figura 6.38- de acuerdo con los conjuntos de atributos y dominios de los atributos actualizados, esto es, R^{act} y W_i^{act} , $i = 1..3$, respectivamente. Tenemos que:

$$R = \{r_2, r_3\}$$

$$W_{r2} = \{0\} \quad W_{r3} = \{1, 2\}$$

Por tanto, hay que completar las reglas de la figura 6.38 para que cubran los atributos de $R^{\text{act}} - R = \{r_1\}$ según los valores de sus dominios, que en el presente caso son $W_{r1}^{\text{act}} - W_{r1} = \{0, 1\} - \{0\} = \{0, 1\}$.

En relación al atributo r_3 , puede observarse que $W_{r3}^{\text{act}} - W_{r3} = \{0, 1, 2\} - \{1, 2\} = \{0\}$.

Respecto al atributo r_2 se tiene una situación similar, pues $W_{r2}^{\text{act}} - W_{r2} = \{0, 1\} - \{0\} = \{1\}$.

Aunque en principio parezca que deberían actualizarse las reglas del paso 1 -figura 6.38- para que r_3 cubra el valor 0 y r_2 cubra el valor 1, en realidad no debe ser así. La razón de ello estriba en que una *OAT_actualizada* está formada simplemente por un conjunto de ejemplos que han sido observados, los cuales constituyen porciones concretas de conocimiento y no estructuras abstractas como los árboles de decisión. Tales ejemplos no pueden ser extendidos con nuevos valores porque simplemente no han sido observados para tales valores.

La figura 6.42 ilustra el conjunto de reglas de la OAT de la figura 6.38 una vez completada.

$$\begin{aligned}(r_1 = *_{0,1}) \wedge (r_2 = 0) \wedge (r_3 = 1) &\rightarrow c_1 \\(r_1 = *_{0,1}) \wedge (r_2 = 0) \wedge (r_3 = 2) &\rightarrow c_1\end{aligned}$$

Figura 6.42. Conjunto completado de reglas completas de la OAT_actual.

- A continuación se ejecuta la siguiente acción del algoritmo:

«Mirar si hay contradicciones entre el conjunto de reglas de la **OAT_actual** completada en el paso anterior y el conjunto de reglas del **ADA_actual** completado en el paso anterior» **(8)**

Se puede afirmar que no existen contradicciones entre las reglas completas de la OAT_actual –figura 6.42- y las reglas completas del ADA_actual –figura 6.41-. Aunque la primera regla de la figura 6.42 y la segunda regla de la figura 6.41 contengan idénticos antecedentes y diferente consecuente, no existe contradicción entre ambas reglas, pues el consecuente de la segunda regla de la figura 6.41 está etiquetado con el concepto *n.a.*, lo cual significa que el ejemplo correspondiente a dicha regla todavía no ha aparecido y por tanto no puede haber contradicción entre ambas reglas. Lo misma situación se da entre las reglas segunda de la figura 6.42 y quinta de la figura 6.41.

- A continuación se ejecuta la siguiente acción del algoritmo:

«si no hay contradicciones **entonces** {*actualizar el ADA_actual*} **(9)**

Como no se detectan contradicciones en el paso (8), entonces se procede con la primera sentencia dentro de la condicional.

i. «Unificar los conjuntos de reglas completas de *OAT_actual* y *ADA_actual*, de manera que las reglas de *OAT_actual* tengan mayor prioridad» **(10)**

Se trata de unificar en un solo conjunto, las reglas de las figuras 6.41 y 6.42, de manera que las de la figura 6.42 tengan mayor prioridad. La figura 6.43 muestra el resultado de acuerdo con un sistema Top-Down de prioridades.

$$\begin{aligned}(r_1 = *_{0,1}) \wedge (r_2 = 0) \wedge (r_3 = 1) &\rightarrow c_1 \\(r_1 = *_{0,1}) \wedge (r_2 = 0) \wedge (r_3 = 2) &\rightarrow c_1 \\(r_1 = *_{0,1}) \wedge (r_2 = 0) \wedge (r_3 = 0) &\rightarrow c_1 \\(r_1 = *_{0,1}) \wedge (r_2 = 0) \wedge (r_3 = 1) &\rightarrow n.a. \\(r_1 = *_{0,1}) \wedge (r_2 = 1) \wedge (r_3 = 0) &\rightarrow n.a. \\(r_1 = *_{0,1}) \wedge (r_2 = 1) \wedge (r_3 = 1) &\rightarrow c_2 \\(r_1 = *_{0,1}) \wedge (r_2 = 0) \wedge (r_3 = 2) &\rightarrow n.a. \\(r_1 = *_{0,1}) \wedge (r_2 = 1) \wedge (r_3 = 2) &\rightarrow n.a.\end{aligned}$$

Figura 6.43. Conjunto de reglas completas dispuestas según prioridades Top-Down.

- A continuación se ejecuta la siguiente acción del algoritmo.
- «Aplicar el procedimiento inductivo, ***inductive_infer*** al conjunto de reglas resultante y obtener el árbol de decisión actualizado, *ADA_actualizado*» (11)

El procedimiento ***inductive_infer*** se aplica a conjuntos de reglas completas sin prioridades establecidas, lo cual significa que a partir del conjunto de reglas de la figura 6.43 debe obtenerse un conjunto equivalente de reglas sin prioridades establecidas –recuérdese que en realidad no existen contradicciones, sólo se trata de «manejar» adecuadamente el símbolo *n.a.*.

Con objeto de eliminar las contradicciones de las reglas, se procederá de la misma forma que en ejemplos anteriores. Así, la figura 6.44 ilustra todas las posibles contradicciones producidas entre las reglas. La Figura 6.45 ilustra un mapa libre de contradicciones.

$r_3 \setminus r_1 r_2$	00	01	11	10
0	c_1	n.a.	n.a.	c_1
1	$c_1, n.a.$	c_2	c_2	$c_1, n.a.$
2	$c_1, n.a.$	n.a.	n.a.	$c_1, n.a.$

Figura 6.44. Ilustración completa de las reglas de la figura 6.43.

En la figura 6.46 se muestran las reglas de la figura 6.45 ya libres de contradicciones y expresadas de forma completa. A continuación, el procedimiento *inductive_infer* se aplica a dicho conjunto de reglas con objeto de generar un árbol de decisión para el aprendizaje con un mínimo número de ramas. La figura 6.47 ilustra el ADA_actualizado resultante.

$r_3 \setminus r_1 r_2$	00	01	11	10
0	c_1	n.a.	n.a.	c_1
1	c_1	c_2	c_2	c_1
2	c_1	n.a.	n.a.	c_1

Figura 6.45. Ilustración sin contradicciones de las reglas de la figura 6.44.

$$\begin{aligned}
 & (r_1 =^*_{0,1}) \wedge (r_2 = 0) \wedge (r_3 =^*_{0,1,2}) \rightarrow c_1 \\
 & (r_1 =^*_{0,1}) \wedge (r_2 = 1) \wedge (r_3 = 0) \rightarrow n.a. \\
 & (r_1 =^*_{0,1}) \wedge (r_2 = 1) \wedge (r_3 = 2) \rightarrow n.a. \\
 & (r_1 =^*_{0,1}) \wedge (r_2 = 1) \wedge (r_3 = 1) \rightarrow c_2
 \end{aligned}$$

Figura 6.46. Conjunto de reglas de la figura 6.45.

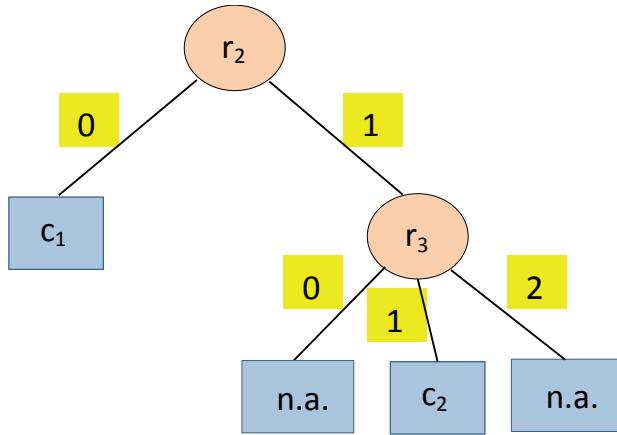


Figura 6.47. ADA_actualizado con un mínimo número de ramas.

La figura 6.48 ilustra un posible **Árbol de decisión final**. El criterio utilizado para seleccionar un tal árbol se conoce como criterio del mínimo número de reglas.

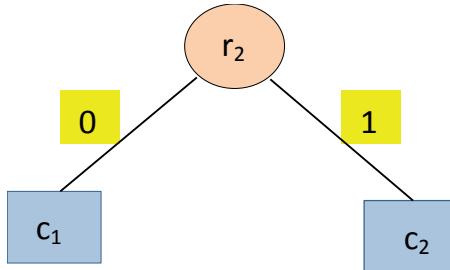


Figura 6.48. ADF con un mínimo número de ramas.

Ejemplo_3.

En este ejemplo se describe la forma de proceder del algoritmo cuando aparecen atributos con valores descritos de forma incompleta en alguno de los ciclos de aprendizaje. También se observa cómo se trata la aparición de nuevos valores del dominio de los atributos al procesar un nuevo ciclo de aprendizaje.

Ciclo_1 –inicio del proceso de aprendizaje-. Se ejecuta por primera vez el algoritmo.

- Se presentan los nuevos ejemplos del ciclo en cuestión. Estos ejemplos constituyen el conjunto, **E**, del algoritmo. (1)

Sea **E = {d₁, d₂}**. La descripción de los ejemplos en términos de atributos juntamente con los conceptos asignados a cada ejemplo, se conoce, como se menciona en el algoritmo de aprendizaje, como **OAT_actual**.

La OAT_actual se ilustra en la figura 6.49.

D/R	r ₁	r ₂	r ₃	C
d ₁	* _{0,1}	0	1	c ₂
d ₂	* _{0,1}	1	1	c ₃

Figura 6.49. OAT_actual del Ciclo_1.

A partir de los anteriores ejemplos se define el conjunto de atributos del ciclo actual, **R**, los dominios de valores simples de cada atributo, W_{r_i} , $i=1, 2, 3$ y los conceptos, **C**, asociados a los ejemplos. Así, queda:

$$E = \{d_1, d_2\}$$

$$R = \{r_1, r_2, r_3\}$$

$$C = \{c_2, c_3\}$$

$$W_{r_1} = \{0, 1\} \quad W_{r_2} = \{0, 1\} \quad W_{r_3} = \{1\}$$

Obsérvese como el atributo r_3 tiene un único valor en su dominio. De acuerdo con [3], aquellos atributos con dominios monovaluados pueden eliminarse de la OAT, pues no aportan capacidad para discernir los valores de los conceptos de C. No obstante, en el caso del aprendizaje inducto-deductivo incremental esta propiedad no es válida, pues aunque en un determinado ciclo un atributo sea monovalorado, esto no significa que en posteriores ciclos puedan aparecer nuevos valores para dicho atributo.

- A continuación se analiza la existencia de contradicciones entre los elementos de E. (2)

A partir de la OAT_actual de la figura 6.49 puede observarse como no existen contradicciones entre los ejemplos de este ciclo.

- A continuación se describe el ADA_actual, el cual se pasa, junto con el conjunto E, como parámetro al algoritmo. (4)

Recuérdese que el ADA_actual del Ciclo_1 de cualquier problema puede representarse mediante un nodo raíz, cuya etiqueta puede ser cualquiera de los atributos del conjunto R de la OAT_actual, y tantas ramas desde la raíz como valores tenga el dominio del atributo seleccionado para etiquetar la propia raíz. El árbol ilustrado en la figura 6.50 constituye un ADA_actual. Existen otros posibles árboles que pueden usarse como ADA_actual, cualquiera de ellos resulta válido.

Las reglas del ADA_actual del árbol de la figura 6.50 se ilustran en la figura 6.51.

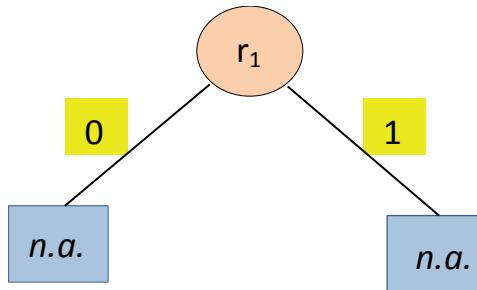


Figura 6.50. *ADA_actual* del ciclo_1.

$$(r_1=0) \rightarrow \text{n.a.}$$

$$(r_1=1) \rightarrow \text{n.a.}$$

Figura 6.51. Reglas del *ADA_actual* de la figura 6.50.

El árbol de la figura 6.50 sólo contiene un atributo, cuyo dominio es binario, y en el que aparece un único concepto, n.a.. Es decir:

$$R' = \{r_1\}$$

$$C' = \{\text{n.a.}\}$$

$$W'_{r_1} = \{0, 1\}$$

- A continuación se procede con el siguiente paso del algoritmo.

«Actualizar, a partir de la **OAT_actual** y del **ADA_actual**, el conjunto de atributos, el conjunto de conceptos y los dominios de cada atributo» (5)

Para ello se procede de la siguiente forma:

$$R^{\text{act}} = R \cup R'$$

$$C^{\text{act}} = C \cup C'$$

$$W_r^{\text{act}} = W_r \cup W'_r$$

Así, los **conjuntos actualizados** quedan como:

$$R^{\text{act}} = \{r_1, r_2, r_3\}$$

$$C^{\text{act}} = \{c_2, c_3, \text{n.a.}\}$$

$$W_{r_1}^{\text{act}} = \{0, 1\} \quad W_{r_2}^{\text{act}} = \{0, 1\} \quad W_{r_3}^{\text{act}} = \{1\}$$

- A continuación se ejecuta la siguiente acción del algoritmo:

«Obtener el conjunto de reglas completas del ADA_actual de acuerdo con el conjunto de atributos y sus dominios en el propio ADA_actual» (6)

El conjunto de reglas completas del ADA_actual se basa en la extensión del conjunto de reglas del ADA_actual –figura 6.51- de acuerdo con el conjunto de atributos, el conjunto de conceptos y los conjuntos de los dominios de los atributos del propio ADA_actual, esto es, se basa en R' , C' y W'_{r_i} respectivamente. En el presente ejemplo, el conjunto de reglas completas es idéntico al que aparece en la figura 6.51.

- A continuación se ejecuta la siguiente acción del algoritmo:

«Completar, si procede, el conjunto de reglas completas de la **OAT_actual** y del **ADA_actual**, de acuerdo con el conjunto de atributos y dominios actualizados, de manera que las reglas resultantes sean completas» (7)

Respecto a la **OAT_actual**, el conjunto de reglas es idéntico al de la figura 6.49.

Respecto al **ADA_actual**, es necesario completar su conjunto de reglas completas – figura 6.51-, puesto que en dicho conjunto no aparecen todos los atributos de R^{act} . La figura 6.52 ilustra el conjunto completado de reglas completas del **ADA_actual**.

$$\begin{aligned} (r_1=0) \wedge (r_2={}^*_{0,1}) \wedge (r_3=1) &\rightarrow n.a. \\ (r_1=1) \wedge (r_2={}^*_{0,1}) \wedge (r_3=1) &\rightarrow n.a. \end{aligned}$$

Figura 6.52. Conjunto completado de reglas completas del **ADA_actual**.

Obsérvese en la figura 6.52 como el atributo r_2 adopta el valor ${}^*_{0,1}$, pues el dominio actualizado de dicho atributo es $W_{r_2}^{act} = \{0,1\}$; mientras que r_3 adopta el valor 1, ya que su dominio actualizado es $W_{r_3}^{act} = \{1\}$, en otras palabras, el único valor posible en este ciclo de aprendizaje para r_3 es 1.

- A continuación se ejecuta la siguiente acción del algoritmo:

«Mirar si hay contradicciones entre el conjunto de reglas de la **OAT_actual** completada en el paso anterior y el conjunto de reglas del **ADA_actual** completado en el paso anterior -esto son los ejemplos que ya han aparecido en los ciclos anteriores y se describen a través del **ADA_actual**» (8)

Como el **ADA_actual** no describe ningún ejemplo que previamente haya aparecido –esto es, todos sus ejemplos se catalogan como *n.a.*-, entonces no existen contradicciones entre las reglas del **ADA_actual** y las de la **OAT_actual**. En otras palabras, los ejemplos del **ADA_actual** cuyo concepto asociado sea *n.a.* no causan ningún tipo de contradicción con los ejemplos de la **OAT_actual** a los que cubren. Así, por ejemplo, el primer ejemplo de la primera fila de la figura 6.52, $(r_1=0, r_2={}^*_{0,1}, r_3=1)$, cuyo concepto asociado es *n.a.*, tiene una intersección no vacía con los dos ejemplos de la figura 6.49 $(r_1={}^*_{0,1}, r_2=0, r_3=1)$ y $(r_1={}^*_{0,1}, r_2=1, r_3=1)$; no obstante, como el ejemplo de la figura 6.52 todavía no ha aparecido, entonces no puede causar contradicción con ninguno de los dos ejemplos de la figura 6.49. Obsérvese como ocurre exactamente lo mismo con el segundo ejemplo de la figura 6.52.

- A continuación se ejecuta la acción condicional del algoritmo:

«**si** no hay contradicciones **entonces** {“actualizar el **ADA_actual**”}» (9)

Como se describe en el punto anterior (8), no existen contradicciones entre los ejemplos de la **OAT_actual** y los ejemplos cubiertos por el **ADA_actual**.

- A continuación se ejecuta la siguiente acción del algoritmo:

- i. «Unificar las reglas de *OAT_actual* y *ADA_actual* –esto es, *OAT_actual* = *OAT_actual* U *ADA_actual*–, de manera que las reglas de *OAT_actual* tengan mayor prioridad;» (10)

El resultado es el conjunto de reglas ilustrado en la figura 6.53.

$$\begin{aligned}
 (r_1=^*_{0,1}) \wedge (r_2=0) \wedge (r_3=1) &\rightarrow c_2 \\
 (r_1=^*_{0,1}) \wedge (r_2=1) \wedge (r_3=1) &\rightarrow c_3 \\
 (r_1=0) \wedge (r_2=^*_{0,1}) \wedge (r_3=1) &\rightarrow \text{n.a.} \\
 (r_1=1) \wedge (r_2=^*_{0,1}) \wedge (r_3=1) &\rightarrow \text{n.a.}
 \end{aligned}$$

Figura 6.53. Conjunto total de reglas del *CICLO_1* dispuestas según prioridades Top-Down.

- Finalmente, se ejecuta la última sentencia del algoritmo:
- ii. «Aplicar el procedimiento inductivo, *inductive_infer*, al conjunto de reglas resultante y obtener el árbol de decisión actualizado, *ADA_actualizado*» (11)

Para ello primero deben eliminarse las «falsas» contradicciones entre las reglas de la figura 6.53, con objeto de obtener un conjunto de reglas sin contradicciones y que respete las prioridades de las reglas de dicha figura. Como se ha comentado, puede observarse que realmente no existen contradicciones entre las reglas de la *OAT_actual* y las del *AD_actual* de la figura 6.53, puesto que estas últimas tienen asociado el valor *n.a.*. No obstante, con objeto de ilustrar cómo se obtiene una base de reglas libre de contradicciones y sin prioridades establecidas entre ellas, se muestra un método sencillo a través de un mapa [18], tal como se muestra en la figura 6.54, en el cual se representan todas las posibles contradicciones producidas entre las reglas. Cada regla puede tener más de un concepto asociado, tal como se ilustra mediante una lista de conceptos en las correspondientes casillas, de manera que el concepto con prioridad más alta está situado más a la izquierda de la lista. En negrita se representan los conceptos con mayor prioridad.

La Figura 6.55 ilustra un mapa libre de contradicciones.

$r_3 \setminus r_1 \ r_2$	00	01	11	10
1	c_2 , n.a.	c_3 , n.a.	c_3 , n.a.	c_2 , n.a.

Figura 6.54. Ilustración completa de las reglas de la figura 6.53.

$r_3 \setminus r_1 \ r_2$	00	01	11	10
1	c_2	c_3	c_3	c_2

Figura 6.55. Ilustración completa sin contradicciones de las reglas de la figura 6.53.

El mapa de la figura 6.55 puede ilustrarse mediante las reglas de la figura 6.56 libres de contradicciones.

$$(r_1 = *_{0,1}) \wedge (r_2 = 0) \wedge (r_3 = 1) \rightarrow c_2$$

$$(r_1 = *_{0,1}) \wedge (r_2 = 1) \wedge (r_3 = 1) \rightarrow c_3$$

Figura 6.56. Conjunto de reglas sin contradicciones final.

Finalmente, la actualización del *ADA_actual* –Figura 6.50- a partir de los ejemplos del *CICLO_1*–figura 6.49-, se basa en aplicar el procedimiento de inferencia, *inductive_infer*, a las reglas de la figura 6.56. En este ejemplo se ha seleccionado el criterio basado en la creación de un árbol con un mínimo número de nodos terminales/hoja, en otras palabras, con un mínimo número de reglas. Pueden existir varios árboles óptimos, uno de ellos se ilustra en la figura 6.57.

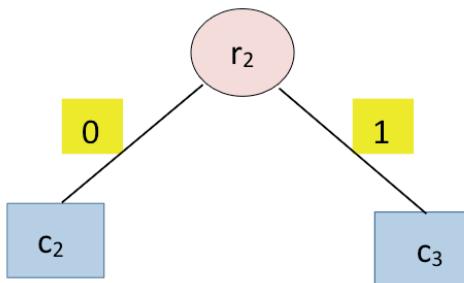


Figura 6.57. Árbol de decisión con un mínimo número de reglas –ADA_actualizado–.

Ciclo_2. Se ejecuta por segunda vez el algoritmo –debido a la aparición de nuevos ejemplos).

- Se presentan los nuevos ejemplos del ciclo en cuestión. Estos ejemplos constituyen el conjunto, **E**, del algoritmo. (1)

Sea **E** = {d₃, d₄, d₅} el conjunto de ejemplos que aparecen en este ciclo, cuya **OAT_actual** se ilustra en la figura 6.58.

D \ R	r ₁	r ₂	r ₃	C
d ₃	0	1	1	c ₃
d ₄	1	0	0	c ₁
d ₅	1	0	1	c ₂

Figura 6.58. OAT_actual del ciclo_2.

Obsérvese en la figura 6.58 como el atributo r₃ adopta el valor '0' para el ejemplo d₄, lo que significa que su dominio ha aumentado respecto de su dominio inicial en el Ciclo_1, también aparece un nuevo concepto, c₁, aspectos que deberán tenerse en cuenta en este Ciclo_2.

A partir de los anteriores ejemplos se define el conjunto de atributos del ciclo actual, **R**, los dominios de cada atributo, **W_{ri}**, i=1, 2, 3 y los conceptos, **C**, descritos. Así, queda:

$$E = \{d_3, d_4, d_5\}$$

$$R = \{r_1, r_2, r_3\}$$

$$C = \{c_1, c_2, c_3\}$$

$$W_{r1} = \{0, 1\} \quad W_{r2} = \{0, 1\} \quad W_{r3} = \{0, 1\}$$

- A continuación se analiza la existencia de contradicciones entre los elementos de E. (2)

A partir de la OAT_actual de la figura 6.58 puede observarse como no existen contradicciones entre los nuevos ejemplos de este ciclo.

- A continuación se describe el ADA_actual, el cual se pasa, junto con el conjunto E, como parámetro al algoritmo. (4)

El ADA_actual del Ciclo_2 se muestra en la figura 6.57. Dicho árbol fue obtenido a partir de las reglas de la figura 6.56. Los conjuntos R', C' y W'_ri asociados al ADA_actual son:

$$R' = \{r_1, r_2, r_3\}$$

$$C' = \{c_2, c_3\}$$

$$W'_{r1} = \{0, 1\} \quad W'_{r2} = \{0, 1\} \quad W'_{r3} = \{1\}$$

- A continuación se ejecuta el siguiente punto del algoritmo.

«Actualizar, a partir de la OAT_actual y el ADA_actual, el conjunto de atributos, el conjunto de conceptos y los dominios de cada atributo». (5)

Para ello se procede de la siguiente forma:

$$R^{\text{act}} = R \cup R'$$

$$C^{\text{act}} = C \cup C'$$

$$D_{ri}^{\text{act}} = D_{ri} \cup D_{ri}'$$

Así, los **conjuntos actualizados** quedan como:

$$R^{\text{act}} = \{r_1, r_2, r_3\}$$

$$C^{\text{act}} = \{c_1, c_2, c_3\}$$

$$W_{r1}^{\text{act}} = \{0, 1\} \quad W_{r2}^{\text{act}} = \{0, 1\} \quad W_{r3}^{\text{act}} = \{0, 1\}$$

- A continuación se ejecuta la siguiente sentencia del algoritmo.

«Obtener el conjunto de reglas completas del ADA_actual de acuerdo con el conjunto de atributos y sus dominios en el propio ADA_actual» (6)

Las reglas completas del ADA_actual se muestran en la figura 6.56.

Obsérvese como el dominio del atributo r₃ del ADA_actual de la figura 6.57 es W'_{r3} = {1}.

- A continuación se ejecuta la siguiente sentencia del algoritmo.

«Completar, si procede, el conjunto de reglas completas de la OAT_actual y del ADA_actual de acuerdo con el conjunto de atributos y dominios actualizados, de manera que las reglas resultantes sean completas» (7)

Obsérvese como el conjunto de reglas completas del ADA_actual se basa en los atributos, R' , y dominios, W'_{r_i} del ciclo de aprendizaje anterior –CICLO_1-. En dicho ciclo, los atributos r_1 y r_2 tienen un dominio binario, mientras que el atributo r_3 es monovaluado, con $W'_{r_3} = \{1\}$. Como lo que se pretende en este punto (7) del algoritmo de aprendizaje es expresar las reglas completas del ADA_actual de acuerdo con el conjunto de atributos, R^{act} y **dominios**, $W_{r_i}^{\text{act}}$, **actualizados**, entonces debemos extender el conjunto de reglas completas del ADA_actual obtenidas en el punto (6) del algoritmo, añadiendo nuevas reglas que cubran los valores no cubiertos del dominio actualizado, $W_{r_i}^{\text{act}}$. El conjunto de tales valores no cubiertos de un atributo arbitrario, r_i , se describe como: $W_{r_i}^{\text{act}} - W'_{r_i}$. En nuestro caso particular se tiene: $W_{r_3}^{\text{act}} - W'_{r_3} = \{0\}$.

La figura 6.59 ilustra el conjunto completado/actualizado de reglas completas del ADA_actual..

r_1	r_2	r_3	C
*	0	1	c_2
*	1	1	c_3
*	0	0	n.a.
*	1	0	n.a.

Figura 6.59. Conjunto de reglas completas completado de ADA_actual.

Obsérvese en la figura 6.59 como el conjunto de reglas completas del ADA_actual está formado por las reglas de la figura 6.56 a las que se han añadido otras dos reglas correspondientes a la actualización de los dominios –esto es, correspondientes a los valores del conjunto $W_{r_3} - W'_{r_3} = \{0\}$.

El conjunto de reglas de la OAT_actual no precisa ser completado, pues $R=R^{\text{act}}$ y $W_{r_i}=W_{r_i}^{\text{act}}$, $i=1..3$.

- A continuación se ejecuta la siguiente acción del algoritmo:

«Mirar si hay contradicciones entre el conjunto de reglas de la **OAT_actual** completada en el paso anterior y el conjunto de reglas del **ADA_actual** completado en el paso anterior»

(8)

Para determinar la existencia de contradicciones resulta práctico unificar en una misma tabla las reglas completas de la OAT_actual y del ADA_actual. Ambos conjuntos de reglas aparecen en las figuras 6.58 y 6.59, respectivamente. La unificación de dichas tablas se muestra en la figura 6.60. Las reglas en dicha tabla ya son completas y se han dispuesto de forma top-down, de manera que las reglas de la OAT_actual tengan mayor prioridad.

Las únicas contradicciones «aparentes» se producen en el par de filas (fila 2, fila 6), cuyos conceptos asociados son, respectivamente, (c_1 , n.a.). Tales contradicciones se conocen como **contradicciones aparentes** o **falsas contradicciones**, pues tal como se ha mencionado en anteriores ejemplos, el concepto *n.a.* no causa en realidad contradicción

con ningún otro concepto. Por lo tanto, las mencionadas filas en realidad no constituyen un par de filas confusión, por lo que podemos afirmar que la tabla de la figura 6.60 está libre de contradicciones. A pesar de lo dicho, a efectos de facilitar la comprensión del algoritmo, en el punto (10) del mismo las *contradicciones aparentes* se tratan como si fueran contradicciones reales.

r₁	r₂	r₃	C
0	1	1	c ₃
1	0	0	c ₁
1	0	1	c ₂
* 0,1	0	1	c ₂
* 0,1	1	1	c ₃
* 0,1	0	0	n.a.
* 0,1	1	0	n.a.

Figura 6.60. Unificación de los conjuntos de reglas completas de la OAT_actual y del ADA_actual.

- A continuación se ejecuta la acción condicional del algoritmo:

«**si** no hay contradicciones **entonces** {*actualizar el ADA_actual*}» (9)

No existen contradicciones en la tabla de la figura 6.60.

- A continuación se ejecuta la siguiente acción del algoritmo:

i. «Unificar los conjuntos de reglas completas de *OAT_actual* y *ADA_actual*, de manera que las reglas de *OAT_actual* tengan mayor prioridad» (10)

La tabla de la figura 6.60 ya constituye la unificación la OAT_actual y del ADA_actual, de manera que las reglas de la OAT_actual tienen mayor prioridad. No obstante, para que el procedimiento *inductive_infer* sea aplicable a la mencionada tabla, ésta debe, en el sentido sintáctico, estar libre de contradicciones, por lo que deben eliminarse las *contradicciones aparentes*.

La tabla de la figura 6.61 ilustra la tabla de la figura 6.60 libre de contradicciones.

r₁	r₂	r₃	C
0	1	1	c ₃
1	0	0	c ₁
1	0	1	c ₂
* 0,1	0	1	c ₂
* 0,1	1	1	c ₃
0	0	0	n.a.
* 0,1	1	0	n.a.

Figura 6.61. Unificación de los conjuntos de reglas de la OAT_actual y del ADA_actual sin contradicciones aparentes.

- A continuación se ejecuta la siguiente acción del algoritmo:
- ii. «Aplicar el procedimiento inductivo, *inductive_infer*, al conjunto de reglas resultante y obtener el árbol de decisión actualizado, **ADA_actualizado**» (11)

A partir de la tabla de la figura 6.61 ya se está en condiciones de aplicar *inductive_infer* con objeto de generar el ADA_actualizado. Para ello se ha considerado la generación del Existen diferentes árboles óptimos, uno de los cuales se muestra en la figura 6.62.

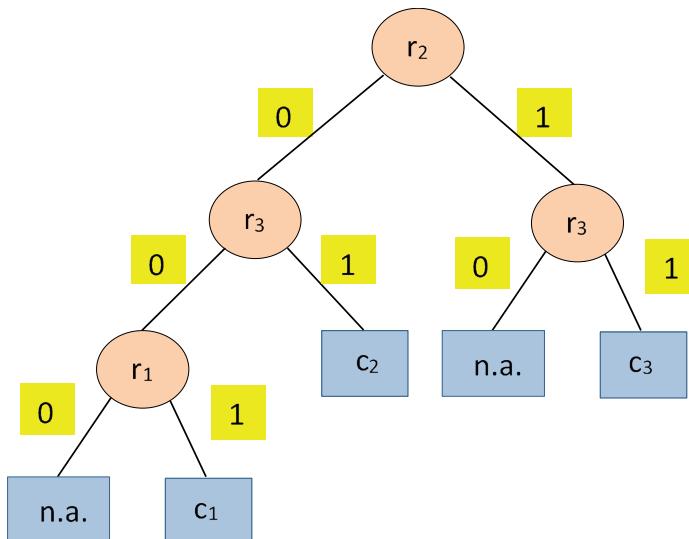


Figura 6.62. Árbol de decisión con un mínimo número de reglas –ADA_actualizado-.

Ciclo_3. Se ejecuta por tercera vez el algoritmo –debido a la aparición de nuevos ejemplos-.

- Sea $E = \{d_6, d_7\}$ el conjunto de ejemplos nuevos que aparecen en este ciclo, cuya **OAT_actual** se ilustra en la figura 6.63. (1)

D \ R	r ₁	r ₂	r ₃	C
d ₆	1	*	0	c ₁
d ₇	1	1	1	c ₃

Figura 6.63. OAT_actual del Ciclo_3.

A partir de la OAT_actual se observa::

$$E = \{d_6, d_7\}$$

$$R = \{r_1, r_2, r_3\}$$

$$C = \{c_1, c_3\}$$

$$W_{r1} = \{1\}$$

$$W_{r2} = \{0, 1\}$$

$$W_{r3} = \{0, 1\}$$

- A continuación se analiza la existencia de contradicciones entre los elementos

de E.

(2)

A partir de la OAT_actual de la figura 6.63 se observa que no existen contradicciones.

- A continuación se describe el ADA_actual, el cual se pasa, junto con el conjunto E, como parámetro al algoritmo.

(4)

El ADA_actual se muestra en la figura 6.62. Dicho árbol se obtuvo a partir del conjunto de reglas de la figura 6.61, cuyos conjuntos R', C' y W'_ri son:

$$R' = \{r_1, r_2, r_3\}$$

$$C' = \{c_1, c_2, c_3, n.a.\}$$

$$W'_{r_1} = \{0, 1\}$$

$$W'_{r_2} = \{0, 1\}$$

$$W'_{r_3} = \{0, 1\}$$

Las reglas del ADA_actual de la figura 6.62 se ilustran en la figura 6.61. Estas reglas son ya completas.

- A continuación se ejecuta el siguiente punto del algoritmo.

«Actualizar, a partir de la OAT_actual y el ADA_actual, el conjunto de atributos, el conjunto de conceptos y los dominios de cada atributo».

(5)

Téngase en cuenta que el ADA_actual de la figura 6.62 se obtuvo a partir del conjunto de reglas de la figura 6.61, cuyos conjuntos de atributos R', conceptos C' y dominios W'_ri se han descrito en el punto (4).

Por lo que respecta a la OAT_actual, ilustrada en la figura 6.63, obsérvese como el dominio del atributo r₁, W_{r1}, es monovalorado, W_{r1}={1}, a diferencia del dominio W'_ri que contiene dos valores, {0,1}.

La actualización del conjunto de atributos, R, los dominios de los atributos y los conceptos descritos en la OAT queda como:

$$R^{act} = R \cup R'$$

$$C^{act} = C \cup C'$$

$$W_{ri}^{act} = W_{ri} \cup W'_ri$$

Así queda:

$$R^{act} = \{r_1, r_2, r_3\}$$

$$C^{act} = \{c_1, c_2, c_3, n.a.\}$$

$$W_{r1}^{act} = \{0, 1\} \quad W_{r2}^{act} = \{0, 1\} \quad W_{r3}^{act} = \{0, 1\}$$

- A continuación se ejecuta la siguiente sentencia del algoritmo.

«Obtener el conjunto de reglas completas del ADA_actual de acuerdo con el conjunto de atributos y sus dominios en el propio ADA_actual»

(6)

El conjunto de reglas completas del ADA_actual está formado por la reglas de la figura 6.61. No obstante, con objeto de facilitar la comprensión de la resolución, se ha simplificado dicho conjunto de reglas. El resultado queda tal como se ilustra en la figura 6.64.

$$(r_1=0) \wedge (r_2=0) \wedge (r_3=0) \rightarrow n.a.$$

$$\begin{aligned}
 (r_1=1) \wedge (r_2=0) \wedge (r_3=0) &\rightarrow c_1 \\
 (r_1=^*_{0,1}) \wedge (r_2=0) \wedge (r_3=1) &\rightarrow c_2 \\
 (r_1=^*_{0,1}) \wedge (r_2=1) \wedge (r_3=0) &\rightarrow \text{n.a.} \\
 (r_1=^*_{0,1}) \wedge (r_2=1) \wedge (r_3=1) &\rightarrow c_3
 \end{aligned}$$

Figura 6.64. Reglas completas del **ADA_actual** de la figura 6.62.

- A continuación se ejecuta la siguiente sentencia del algoritmo.

«Completar, si procede, el conjunto de reglas completas de la **OAT_actual** y del **ADA_actual** de acuerdo con el conjunto de atributos y dominios actualizados, de manera que las reglas resultantes sean completas» (7)

Obsérvese como los **conjuntos actualizados** de atributos, R^{act} , conceptos, C^{act} , y dominios, W_{ri}^{act} , $i=1..3$, son exactamente idénticos a los conjuntos de atributos, R' , de conceptos, C' , y de dominios, W'_i , $i=1..3$, del **ADA_actual**, por lo cual no es necesario tomar medidas adicionales al respecto.

No obstante, no ocurre lo mismo con respecto los conjuntos, R , C , y W_{ri} , $i=1..3$, de la **OAT_actual**. La única diferencia reside en el dominio del atributo r_1 , W_{r1} , cuyo valor actualizado, W_{r1}^{act} , es $\{0,1\}$, mientras que el valor de W_{r1} en la OAT_actual es $\{1\}$.

¿Qué medidas deben adoptarse en este caso? No hace falta tomar ninguna medida en los casos en que el dominio de algún atributo de la OAT_actual esté incluido –esto es, sea un subconjunto– en el dominio del correspondiente atributo del ADA_actual. No obstante, sí que deben tomarse las medidas oportunas en cuanto a la extracción de las reglas del ADA_actual cuando el dominio de algún atributo del ADA_actual esté incluido en el dominio del correspondiente atributo de la OAT_actual, tal como se ha descrito en el CICLO_2 del presente ejemplo.

Por tanto, los conjuntos de reglas completas del ADA_actual y de la OAT_actual quedan tal como se ilustran en la figura 6.64 y 6.63, respectivamente.

- A continuación se ejecuta la siguiente acción del algoritmo:

«Mirar si hay contradicciones entre el conjunto de reglas de la **OAT_actual** completada en el paso anterior y el conjunto de reglas del **ADA_actual** completado en el paso anterior» (8)

En la figura 6.65 aparecen de forma unificada los conjuntos de reglas de la OAT_actual y del ADA_actual, donde puede observarse que no existen contradicciones entre dichos conjuntos de reglas. Tales reglas están dispuestas según el criterio de prioridades top-down, donde las reglas de la OAT_actual tienen mayor prioridad.

- A continuación se ejecuta la acción condicional del algoritmo:

«**si** no hay contradicciones **entonces** {“actualizar el ADA_actual”}» (9)

Al no existir contradicciones, se continúa a través del ámbito de la acción condicional.

r₁	r₂	r₃	C
1	* _{0,1}	0	c ₁
1	1	1	c ₃
0	0	0	n.a.
1	0	0	c ₁
* _{0,1}	0	1	c ₂
* _{0,1}	1	0	n.a.
* _{0,1}	1	1	c ₃

Figura 6.65. Unificación de los conjuntos de reglas de la OAT_actual y del ADA_actual.

- A continuación se ejecuta la siguiente acción del algoritmo:

«Unificar los conjuntos de reglas completas de *OAT_actual* y *ADA_actual*, de manera que las reglas de *OAT_actual* tengan mayor prioridad» **(10)**

Las tabla de la figura 6.65 ya constituye la unificación de las reglas de la OAT_actual y del ADA_actual, de manera que las reglas de la OAT_actual tienen mayor prioridad. No obstante dicha tabla contiene contradicciones «aparentes», las cuales deben ser eliminadas para que el procedimiento *inductive_infer* pueda aplicarse a la misma. La figura 6.66 ilustra la tabla 6.65 libre de confusiones aparentes.

r₁	r₂	r₃	C
1	* _{0,1}	0	c ₁
1	1	1	c ₃
0	0	0	n.a.
1	0	0	c ₁
* _{0,1}	0	1	c ₂
0	1	0	n.a.
0	1	1	c ₃

Figura 6.66. Unificación de los conjuntos de reglas de la OAT_actual y del ADA_actual libre de *contradicciones aparentes*.

Obsérvense las diferencias entre las figuras 6.65 y 6.66.

- A continuación se ejecuta la siguiente acción del algoritmo:

«Aplicar el procedimiento inductivo, *inductive_infer*, al conjunto de reglas resultante y obtener el árbol de decisión actualizado, *ADA_actualizado*» **(11)**

A partir de las reglas libres de contradicciones de la figura 6.66, se ha obtenido un ADA_actualizado con un mínimo número de ramas, tal como se muestra en la figura 6.67.

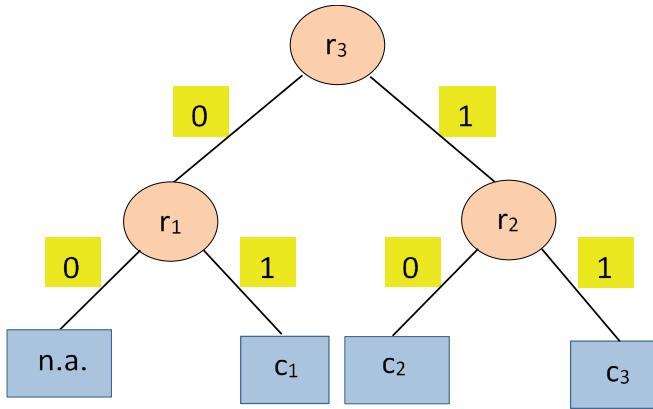


Figura 6.67. Árbol de decisión con un mínimo número de reglas –ADA_actualizado–.

La figura 6.68 ilustra un **árbol de decisión final –ADF** obtenido a partir del ADA_actualizado de la figura 6.67 mediante el criterio del *mínimo número de reglas*. Como se comentó en el EJEMPLO_2 de esta sección, un árbol de decisión final se basa en el concepto de dominio/universo cerrado, por tanto, no contendrá ningún ejemplo clasificado como *n.a.*.

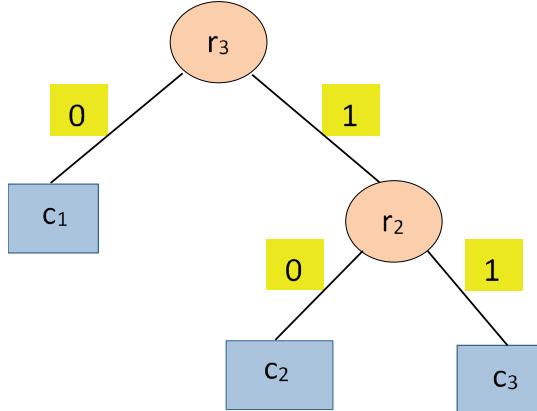


Figura 6.68. Árbol de decisión final con un mínimo número de reglas.

6.4 Conclusiones

Un análisis detallado del algoritmo de aprendizaje descrito permitirá extraer algunas propiedades importantes:

- El algoritmo implementa un proceso de **aprendizaje dinámico**, que permite en cada ciclo del mismo actualizar/enriquecer la hipótesis actual a partir de la aparición de nuevos ejemplos.
- La estructura de datos utilizada para permitir la continua evolución de la hipótesis se conoce como **árbol de decisión para el aprendizaje –ADA**. Dicha hipó-

tesis cubre todos los ejemplos aparecidos hasta el momento y los que todavía no han aparecido, dentro del marco de los dominios de los atributos que han aparecido hasta el ciclo actual de aprendizaje. Se trata de una estructura de dominio cerrado, pues cubre todas las posibilidades del dominio de los atributos hasta el ciclo de aprendizaje actual.

- La estructura de datos para soportar la hipótesis deductiva final, es decir la que se utilizará por parte del Dispositivo de *Actuación-Decisión*, se conoce como **árbol de decisión final –ADF-**. El ADF en un ciclo de aprendizaje dado puede obtenerse a partir del ADA_actualizado de dicho ciclo, para lo cual sólo se tienen en cuenta los ejemplos que han aparecido hasta el ciclo dado. Se trata también de una estructura de dominio cerrado.
- En un ciclo de aprendizaje pueden aparecer nuevos atributos no aparecidos en anteriores ciclos. A su vez, también pueden aparecer nuevos valores de los dominios de los atributos y nuevos conceptos. Por ello es necesario actualizar en cada ciclo el conjunto de atributos y sus dominios. Esta es una importante propiedad que dota de flexibilidad y potencia al proceso de adquisición de conocimiento. A través de esta propiedad se permite que el usuario aporte, en cualquier momento, el conocimiento «disponible» que considera oportuno sobre los ejemplos. Por tanto, no se obliga a que inicialmente los ejemplos vengan descritos mediante un número prefijado de atributos con dominios también prefijados.
- Cada ciclo de aprendizaje del algoritmo produce una evolución necesariamente no negativa de la hipótesis. El término «no negativa» se usa debido a que existe la posibilidad de que los ejemplos que aparecen en un determinado ciclo no aporten conocimiento cualitativo suficiente que permita mejorar la hipótesis. No obstante, la hipótesis nunca podrá empeorar en cualquiera de los ciclos. Se trata de una forma natural de aprendizaje.
- La posibilidad de inferencia inductiva sobre ejemplos incompletamente especificados es una de las grandes aportaciones al aprendizaje, además, facilita algunas de las anteriores propiedades.
- La activación de un ciclo de aprendizaje nuevo se produce por tres razones: simplemente por la aparición de nuevos ejemplos –tal como se ha descrito en los ejemplos de la sección 6.3-; simplemente por la aparición de nuevas experiencias proporcionadas por el dispositivo de Actuación-Decisión; o por ambas razones.
- La precisión de la Hipótesis es siempre del 100%, excepto en los casos en que no existan bases de atributos. En tales casos, el Sistema reclama al usuario una nueva forma de expresar/describir los conceptos asociados a los ejemplos, de manera que la nueva descripción permita la presencia de bases de atributos, permitiendo garantizar la generación de una Hipótesis libre de contradicciones. En [3] se presentan algunas ideas con respecto a la situación mencionada.

Examinando el algoritmo de aprendizaje de la sección 6.2, puede observarse como en cada ciclo de aprendizaje se prevé el caso de que no existan bases de atributos.

- El algoritmo para el aprendizaje es claro, sencillo y está bien estructurado, lo cual facilita su implementación en un lenguaje de programación.
- El algoritmo no está limitado en el tiempo, es decir, en cualquier momento en el que se produzcan nuevas experiencias por parte del Dispositivo de Actuación-Decisión y/o aparezcan nuevos ejemplos del entorno, se procederá a la evolución/actualización de la hipótesis de acuerdo con el nuevo conocimiento.
- En [4, 8, 13, 15, 16, 17] se describen algunas de las aplicaciones de la inferencia inductiva en diversos campos, como la medicina, el análisis del mercado de valores, la asistencia a personas discapacitadas y la clasificación de páginas web. Dichas aplicaciones se desarrollaron mediante la plataforma inductiva ULB-IK [9], concebida para OATs completamente especificadas y con datos de diferentes tipologías.
- La literatura sobre el aprendizaje de árboles de decisión a partir de conjuntos de reglas es muy escasa. En [7, 14] se describen dos métodos cuya orientación y objetivos son completamente diferentes a los de este trabajo.

7. EL APRENDIZAJE FEEDBACK-2

Conocido como *aprendizaje guiado por la experiencia*, se trata de un modelo de retroalimentación cuya función es la actualización de la *Hipótesis/Base de conocimientos* actual a partir de las «experiencias» obtenidas en la ejecución del Dispositivo de *Actuación-Decisión* y, posiblemente, la aparición de nuevos ejemplos del *Entorno*.

7.1 Las experiencias como consecuencia de la toma de decisiones

El dispositivo de *Actuación-Decisión* se encarga de **tomar decisiones** sobre la *Base de hechos –percepciones*-actual haciendo uso de la *Hipótesis/B.C.* Dichas **decisiones** se ejecutan sobre el **ambiente** del agente y a continuación se someten a un proceso de **verificación** por parte del dispositivo de *Evaluación de Decisiones* –*Evaluación-Decisiones-*, cuyo resultado proporcionará las «consecuencias» o «experiencias» -esto es, el éxito/fracaso, entre posiblemente otros aspectos...- de la toma de decisiones, tal como se ilustra en la figura 7.1, extensión de la figura 2.2.

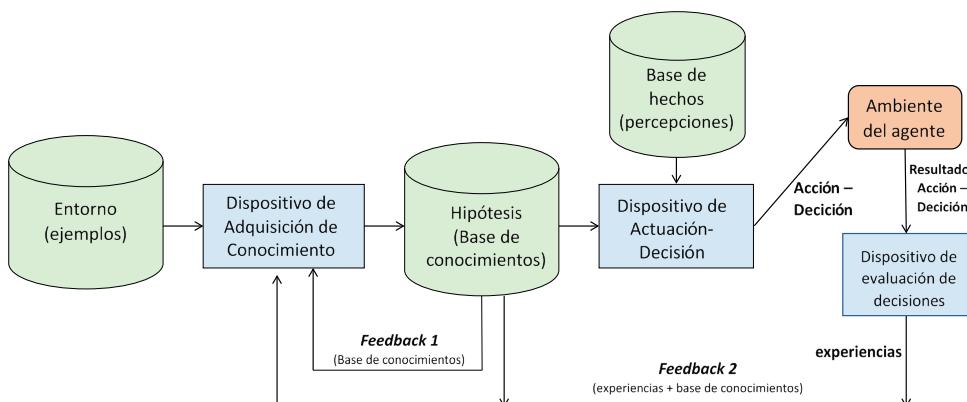


Figura 7.1. Modelo de un sistema de aprendizaje Feedback-2.

Las «experiencias» constituyen una guía en la evolución de la *Hipótesis/B.C.* hacia situaciones cada vez más satisfactorias, las cuales podrían ser cambiantes en el tiempo -y no necesariamente fijas e invariables. Debe recalcarse que el tipo de experiencias de un sistema de aprendizaje depende de la naturaleza del problema para el que fue diseñado. Así, un mismo sistema de aprendizaje puede incorporar experiencias de distinta índole y fuente, por ejemplo, experiencias guiadas por el usuario/experto del sistema, el cual puede sentirse o no satisfecho con las decisiones adoptadas por sistema de aprendizaje; experiencias generadas por el propio sistema, que puede detectar ciertas situaciones que exigen consideración; experiencias debidas a ciertos cambios en las características del entorno, etc. Estas consideraciones conducen a la reflexión de que el concepto «experiencias» puede resultar complejo y excesivamente amplio para poder ser abordado

de forma unificada. En este apartado solo se considerarán algunas cuestiones generales y relevantes de las mismas.

De acuerdo con las características del tipo de aprendizaje que se está estudiando, el proceso de toma de decisiones por parte del dispositivo de **Actuación-Decisión** se basa simplemente en «clasificar», mediante la *Hipótesis/B.C.*, los elementos del ambiente proporcionados por la *Base de hechos*. El nivel de satisfacción de dicha clasificación determinará la calidad de la *Hipótesis/B.C.* generada hasta el momento actual. Esta es la razón por la que las decisiones adoptadas deben ser evaluadas por el dispositivo de **evaluación de decisiones**.

Por tanto, la primera cuestión que surge cuando se pretende que un modelo de aprendizaje enriquezca de forma autónoma la *Hipótesis/B.C.* en la que se basa para tomar sus decisiones, es cómo dotar al modelo de su propia capacidad para mejorar la funcionalidad de la susodicha *Hipótesis/B.C.*. En otras palabras, ¿cómo conseguir que el funcionamiento prolongado del modelo de aprendizaje conduzca a una mejora paulatina de su propia función?

Los **tipos de experiencias** resultantes se han etiquetado de acuerdo con la fuente original donde se generan:

- a. Experiencias guiadas por el usuario/experto del sistema. Estas experiencias no permiten –obviamente- una total autonomía del sistema de aprendizaje, puesto que es el usuario quien decide proporcionar las experiencias al sistema. No obstante, en determinados problemas no resulta difícil establecer ciertos umbrales en algunas de las características asociadas a la *Hipótesis/B.C.* que permitan identificar cuándo deben originarse nuevas experiencias.
- b. Experiencias proporcionadas por el sistema de aprendizaje a través del dispositivo de **evaluación de decisiones** –Evaluación-Decisiones-. Tales experiencias permiten al sistema de aprendizaje una total autonomía del proceso de mejora de su función.

A continuación, se exponen, algunas **experiencias generales relevantes** a la hora de **guiar el progreso de la Hipótesis/Base de conocimientos** como consecuencia de la toma de decisiones por parte del dispositivo de *Actuación-Decisión*.

- i. Se precisa **cambiar el criterio** sobre las características que debe satisfacer la *Hipótesis/B.C.* –Árbol de Decisión-. Se debe a cambios en el mundo del agente o en las observaciones del experto. Este es el caso en que el criterio que define las características del árbol de decisión objetivo deja de ser adecuado.

Se trata de experiencias guiadas por el usuario/experto del sistema.

- ii. Algunas **decisiones** tomadas por el Dispositivo de *Actuación-Decisión* han sido **erróneas**. Esto significa que no se ha decidido correctamente en base a determinadas percepciones, lo cual se traduce en la existencia de **contradicciones** entre el conocimiento proporcionado por las *percepciones* –Base de Hechos– y el conocimiento proporcionado por la *Hipótesis* –Árbol de Decisión-. Una solución, posiblemente entre otras, a tales situaciones podría consistir en agregar

el conocimiento proporcionado por las percepciones a la *Hipótesis* mediante el dispositivo de *Adquisición de Conocimiento*, con la intención de mejorar la funcionalidad de la propia *Hipótesis*.

iii. La *Hipótesis* no permite clasificar algún ejemplo de la Base de Hechos, lo que significa que resulta «incompleta» o quizás «inconsistente». Por ejemplo, el valor de alguno de los atributos en términos del cual se describe alguna de las percepciones –ejemplos- de la *Base de Hechos* no ha aparecido en ninguno de los ejemplos del *Entorno* considerados hasta el momento. En este caso se trataría, si procede, de **completar** la *Hipótesis* a partir del conocimiento aportado por la susodicha percepción.

Nótese que para saber si una decisión tomada a partir de la *Hipótesis* ha sido errónea o incompleta –experiencias de tipo ii. o iii,-, debe conocerse cuál es la decisión real correcta.

Este tipo de experiencias puede ser proporcionado por el propio sistema de aprendizaje, sin necesidad de la asistencia del usuario/experto, siempre y cuando el *Dispositivo de Evaluación de Decisiones* tenga acceso a la hipotética decisión real correcta.

iv. Las **decisiones** tomadas por el dispositivo de *Actuación-Decisión* han sido **acertadas**. En este caso, la *Hipótesis* resulta adecuada, pues permite la toma de decisiones apropiadas. Por tanto, no parece ser necesario, en principio, tomar acciones sobre la *Hipótesis*. Hay que enfatizar de nuevo el hecho de que para saber si una decisión tomada a partir de la *Hipótesis* ha sido apropiada, debería conocerse cuál es la decisión real correcta.

Resulta importante que las experiencias de tipo iv. permitan el **refuerzo** de la validez la *Hipótesis*, lo cual puede resultar especialmente útil en los sistemas que utilizan determinados componentes probabilísticos para el cálculo de las bases de atributos que van a utilizarse.

Este tipo de experiencias puede ser proporcionado por el propio sistema de aprendizaje, sin necesidad de la asistencia del usuario/experto, siempre y cuando el *Dispositivo de Evaluación de Decisiones* tenga acceso a la hipotética decisión real correcta.

v. La **base de atributos** en la que se cimenta la hipótesis **no resulta satisfactoria** a la vista del usuario/experto, el cual decide cambiarla. Esta es una cuestión muy común entre usuarios que realmente son expertos en el campo de la aplicación. Por ejemplo, en medicina resulta común que los usuarios/expertos difieran en algunos aspectos relacionados con la base de atributos, debido sobre todo a los rígidos procedimientos en los que se basan para hacer sus diagnósticos.

El conocimiento generado por el dispositivo de *Evaluación de Decisiones* sobre las experiencias descritas y posiblemente otras experiencias que se consideren, será proporcionado al dispositivo de *Adquisición de Conocimiento*, el cual se encargará de procesarlo –juntamente con posibles ejemplos suministrados por el *Entorno* en ciclos posteriores-, garantizando la evolución de la *Hipótesis*.

Las características sobre los detalles formales de algunas de las experiencias anteriormente descritas, así como su procesamiento por parte del dispositivo de *Adquisición de Conocimiento*, se tratan en la siguiente subsección.

7.2 Implantación de experiencias: algunos casos prácticos

El proceso de aprendizaje inductivo autónomo descrito centra la atención en la provisión de nuevos ejemplos al *Dispositivo de Adquisición de Conocimiento* por parte del *Entorno*, con objeto de mejorar la *Hipótesis*. Con ello se aumenta la fiabilidad de la *Hipótesis -árbol de decisión-* obtenida en los ciclos de aprendizaje. Para poder **implantar** las **experiencias** es necesario, primero, proceder a su **especificación** en **términos formales**, y luego, determinar los **métodos** para que dichas experiencias tengan los efectos deseados. A modo de aproximación, en esta subsección se caracterizan formalmente algunos argumentos relacionados con las experiencias generales consideradas en la subsección anterior, junto con la manera de proceder por parte del DAC del modelo de aprendizaje. Tales argumentos pueden considerarse como experiencias concretas o simplemente como una parte del proceso a seguir para implantar una experiencia más general.

Argumento 1.

«Después de un preceptivo análisis de la hipótesis del ciclo de aprendizaje actual, el usuario se da cuenta que **algún atributo de la OAT resulta poco relevante** y decide eliminarlo, con objeto de que en los sucesivos ciclos de aprendizaje no se considere tal atributo»

Con este argumento se **especifica** que un determinado atributo, r_i , debe ser eliminado definitivamente de la *Hipótesis actual* y de las futuras *Hipótesis* que se generen en los ciclos de aprendizaje posteriores.

El **resultado** se alcanza en dos pasos, en el orden indicado: **primero** se eliminará el correspondiente atributo, r_i , del conjunto de atributos, R , que participan en el conjunto de reglas completas del *ADA_actual*; **segundo**, se generará una nueva *Hipótesis actual* a partir de conjunto de reglas modificado. Queda claro que las futuras hipótesis que se establezcan en ciclos de aprendizaje posteriores no podrán contener el atributo eliminado, a menos que dicho atributo vuelva a aparecer como descriptor de ejemplos del *Entorno*.

La **representación formal** de este argumento resulta sencilla. Es suficiente con una tupla del siguiente tipo:

<Arg_1, símbolo_atributo>

Donde el primer elemento de la tupla indica que se refiere al *Argumento 1* y el segundo consiste en el símbolo del atributo que se desea eliminar de la *Hipótesis*, por ejemplo r_3 .

Ejemplo. Considérese un determinado ciclo de aprendizaje, cuyo árbol de decisión final, ADF, se ilustra en la figura 7.2. Considérese que dicho ADF ha sido obtenido a partir del ADA_actual del ciclo de aprendizaje considerado. La figura 7.3 ilustra dicho ADA_actual.

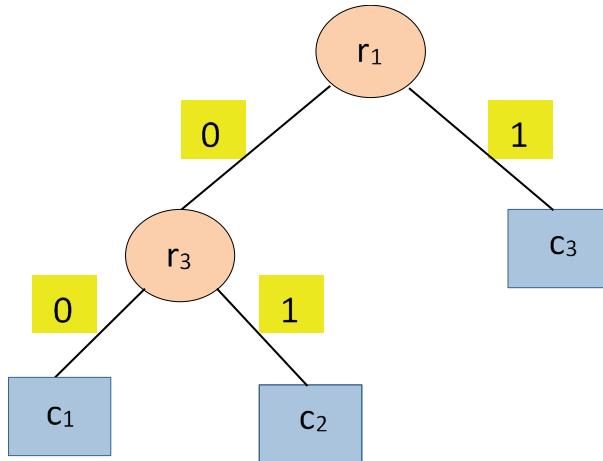


Figura 7.2 ADF del ciclo de aprendizaje dado.

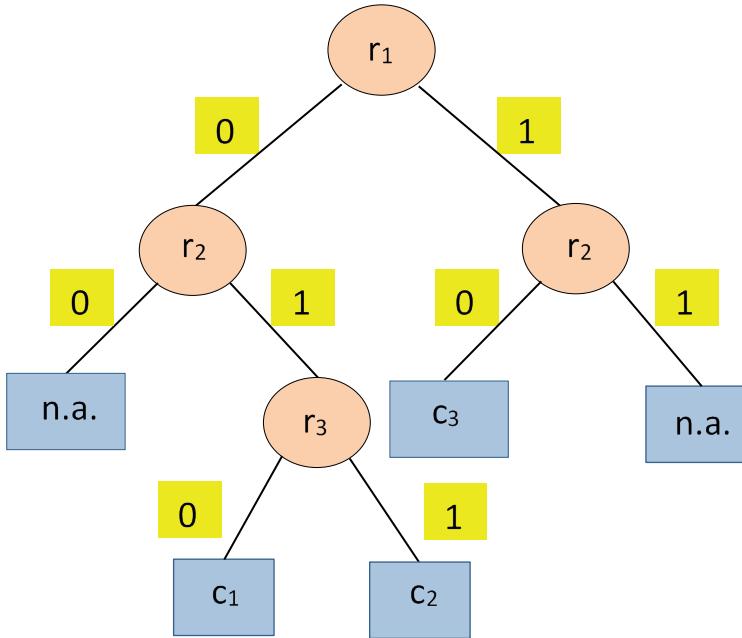


Figura 7.3 ADA_actual del ciclo de aprendizaje dado.

Supóngase que el usuario desea eliminar definitivamente el atributo r_1 de la hipótesis –ADF– actual.

Los dominios W_{r_i} de valores simples de los atributos r_1 , r_2 y r_3 son:

$$W_{r_1} = \{0, 1\},$$

$$W_{r_2} = \{0, 1\},$$

$$W_{r_3} = \{0, 1\}.$$

El conjunto de reglas completas del ADA_actual de la figura 7.3 se ilustra en la figura 7.4.

$$\begin{aligned}
 (r_1=0) \wedge (r_2=0) \wedge (r_3=^{*}_{0,1}) &\rightarrow \text{n.a.} \\
 (r_1=0) \wedge (r_2=1) \wedge (r_3=0) &\rightarrow c_1 \\
 (r_1=0) \wedge (r_2=1) \wedge (r_3=1) &\rightarrow c_2 \\
 (r_1=1) \wedge (r_2=0) \wedge (r_3=^{*}_{0,1}) &\rightarrow c_3 \\
 (r_1=1) \wedge (r_2=1) \wedge (r_3=^{*}_{0,1}) &\rightarrow \text{n.a.}
 \end{aligned}$$

Figura 7.4. Reglas del ADA_actual de la figura 7.3.

Para eliminar definitivamente el atributo r_1 , es suficiente con suprimir dicho atributo de la OAT_actual. El conjunto de reglas resultantes constituirá realmente la OAT_actual del ciclo considerado. La figura 7.5 ilustra dicha OAT.

$$\begin{aligned}
 (r_2=0) \wedge (r_3=^{*}_{0,1}) &\rightarrow \text{n.a.} \\
 (r_2=1) \wedge (r_3=0) &\rightarrow c_1 \\
 (r_2=1) \wedge (r_3=1) &\rightarrow c_2 \\
 (r_2=0) \wedge (r_3=^{*}_{0,1}) &\rightarrow c_3 \\
 (r_2=1) \wedge (r_3=^{*}_{0,1}) &\rightarrow \text{n.a.}
 \end{aligned}$$

Figura 7.5. Reglas del ADA_actual una vez eliminado el atributo r_1 .

A continuación se aplica el procedimiento inductivo, *inductive_infer*, a la OAT_actual de la figura 7.5 y, si existen bases se atributos, generará un ADA_actual de acuerdo con el criterio considerado. La figura 7.6 ilustra un ADA_actual obtenido a partir de la OAT_actual de la figura 7.5.

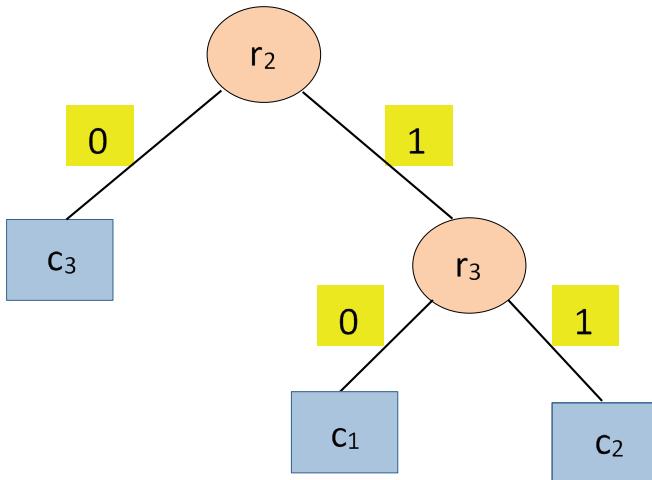


Figura 7.6. ADA_actual obtenido a partir de la OAT_actual de la figura 7.5.

Obsérvese como el ADA_actual de la figura 7.5 no contiene ningún nodo hoja categorizado mediante el símbolo n.a.. A continuación se procede al cálculo del ADF del

ciclo dado, de acuerdo con el criterio considerado. Si dicho criterio establece, por ejemplo, que el ADF debe contener un mínimo número de ramas, entonces el resultado coincidirá con el ADA_actual de la figura 7.6.

Argumento 2.

«El usuario se apercibe que **algunas reglas** que forman parte de la hipótesis -árbol de decisión- **no son correctas**, son **irrelevantes**,... o simplemente **no reflejan** el resultado que deberían revelar y decide eliminarlas, total o parcialmente, es decir, eliminando únicamente algunos ejemplos cubiertos por dichas reglas. De esta forma, las futuras hipótesis se actualizarán sin las reglas eliminadas»

Con este argumento se **especifica** que un determinado número de reglas de la Hipótesis debe eliminarse total o parcialmente, lo cual se traduce en eliminar los correspondientes ejemplos cubiertos por dichas reglas.

El **resultado** se alcanza en dos pasos, en el orden indicado: **primero** se eliminarán, del conjunto de reglas completas del *ADA_actual*, todos los ejemplos cubiertos por las reglas que el usuario quiere eliminar; **segundo**, se generará una nueva *Hipótesis actual* a partir de conjunto de reglas resultante.

La **representación formal** de este argumento también resulta sencilla. Es suficiente con una tupla del siguiente tipo:

<Arg_2, regla₁, regla₂,...,regla_k >

Donde el primer elemento de la tupla indica que se refiere al *Argumento 2* y el resto de argumentos corresponden a las reglas que se quieren eliminar.

Ejemplo. Considérese un determinado ciclo de aprendizaje, cuyo árbol de decisión final, ADF, se ilustra en la figura 7.2. El ADA_actual a partir del cual se ha obtenido el ADF se ilustra en la figura 7.3. Una vez analizado el ADF, el usuario/experto estima que la regla de la rama derecha de la raíz del árbol, etiquetada como r_e , no refleja el resultado que él considera que debería reflejar. La figura 7.7 ilustra el ADF con la regla a eliminar, total o parcialmente. Dicha regla se representa formalmente como: $(r_1=1) \rightarrow c_3$.

A continuación hay que determinar qué reglas del ADA_actual quedan cubiertas por la regla considerada del ADF. El ADA_actual se ilustra en la figura 7.3, el cual por motivos de claridad se ilustra en la figura 7.8, donde se visualizan las reglas cubiertas del ADA_actual.

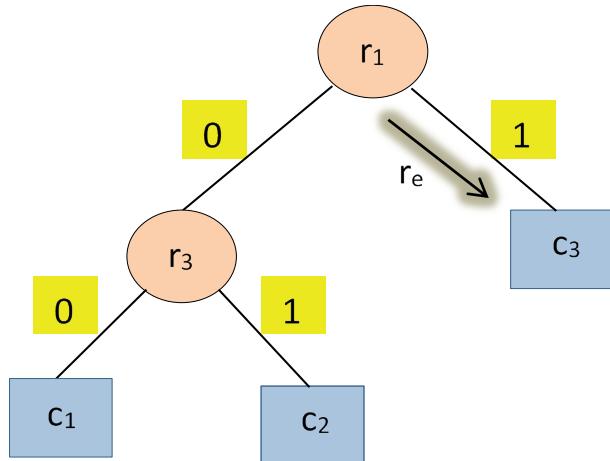


Figura 7.7. ADF con la regla r_e a eliminar.

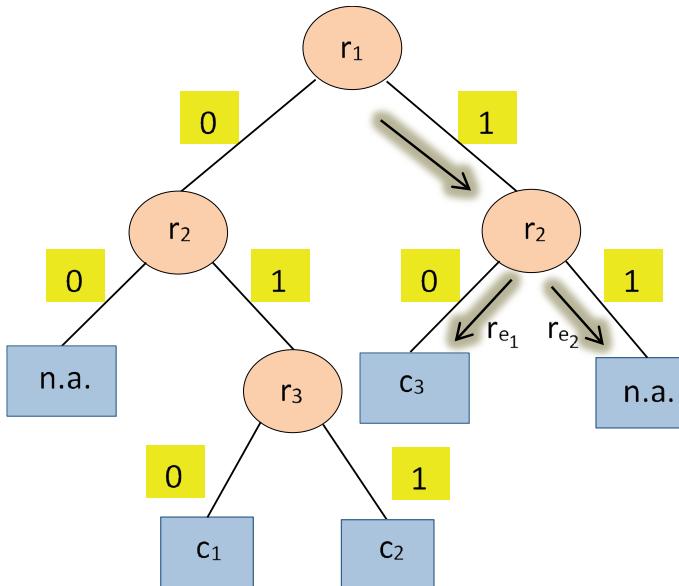


Figura 7.8. ADA_actual con las reglas cubiertas por la regla r_e .

Obsérvese en la figura 7.8 como la regla r_e cubre las reglas r_{e1} y r_{e2} . Los dominios W_{ri} de valores simples de los atributos r_1 , r_2 y r_3 son:

$$W_{r1} = \{0, 1\},$$

$$W_{r2} = \{0, 1\},$$

$$W_{r3} = \{0, 1\}.$$

El conjunto de reglas completas del ADA_actual se ilustra en la figura 7.4, la cual se vuelve a mostrar en la figura 7.9, donde se reflejan las reglas a eliminar.

$$(r_1=0) \wedge (r_2=0) \wedge (r_3=^*_{0,1}) \rightarrow \text{n.a.}$$

$$(r_1=0) \wedge (r_2=1) \wedge (r_3=0) \rightarrow c_1$$

$$(r_1=0) \wedge (r_2=1) \wedge (r_3=1) \rightarrow c_2$$

$$r_{e1}: (r_1=1) \wedge (r_2=0) \wedge (r_3=^*_{0,1}) \rightarrow c_3$$

$$r_{e2}: (r_1=1) \wedge (r_2=1) \wedge (r_3=^*_{0,1}) \rightarrow \text{n.a.}$$

Figura 7.9. Reglas del ADA_actual con las reglas r_{e1} y r_{e2} a eliminar.

Así pues, se tiene:

$$r_{e1}: (r_1=1) \wedge (r_2=0) \wedge (r_3=^*_{0,1}) \rightarrow c_3$$

$$r_{e2}: (r_1=1) \wedge (r_2=1) \wedge (r_3=^*_{0,1}) \rightarrow \text{n.a.}$$

Al analizar las dos anteriores reglas, el usuario se da cuenta que r_{e2} representa un ejemplo que todavía no ha aparecido, con lo cual dicha regla se excluye automáticamente de la eliminación, excepto en el caso que el usuario decida que la precondición del ejemplo, esto es, $(r_1=1) \wedge (r_2=1) \wedge (r_3=^*_{0,1})$, nunca puede aparecer debido a que no forma parte del universo de ejemplos.

Con objeto de analizar la regla r_{e1} , ésta se descompone en los ejemplos cubiertos por la misma, que son:

$$r_{e11}: (r_1=1) \wedge (r_2=0) \wedge (r_3=^*_{0,1}) \rightarrow c_3$$

$$r_{e12}: (r_1=1) \wedge (r_2=0) \wedge (r_3=^*_{0,1}) \rightarrow c_3$$

Al observar los ejemplos, el usuario decide que ambos no son susceptibles de aparecer nunca, pues no forman parte del universo de ejemplos, todo ello a pesar de que el *Entorno* los proporcionó en algún momento en alguno de los ciclos de aprendizaje del sistema. Como se observa, se trata de una decisión bastante radical, pero como el usuario se considera un experto, entonces hay que asumirla.

Una vez finalizado el análisis por parte del usuario/experto, éste decide que sólo se eliminarán las reglas r_{e11} y r_{e12} , mientras que la regla r_{e1} permanecerá.

Una vez tomadas las decisiones por el usuario/experto, el conjunto de reglas de la OAT_actual queda tal como se ilustra en la figura 7.10.

$$(r_1=0) \wedge (r_2=0) \wedge (r_3=^*_{0,1}) \rightarrow \text{n.a.}$$

$$(r_1=0) \wedge (r_2=1) \wedge (r_3=0) \rightarrow c_1$$

$$(r_1=0) \wedge (r_2=1) \wedge (r_3=1) \rightarrow c_2$$

$$r_{e2}: (r_1=1) \wedge (r_2=1) \wedge (r_3=^*_{0,1}) \rightarrow \text{n.a.}$$

Figura 7.10. Conjunto de reglas de la OAT_actual una vez consideradas las decisiones del usuario/experto.

A continuación se aplica el procedimiento inductivo, *inductive_infer*, a la OAT_actual de la figura 7.10 y, si existen bases se atributos, generará un posible ADA_actual de acuerdo con el criterio considerado. La figura 7.11 ilustra un ADA_actual con un mínimo número de ramas obtenido a partir de la OAT_actual de la figura 7.10.

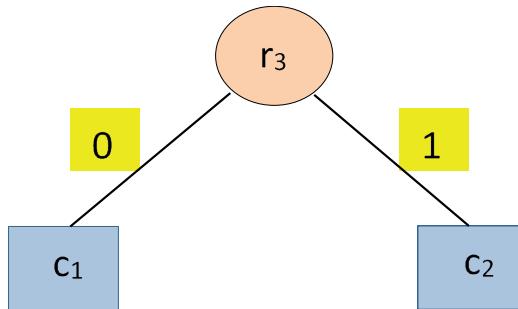


Figura 7.11. ADA_actual obtenido a partir de la OAT_actual de la figura 7.10.

Obsérvese como el ADA_actual de la figura 7.11 no contiene ningún nodo hoja etiquetado con el símbolo n.a.. A continuación se procede al cálculo del ADF del ciclo dado, de acuerdo con el criterio considerado. Si dicho criterio establece, por ejemplo, que el ADF debe contener un mínimo número de ramas, entonces el resultado coincidirá con el ADA_actual de la figura 7.11.

Una importante conclusión que se deduce de este ejemplo se refiere al hecho de que los ejemplos que hayan sido considerados como “no susceptibles de aparecer” por el usuario experto, no deben ser aceptados por el DAC en caso de que el entorno se los vuelva a proporcione en ciclos posteriores. En otras palabras, este tipo de experiencias debe suponer, además de una mejora de la Hipótesis actual -árbol de decisión-, un filtro a los ejemplos que proporciona el Entorno.

Argumento 3.

«El usuario decide proporcionar al DAC alguna nueva regla con la que mejorar la Hipótesis. Este argumento puede ser útil en los casos en que la Hipótesis haya clasificado erróneamente algún ejemplo de la Base de Hechos, de manera que el usuario desea proporcionar al DAC el conocimiento correcto sobre el/los ejemplo/s mal clasificado/s, con objeto de que éste mejore la Hipótesis. También puede resultar útil en otras situaciones»

Con este argumento se **especifica** que un determinado número de reglas proporcionadas por el usuario/experto, deben añadirse al conjunto de reglas completas del ADA_actual y a partir de ellas debe obtenerse un ADF mejorado.

Hay que tener en cuenta que, si la causa de esta acción se debe a la existencia de **errores** en alguna de las reglas de la Hipótesis, entonces el hecho de simplemente añadir nuevas reglas con la intención de mejorar la Hipótesis, producirá contradicciones en la base de reglas completas del ADA_actual. Para evitar la aparición de contradicciones, primero deben eliminarse de la base de reglas completas del ADA_actual aquellas reglas que causan el error en la clasificación y sólo después se añadirán las reglas que corregirán los errores en la clasificación. Para eliminar las reglas erróneas de la base de reglas completas puede usarse el Argumento 2.

Si, por otra parte, el usuario/experto desea añadir nuevas reglas simplemente porque cree que con ellas se mejorará la Hipótesis –aunque no haya habido errores en la clasificación-, entonces debe tener presente que las reglas añadidas no deben causar confusión con las de la base de reglas completa. De todas formas, cualquier acción en este sentido que tome el usuario debe, previamente, haber sido analizada rigurosamente.

El **resultado** se alcanza en dos pasos, en el orden indicado: **primero** se añadirán en la base de reglas completas del *ADA_actual*, aquellas que el usuario decida; **segundo**, se generará una nueva nueva *Hipótesis actual* a partir de conjunto de reglas completas resultante. Obsérvese que no se ha incluido el hecho de eliminar las reglas erróneas de la base de reglas completa, cuya acción se considera que ha sido previamente ejecutada a través del Argumento 2.

La **representación formal** de este argumento también resulta sencilla. Es suficiente con una tupla del siguiente tipo:

$\langle \text{Arg_3}, \text{regla}_1, \text{regla}_2, \dots, \text{regla}_k \rangle$

Donde el primer elemento de la tupla indica que se refiere al *Argumento 3* y el resto de argumentos corresponden a las reglas que se quieren añadir.

Ejemplo. Considérese un determinado ciclo de aprendizaje, cuyo árbol de decisión final, ADF, se ilustra en la figura 7.2. El ADA_actual a partir del cual se ha obtenido el ADF se ilustra en la figura 7.3. Una vez analizado el ADF, el usuario/experto estima que la regla de la rama derecha de la raíz del árbol, etiquetada como r_e , no refleja el resultado que él considera que debería reflejar, tal como se ilustra en la figura 7.12.

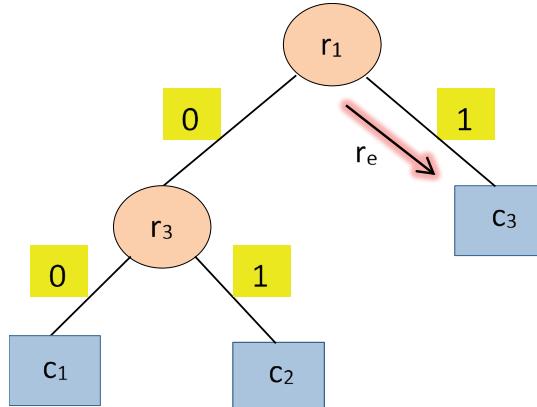


Figura 7.12. ADF con la regla, r_c , errónea.

La razón por la que la regla r_e ha sido considerada errónea se debe a que el *Dispositivo de Actuación-Decisión*, ha clasificado el hecho $(r_1=1) \wedge (r_2=0) \wedge (r_3=1)$ como perteneciente a la clase conceptual c_3 , mientras que una vez analizada dicha decisión por parte del *Dispositivo de evaluación de decisiones*, se ha comprobado que el mencionado hecho pertenece realmente a la clase conceptual c_0 .

A partir de aquí, el usuario/experto decide añadir a la base de reglas completa, la regla de la figura 7.13.

$$(r_1=1) \wedge (r_2=0) \wedge (r_3=1) \rightarrow c_2$$

Figura 7.13. Regla correctora del error en la clasificación.

Antes de ejecutar el argumento de añadir la nueva regla a la base de reglas completa, debe comprobarse qué ejemplo cubierto por la regla r_e -véase la figura 7.12- ha sido el causante del error en la clasificación.

Si examinamos la base de reglas completas del ADA_actual, que aparece en la figura 7.4, observamos que la regla r_e , cuyo formato es:

$$r_e: (r_1=1) \rightarrow c_3$$

cubre las dos últimas reglas de la base de reglas completas del ADA_actual, tal como se ilustra en la figura 7.14. No obstante, como la última de las reglas se categoriza en el conjunto de ejemplos, n.a., que todavía no han aparecido, entonces dicha regla puede obviarse. Así, la penúltima regla es la única realmente cubierta por r_e .

$$(r_1=0) \wedge (r_2=0) \wedge (r_3=^*_{0,1}) \rightarrow \text{n.a.}$$

$$(r_1=0) \wedge (r_2=1) \wedge (r_3=0) \rightarrow c_1$$

$$(r_1=0) \wedge (r_2=1) \wedge (r_3=1) \rightarrow c_2$$

$$(r_1=1) \wedge (r_2=0) \wedge (r_3=^*_{0,1}) \rightarrow c_3$$

$$(r_1=1) \wedge (r_2=1) \wedge (r_3=^*_{0,1}) \rightarrow \text{n.a.}$$

Figura 7.14. Reglas del ADA_actual cubiertas por la regla r_e .

La regla cubierta:

$$(r_1=1) \wedge (r_2=0) \wedge (r_3=^*_{0,1}) \rightarrow c_3$$

puede descomponerse en dos reglas simples, tal como muestra la figura 7.15:

$$(r_1=1) \wedge (r_2=0) \wedge (r_3=0) \rightarrow c_3$$

$$(r_1=1) \wedge (r_2=0) \wedge (r_3=1) \rightarrow c_3$$

Figura 7.15. Posibles reglas causantes del error en la clasificación.

Teniendo en cuenta que la **regla correctora** del error en la clasificación es:

$$(r_1=1) \wedge (r_2=0) \wedge (r_3=1) \rightarrow c_2$$

observamos que la segunda regla de la figura 7.15 es la que realmente **causa el error**. Por tanto, dicha regla debe ser suprimida previamente a añadir la regla correctora del error.

Una vez eliminada la regla causante del error, se añade la regla correctora de la figura 7.13. Así, la base de reglas completa queda tal como de ilustra en la figura 7.16. Obsérvese como la regla correctora se ha añadido al final de la base de reglas completa.

$$(r_1=0) \wedge (r_2=0) \wedge (r_3=^{*}_{0,1}) \rightarrow \text{n.a.}$$

$$(r_1=0) \wedge (r_2=1) \wedge (r_3=0) \rightarrow c_1$$

$$(r_1=0) \wedge (r_2=1) \wedge (r_3=1) \rightarrow c_2$$

$$(r_1=1) \wedge (r_2=0) \wedge (r_3=0) \rightarrow c_3$$

$$(r_1=1) \wedge (r_2=1) \wedge (r_3=^{*}_{0,1}) \rightarrow \text{n.a.}$$

$$(r_1=1) \wedge (r_2=0) \wedge (r_3=1) \rightarrow c_2$$

Regla
añadida

Figura 7.16. Base de reglas completas.

A partir de la base de reglas completas de la figura 7.16, *inductive_infer* procederá con objeto de inferir un ADA_actual y un ADF. La figura 7.17 ilustra un posible ADF con un mínimo número de ramas.

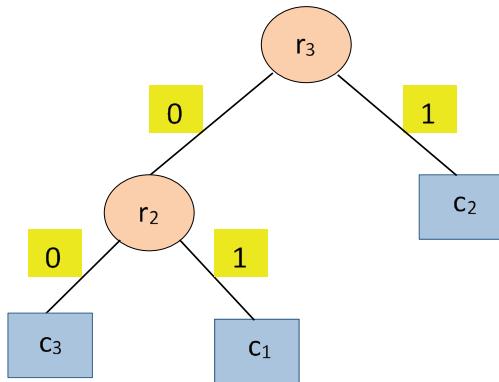


Figura 7.16. ADF con un mínimo número de ramas.

Argumento 4.

«El usuario/experto está satisfecho con los resultados que le proporciona la Hipótesis, a resultas de lo cual proporciona al DAC conocimiento de refuerzo de la confianza sobre la propia hipótesis»

Con este argumento se **especifica** que la Hipótesis es satisfactoria y el usuario/experto quiere dar soporte a la misma. La cuestión aquí es definir lo que se entiende como «dar soporte a la hipótesis». Se trata de otorgar o quitar «confianza» a la hipótesis en base dos cuestiones fundamentales:

- a) A la base de atributos en la que se basa la hipótesis.
- b) A las rutas que se siguen en el proceso de clasificación –esto es, a las características de la estructura jerárquica del árbol de decisión.

Respecto al apartado a), se trata de establecer un grado de confianza a cada uno de los atributos en los que se basa la hipótesis –árbol de decisión-. Una forma de

implementar este grado de confianza consiste en establecer un sistema de pesos sobre los atributos, de manera que cada atributo tendrá asignado un peso que representará su grado de confianza. En [3] se realiza una descripción del método, el cual puede usarse para múltiples propósitos.

En relación al apartado b), se trata de establecer de un grado de confianza sobre las ramas del árbol de decisión. Se refiere por tanto a las características jerárquicas de la hipótesis. Está claro que generar un árbol de decisión en el que cada rama tenga un máximo de confianza puede resultar en muchos casos una tarea imposible, simplemente porque dicho árbol no existe. Una hipotética solución razonable consistiría en definir un grado de confianza global del árbol, esto es, que dependa del grado de confianza de todas sus ramas en conjunto. Por ejemplo, un árbol cuyo grado de confianza depende de la suma de los grados de confianza de cada una de sus ramas; otro ejemplo, un árbol cuyo grado de confianza depende de la suma de los grados de confianza de cada una de sus ramas dividido por el número de ramas; etc...

Resulta común, especialmente en aplicaciones en el campo de la medicina, que los expertos médicos tengan ciertas reticencias sobre las Hipótesis generadas, a pesar de que dichas Hipótesis lleven a cabo una clasificación con un altísimo porcentaje de clasificaciones exitosas. Ello se debe a la influencia que los protocolos ejercen sobre los profesionales expertos.

Los grados de confianza descritos tanto en el punto a) como en el b), pueden otorgarse a los atributos o secuencias de atributos tanto desde el principio del proceso inductivo como a lo largo del mismo, mientras que la modificación de dichos grados se llevará a cabo a lo largo del proceso inductivo, de forma que el DAC conduzca el proceso inductivo hacia la generación de Hipótesis con un alto grado de confianza.

De la anterior discusión se desprende que el Argumento 4 sirve no solo para reforzar la confianza en una *Hipótesis*, también puede explotarse para disminuir el grado de confianza sobre ciertos elementos de la misma –atributos y secuencias de atributos.

El **resultado** se alcanza en dos pasos, en el orden indicado: **primero** se actualizan los grados de confianza del ADF de los apartados a) y/o b) por parte del usuario/experto; **segundo**, se generará una nueva *Hipótesis actual* a partir de conjunto de reglas completas del ADA_actual con los grados de confianza modificados.

Una **representación formal** del punto a) de este argumento podría ser:

<Arg_4.a, (símbolo atributo_i, nuevo_peso_i), (símbolo atributo_j, nuevo_peso_j),... (símbolo atributo_k, nuevo_peso_k)>

Donde el primer elemento de la tupla indica que se refiere al *Argumento 4.a* y el resto de argumentos corresponden a los pares (*atributo, nuevo_peso*) de cuyos atributos quiere modificarse el peso.

Una **representación formal** del punto b) de este argumento podría ser:

<Arg_4.b, (secuencia atributos_i, nuevo_peso_i), (secuencia atributos_j, nuevo_peso_j), ... (secuencia atributos_k, nuevo_peso_k)>

Donde el primer elemento de la tupla indica que se refiere al Argumento 4.b y el resto de argumentos corresponden a los pares (secuencia_de_atributos, nuevo_peso) de cuyas secuencias de atributos quiere modificarse el peso.

Argumento 5.

«El usuario/experto no queda satisfecho con los resultados de la hipótesis y decide **cambiar el criterio utilizado con objeto de generar una hipótesis** más adecuada a la situación»

Con este argumento se **especifica** que el criterio utilizado para generar la Hipótesis o árbol de decisión no es el que realmente desea el usuario.

El **resultado** se alcanza en dos pasos, en el orden indicado: **primero** se cambia el criterio utilizado para generar la hipótesis; **segundo**, el nuevo criterio se aplica al conjunto de reglas completas del ADA_actual no etiquetadas por el símbolo n.a., con objeto de obtener una nueva Hipótesis –ADF- que se ajuste a los requerimientos del usuario/experto.

La **representación formal** de este argumento resulta sencilla. Es suficiente con una tupla del siguiente tipo:

<Arg_5, nuevo_criterio >

Donde el primer elemento de la tupla indica que se refiere al Argumento 5 y el segundo argumento corresponde al nuevo criterio que se aplicará.

Esta función la lleva a cabo el dispositivo de adquisición de conocimiento –DAC-, a través del procedimiento inductivo, **inductive_infer**.

Ejemplo. Considérese la OAT de la figura 7.17 obtenida a partir de un hipotético ADA_actual del que se han eliminado las reglas etiquetadas con el símbolo n.a.. Supóngase que se ha definido un coste de clasificación sobre cada uno de sus atributos. Dicho coste representa el costo que supone el uso de un atributo en la clasificación de un ejemplo.

r ₁	r ₂	r ₃	r ₄	r ₅	C
1	0	0	1	1	X
1	1	0	1	1	X
1	0	1	1	1	X
1	1	1	0	0	X
1	1	0	0	0	X
1	0	1	0	0	X
0	1	0	0	1	Y
0	0	0	0	0	X
0	0	1	1	0	Y
0	1	0	1	0	Y

Figura 7.17. OAT del ejemplo.

Los costes asociados a cada atributo son los siguientes:

$c(r_1)=1$, $c(r_2)=3$, $c(r_3)=4$, $c(r_4)=4$, $c(r_5)=3$, donde $c(r_i)$ representa el coste asignado al atributo r_i .

La intención del usuario/experto es obtener un ADF que permita la clasificación de todos los ejemplos de la OAT con un mínimo costo. Para ello decidió utilizar el criterio basado en obtener primero, una base de mínimo coste y a continuación generar, a partir de la base obtenida, un árbol cuya clasificación de los ejemplos de la OAT tenga un costo mínimo. Los resultados de aplicar el mencionado criterio fueron los siguientes:

Solo existe una única base de mínimo costo, R_x , formada por los atributos $R_x=\{r_4, r_5\}$. El coste de una base se define como la suma de los costes de sus atributos, que en el presente ejemplo son $c(r_4) + c(r_5) = 4 + 3 = 7$.

A continuación se procedió a generar, a partir de la base anterior, un ADF con un mínimo costo de clasificación de todos los atributos de la OAT. El resultado se muestra en la figura 7.18. Obsérvese como en cada nodo hoja del árbol se incluye el número de ejemplos de la OAT clasificados de acuerdo a la clase representada por el nodo.

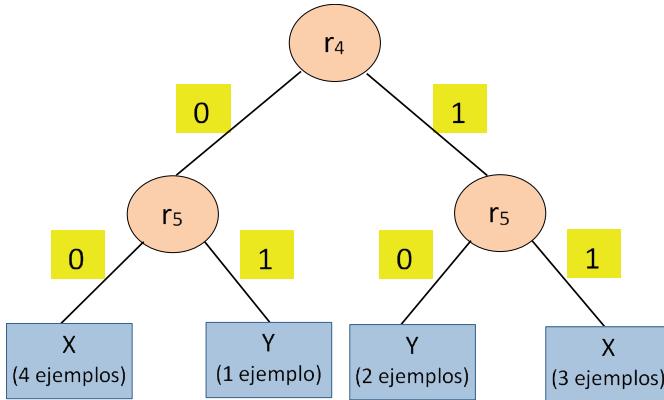


Figura 7.18. ADF con un mínimo costo de clasificación.

El costo total del ADF de la figura 7.18 es: $7*4 + 7*1 + 7*2 + 7*3 = 70$.

Una vez analizado el ADF de la figura 7.18, el usuario/experto no quedó satisfecho y decidió cambiar el criterio de generación del ADF por el siguiente: «**generar un ADF de mínimo costo de clasificación a partir de todos los atributos de la OAT**». El resultado es el árbol de la figura 7.19. Obsérvese como dicho ADF no se basa en una base de mínimo coste.

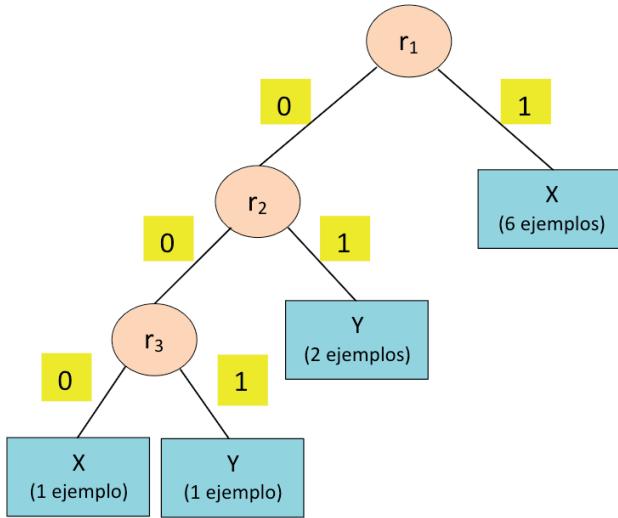


Figura 7.19. ADF con un mínimo costo de clasificación.

El costo total del ADF de la figura 7.18 es: $1*6 + 2*4 + 1*8 + 1*8 = 30$.

Se observa como el cambio de criterio por parte del usuario ha permitido obtener un cambio sensible en la mejora del ADF.

Argumento 6.

«El usuario decide **cambiar la base de atributos** en la que se basa la hipótesis por otra base, de acuerdo a algún criterio de selección de la base»

Este argumento es parecido al anterior y **especifica** que el criterio de selección de la base de atributos utilizado para posteriormente generar el ADF, debe ser substituido por otro criterio.

El **resultado** se alcanza en dos pasos, en el orden indicado: **primero** se cambia el criterio utilizado para obtener la base de atributos; **segundo**, el nuevo criterio se aplica al conjunto de reglas completas del ADA_actual no etiquetadas por el símbolo n.a., con objeto de obtener una nueva Hipótesis –ADF- que se ajuste a los requerimientos del usuario/experto.

La **representación formal** de este argumento puede realizarse con una tupla del siguiente tipo:

`<Arg_6, nuevo_criterio >`

Donde el primer elemento de la tupla indica que se refiere al *Argumento 6* y el segundo argumento corresponde al nuevo criterio que se aplicará.

Esta función la lleva a cabo el dispositivo de adquisición de conocimiento –DACK-, a través del procedimiento inductivo, **inductive_infer**.

Argumento 7.

«El usuario decide **cambiar la base de atributos** en la que se basa la hipótesis, de manera que la nueva base **incluya algún atributos** en particular»

Este argumento resulta similar al anterior y **específica** que el criterio de selección de la base de atributos utilizado para posteriormente generar el ADF, debe ser substituido por otro criterio que incluya en la base seleccionada algún atributo de libre elección del usuario/experto.

El **resultado** se alcanza en dos pasos, en el orden indicado: **primero** se cambia el criterio utilizado para obtener la base de atributos, indicando que la nueva base debe incluir los atributos de libre elección del usuario; **segundo**, el nuevo criterio se aplica al conjunto de reglas completas del ADA_actual no etiquetadas por el símbolo n.a., con objeto de obtener una nueva Hipótesis –ADF- que se ajuste a los requerimientos del usuario/experto.

La **representación formal** de este argumento puede realizarse con una tupla del siguiente tipo:

<Arg_7, nuevo_criterio, r_i, r_j, \dots, r_k >

Donde el primer elemento de la tupla indica que se refiere al *Argumento 7*, el segundo argumento corresponde al nuevo criterio que se aplicará, finalmente, r_i, r_j, \dots, r_k son los atributos que necesariamente debe incluir la nueva base.

Esta función la lleva a cabo el dispositivo de adquisición de conocimiento –DAC-, a través del procedimiento inductivo, *inductive_infer*.

Obsérvese como este argumento, al igual que el Argumento 6, sólo afectan al criterio de selección de la base de atributos y no al criterio de selección del ADF.

Ejemplo. Considérese la OAT de la figura 7.17 obtenida a partir de un hipotético ADA_actual del que se han eliminado las reglas etiquetadas con el símbolo n.a.. Considérense los mismos costes de los atributos descritos en el Argumento 5.

Supóngase que en el ciclo de aprendizaje actual se ha obtenido el ADF de la figura 7.18, aplicando el criterio del árbol de mínimo costo a partir de una base de costo mínimo, tal como se especifica en el Argumento 5. Por las razones que sea, el usuario/experto no queda satisfecho con la base de atributos, $\{r_4, r_5\}$ utilizada para obtener el ADF de la figura 7.18 y desea, primero, obtener una base de atributos de mínimo coste que incluyan los atributos r_1, r_4 . Una vez obtenida la base, el usuario/experto establece que debe generarse un árbol de decisión con un mínimo costo a partir de dicha base.

Analizando la OAT de la figura 7.17, se observa que existen dos bases de mínimo costo que incluyen los atributos r_1 y r_4 , cuyo costo es **8**: $\{r_1, r_4, r_5\}$ y $\{r_1, r_2, r_4\}$. Si no se ha especificado ninguna preferencia sobre las bases de mínimo costo obtenidas, entonces cualquiera de ellas resulta válida para proceder a la generación de un ADF de mínimo costo.

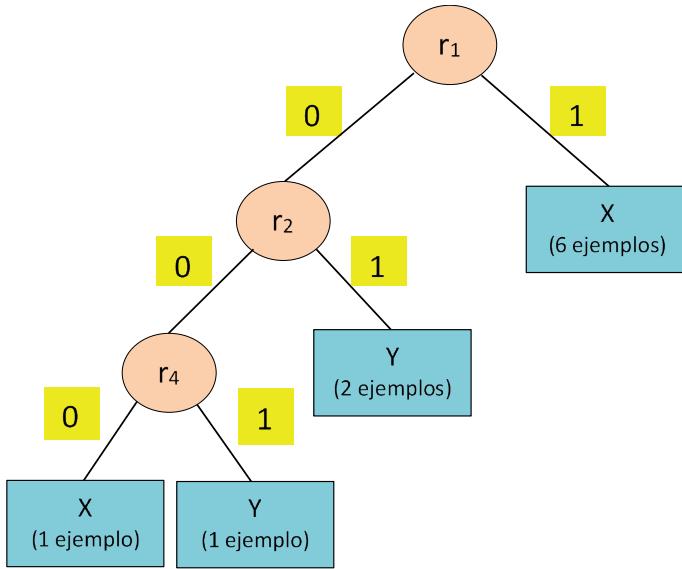


Figura 7.20 ADF de mínimo costo obtenido a partir de la base {r₁, r₂, r₄}.

Los ADF de mínimo costo obtenidos a partir de la base {r₁, r₄, r₅} tienen un costo de **32**, mientras que los ADF de mínimo costo obtenidos a partir de {r₁, r₂, r₄} tienen un costo de **30**, por tanto resultan mejores. A modo de ejemplo, en la figura 7.20 se lustra un ADF de mínimo costo obtenido a partir de la base {r₁, r₂, r₄}.

7.3 Conclusiones del aprendizaje Feedback-2

En esta sección se ha presentado, por una parte, un enfoque sobre el significado de –algunas de– las experiencias en un sistema autónomo de aprendizaje inductivo, tal como se ha enfocado el diseño del modelo de aprendizaje, siendo un tema que queda abierto a la investigación. No se han encontrado referencias bibliográficas acordes con la orientación que se le ha dado a la cuestión. Por otra parte, se han presentado algunos argumentos con objeto de implantar las experiencias consideradas a nivel práctico. En este sentido, ha podido observarse que resulta perfectamente factible llevar a la práctica las experiencias consideradas.

A continuación, se presentan algunos **aspectos relevantes** sobre la cuestión:

- i. El costo computacional de cada ciclo del proceso de aprendizaje debe ser asumible, como en el caso del aprendizaje Feedback-1.
- ii. Por lo que respecta a los cinco tipos de experiencias descritas en el punto 7.1, puede afirmarse que la complejidad computacional del sistema Feedback-2 es del mismo orden que la del sistema Feedback-1, excepto posiblemente en las experiencias de tipo i. y v., donde se requiere el cambio del criterio en el que se basa el proceso de generación de la *Hipótesis/B.C.* y de la obtención de una base

de atributos adecuada. En otras palabras, si el criterio satisfecho por la *Hipótesis/B.C.* en un momento dado puede describirse mediante un algoritmo voraz de orden polinomial –por ejemplo, el criterio ID3 [19]- mientras las experiencias obtenidas sugieren el cambio hacia un criterio de optimalidad para cuya implementación algorítmica se precisan algoritmos de orden exponencial, entonces la complejidad computacional del sistema se verá, obviamente, afectada.

iii. El uso persistente de las experiencias a lo largo del proceso de aprendizaje garantiza un «entrenamiento» **permanente** a lo largo de todo el ciclo de vida del proceso. Esta es una de las grandes aportaciones del modelo de aprendizaje presentado.

iv. El uso de las experiencias permite una mejora de diversos aspectos relacionados con la Hipótesis, desde la base de atributos en la que se basa la hipótesis, hasta la estructura jerárquica de la propia Hipótesis.

v. En el caso del Feedback-2, la precisión de la Hipótesis es, como en el Feedback-1, siempre del 100%, excepto en los casos en que no existan bases de atributos. En tales casos se procede de acuerdo con el algoritmo de la sección 6.2.

vi. El Feedback-2 permite que la Hipótesis se ajuste no sólo en función de la precisión de la hipótesis, como ocurre en la mayoría de aportaciones, también facilita las experiencias del usuario para un ajuste jerárquico/estructural de la Hipótesis. Este aspecto resulta muy importante en aquellas aplicaciones donde el usuario experto dispone de unos protocolos profesionales para la clasificación de los conceptos, como es el caso de la medicina.

vii. Una cuestión importante mencionada en el prólogo de este libro, que permitiría aumentar el grado de automatización del modelo de aprendizaje, es la creación de un sistema de pesos sobre los atributos de manera que el propio sistema sea capaz, a través de las experiencias, ajustar automáticamente dichos pesos con objeto de mejorar la hipótesis, tanto por lo que respecta a cuestiones de precisión como a cuestiones jerárquicas o estructurales de la Hipótesis.

viii. Con objeto de lograr que el procedimiento de aprendizaje inductivo descrito sea completamente autónomo, resulta fundamental que la totalidad de las experiencias sean automatizables, sin intervención alguna por parte del usuario/experto. Aunque muchas de las experiencias son realmente automatizables, resulta evidente que algunas de ellas recaen sobre la voluntad del usuario/experto. La cuestión en este punto es, ¿somos los humanos totalmente autónomos?, ¿acaso los expertos no consultan con sus agregados acerca de la toma de algunas decisiones? etc... Estas cuestiones llevan a plantearnos dos interrogantes:

a. El hecho de conseguir una total autonomía del sistema, ¿debe constituir un imperativo que guíe las posteriores investigaciones sobre el mismo?

b. Dada la diversidad y disparidad de aplicaciones del modelo, ¿no sería quizás mejor la creación de una plataforma para la edición de experiencias de manera que el diseño del modelo sea fácilmente adaptable a la aplicación para la que se concibe?

- ix. Las propiedades que debe preservar la *Hipótesis/B.C.* en cada ciclo de aprendizaje a partir del modelo Feedback-2, son las mismas que las descritas para el aprendizaje Feedback-1 –véase la sección 3.
- x. Las razones que justifican la importancia del modelo Feedback-2, así como las características de cada componente del mismo, también son las mismas que las expuestas para el aprendizaje Feedback-1 –véase la sección 3.

REFERENCIAS

- [1] D. R. Cohen and E. A. Feigenbaum (Eds), *The Handbook of Artificial Intelligence*. Vol 3, Pitman, 1982.
- [2] Z. Pawlak, *Rough Sets. Theoretical Aspects of Reasoning About Data*. Warsaw University of Technology. Institute of Computer Science, 1990.
- [3] G. Fiol, *Contribución a la Adquisición Inductiva de Conocimiento*. Doctoral dissertation. University of the Balearic Islands, 1991.
- [4] G. Fiol, J. Miró-Nicolau, "A Diagnosis Problem Approach based on Inductive Acquisition of Knowledge from Examples", *Heuristics*, 1993, pp. 54-65.
- [5] G. Fiol, J. Miró-Nicolau and J. Miró-Julià, "A New Perspective in the Inductive Acquisition of Knowledge from Examples", *Lecture Notes in Computer Science* 682, 1993, pp. 219–228.
- [6] G. Fiol, "On Qualitative Knowledge in a Rule Based Knowledge Base", *Proceedings of the IJCAI-93 workshop on Validation, Verification and Test of KBSs*, 1993, pp. 27-36.
- [7] I.F. Iman and R.S. Michalski, "Learning Decision Trees from Decision Rules: A Method and Initial Results from a Comparative Study", *Journal of Intelligent Information Systems*, 2, 1993, pp. 279–304.
- [8] G. Fiol et al, "Computer-Aided Causal Diagnosis of Ascites. Analysis of a Prototipe", *Information, Intelligence and Systems* vol. 2, 1996, pp. 1102-1107.
- [9] G. Fiol, "UIB-IK: A Computer System for Decision Trees Induction", *Lecture Notes in Artificial Intelligence* 1609, 1999, pp. 601-611.
- [10] G. Fiol, "Inductive Learning from Incompletely Specified Examples", *Frontiers in Artificial Intelligence and Applications* 100, 2003, pp. 286–295.
- [11] G. Fiol, "Learning from Incompletely Specified Object Attribute Tables with Continuous Attributes", *Frontiers in Artificial Intelligence and Applications* 113, 2004, pp. 145–152.
- [12] S. Russell and P. Norvik, *Inteligencia Artificial. Un enfoque moderno*, 2^a Edición, Pearson Educación S.A., España, 2004.
- [13] G. Fiol, D. Arellano, F. J. Perales, P. Bassa and M. Zanlongo, "The Intelligent Butler: A Virtual Agent for Disabled and Elderly People Assistance", *Advances in Soft Computing* vol. 50/2009, 2008, pp. 375-384.
- [14] A. Abdelhalim and I. Traore, "A New Method for Learning Decision Trees from Rules", *Proceedings of the Eighth International Conference on Machine Learning and Applications*, 2009, pp. 693-698.
- [15] G. Fiol, M. Miró-Julià, "A Contribution for Elderly and Disabled Care Using Intelligent Approaches", *Lecture Notes in Computer Science* 5518, 2009, pp. 902-905.
- [16] G. Fiol and M. Miró, "Stock Market Analysis using Data Mining Techniques: a Practical Application", *International Journal of Artificial Intelligence* vol. 6, num. S11, 2011, pp. 129-143.

- [17] G. Fiol, M. Miró and E. Herraiz, "Data Mining Techniques for Web Page Classification", Advances in Intelligent and Soft Computing vol. 89, 2011, pp. 61-68.
- [18] G. Fiol, "On Discovering Qualitative Knowledge in Rule Based Knowledge Bases. An Intelligent Approach". Proceedings of CISTI'2020 - 15th Iberian Conference on Information Systems and Technologies, 2020.
- [19] J. R. Quinlan: Induction of Decision Trees. Machine Learning, Vol. 1, No. 1, 1986, pp. 81-106.



- 🌐 www.atenaeditora.com.br
- ✉️ contato@atenaeditora.com.br
- 📷 [@atenaeditora](https://www.instagram.com/atenaeditora)
- ⬇️ www.facebook.com/atenaeditora.com.br



- 🌐 www.atenaeditora.com.br
- ✉️ contato@atenaeditora.com.br
- 📷 [@atenaeditora](https://www.instagram.com/atenaeditora)
- ⬇️ www.facebook.com/atenaeditora.com.br