Capítulo 2 RAZONAMIENTO APROXIMADO EN MODELADO DEL ALUMNO

2.1 Introducción

En el capítulo anterior hemos visto que el proceso de construir y mantener un modelo del alumno se basa en inferir a partir de sus interacciones con el sistema (respuestas a las preguntas planteadas, pantallas visitadas, etc.) cuál es su estado de conocimiento. Aparte de lo complicado que puede resultar realizar este tipo de inferencias, hay varias fuentes de incertidumbre que pueden dificultarlo aún más. En efecto, la información que pueda proporcionar el comportamiento del alumno es incierta, dada la gran cantidad de factores que pueden influir en él. Una respuesta incorrecta puede deberse a muchas causas diferentes, como errores de concepto, falta de conocimiento, deficiencias en la adquisición de habilidades, pero también a errores en los cálculos o incluso a un fallo al elegir la respuesta correcta. De la misma forma, una respuesta correcta puede demostrar que el alumno ha alcanzado cierto nivel de conocimiento, pero también puede deberse a haber acertado por casualidad, como puede ocurrir sobre todo cuando se plantean preguntas tipo test. Además, si el objetivo del sistema es la enseñanza no basta sólo con poder clasificar una respuesta como correcta o incorrecta sino que también es importante saber *por qué* esa pregunta fue respondida correcta o incorrectamente, ya que de otro modo será imposible seleccionar la estrategia instructora más adecuada para la situación actual del alumno.

En Inteligencia Artificial (IA) se han desarrollado varias teorías para razonamiento aproximado. Revisaremos brevemente los enfoques más significativos, utilizando ejemplos de modelado del alumno. Posteriormente discutiremos cómo estas técnicas se han aplicado en el problema del modelado del alumno, y compararemos estas técnicas entre sí presentando sus ventajas e inconvenientes.

2.2 Técnicas de razonamiento aproximado

En este apartado presentaremos de una forma muy breve las diferentes técnicas de razonamiento aproximado que se han aplicado al problema de modelado del alumno. Con esta presentación no se pretende hacer una descripción exhaustiva ni un análisis detallado de dichas técnicas, sino más bien presentar de forma introductoria los aspectos básicos de cada teoría para después poder analizar las distintas aplicaciones que se han hecho al modelado del alumno.

2.2.1 Sistemas basados en reglas (MYCIN)

Quizás la primera teoría que se aplicó con éxito para el problema de tratamiento de la incertidumbre en IA fue el modelo de los *factores de certeza*, tal como se desarrolló para el sistema MYCIN (Shortlife, 1976), un sistema experto que diagnostica enfermedades infecciosas. En este modelo la información se estructura en *hechos* y *reglas* (afirmaciones de la forma SI-ENTONCES). Asociados a estos hechos y reglas aparecen los factores de certeza, que son números entre –1 y 1 que se usan para expresar el grado de creencia de dos formas distintas:

a) Para expresar el grado de creencia en una hipótesis, dada la evidencia disponible hasta el momento.
b) Para indicar el grado de creencia en una conclusión que se establece a partir de una premisa en una regla.

Un factor de creencia cercano a 1 implica que la evidencia disponible apoya fuertemente la hipótesis. Un factor de certeza cercano a –1 implica que la evidencia disponible apoya la negación de la hipótesis. Un factor de certeza de 0 indica que la evidencia disponible no apoya ni la hipótesis ni su negación. Un factor de certeza de una regla se usa para expresar la confianza en determinada agrupación antecendente-consecuente.

Veamos un ejemplo sencillo:

Regla 1: SI el alumno conoce el concepto 1, y
proponemos al alumno una pregunta sobre los conceptos 1 y 2, y
la respuesta del alumno no es correcta,
ENTONCES
el alumno no conoce el concepto 2.

Supongamos que el factor de certeza CF de la regla es 0.6, y que los factores de certeza de las hipótesis son:

h_1: el alumno conoce el concepto1, $CF(h_1) = 0.8$
h_2: hacemos al alumno una pregunta sobre los conceptos 1 y 2, $CF(h_2) = 1$
h_3: la respuesta del alumno no es correcta, $CF(h_3) = 1$

Esto quiere decir que tenemos una creencia de 0.8 en que el alumno conoce el concepto 1, y que le hemos propuesto una pregunta relativa a los conceptos 1 y 2 que no ha sabido contestar adecuadamente. En este caso, la tarea de diagnóstico consistiría en determinar la creencia que tendría el sistema en que el alumno no conozca el concepto 2 (h_4). Para responder a esta pregunta, Buchanan y Shortlife desarrollaron en (Buchanan & Shortlife, 1984) reglas para combinar la evidencia y actualizar las creencias[1], intentando imitar el modo de razonamiento humano en este contexto. En nuestro ejemplo, parece lógico que nuestra creencia en $h_1 \wedge h_2 \wedge h_3$ sea igual al mínimo de los tres factores de certeza, es decir, 0.8, y que entonces la creencia en h_4 sea 0.48, que es exactamente lo que hacen las reglas de actualización de MYCIN.

La principal ventaja de este enfoque es que los cálculos que hay que realizar para la propagación de la incertidumbre son muy fáciles de comprender, realizar e implementar. Aunque MYCIN tuvo mucho éxito en su dominio (diagnóstico médico), Heckerman demostró no sólo que el modelo contiene graves incoherencias, sino que es imposible construir un modelo coherente de factores de certeza (Heckerman, 1986).

2.2.2 Lógica difusa

En la sección anterior hemos discutido la representación de la incertidumbre como grado de creencia. La lógica difusa (Zadeh, 1965) es otro enfoque para cuantificar grados de conocimiento, pero en un sentido diferente: se relaciona con la vaguedad y la imprecisión, que son elementos inherentes en el lenguaje natural. Por ejemplo, es habitual el uso de frases como "Juan es *bastante bueno* en Matemáticas, por tanto será capaz de resolver este problema que no es *demasiado difícil*". En esta sección introduciremos los conceptos básicos sobre lógica difusa utilizando ejemplos sencillos. Una buena introducción a la lógica difusa y sus aplicaciones es (Mendel, 1995), y una descripción más completa se puede encontrar en (Dubois & Prade, 1980).

Para representar la imprecisión, la lógica difusa utiliza los siguientes conceptos:

- *Conjuntos difusos*. Un conjunto difuso A es un conjunto cuya función característica o función de pertenencia m_A toma valores en el intervalo [0,1].

[1] Una descripción sencilla de las reglas aparece en (González y Dankel, 1993).

Supongamos que queremos determinar el grado de dificultad de una pregunta, y que tenemos el tanto por ciento de alumnos que la han contestado correctamente. Definimos entonces cuatro (por ejemplo) conjuntos difusos: *Difícil, No demasiado difícil, Bastante fácil y Fácil.* Si representamos en el eje de abscisas el tanto por ciento de alumnos que responden correctamente a la pregunta y en el eje de ordenadas el valor de la función de pertenencia, obtenemos la gráfica de las funciones de pertenencia que aparece representada en la Figura 2.1:

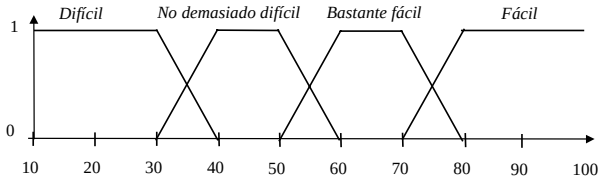

Figura 2.1 Funciones de pertenencia para los conjuntos difusos

Los conjuntos difusos y las funciones de pertenencia difusas pueden utilizarse de dos formas diferentes:

a) Para estimar grados de pertenencia a un conjunto. Por ejemplo, si sabemos que sólo el 35% de los alumnos respondieron correctamente a la pregunta, ¿en qué grado es *difícil* la pregunta?

b) Para expresar posibilidades en una situación con información incompleta. Por ejemplo, si decimos que una pregunta es *fácil*, ¿cuántos alumnos la responderán correctamente? En este caso, podemos interpretar la función de pertenencia $m_{fácil}$ como una distribución de posibilidad que indica preferencias en los valores que puede tomar esta variable.

Las operaciones sobre conjuntos difusos (unión, intersección, etc.) se definen como análogos a las operaciones correspondientes en conjuntos ordinarios.

- *Variables difusas*. Una variable difusa *A* es una variable que toma como valores conjuntos difusos. En nuestro ejemplo, podemos definir una variable *X* = "grado de dificultad de una pregunta", pudiendo entonces *X* tomar cuatro valores posibles: *Difícil, No demasiado difícil, Bastante fácil y Fácil.*

- *Relaciones difusas,* que son conjuntos difusos definidos sobre el conjunto producto. Por ejemplo, podemos definir una relación difusa como "la dificultad de las preguntas *X* e *Y* es la misma" en términos del tanto por ciento de alumnos que dan respuesta correcta a cada una de las preguntas. Como ejemplo, en la Tabla 2.1 damos una posible función de pertenencia para esta relación difusa:

X/Y	0%	25%	50%	75%	100%
0%	1	0.3	0.01	0	0
25%	0.3	1	0.3	0.01	0
50%	0.01	0.3	1	0.3	0.01
75%	0	0.01	0.3	1	0.3
100%	0	0	0.01	0.3	1

Tabla 2.1 Función de pertenencia de la relación difusa X=Y

- *Reglas difusas*, que relacionan dos o más afirmaciones difusas. Las reglas difusas se utilizan (como en otras técnicas de razonamiento no exacto) para determinar la creencia en la conclusión dado la evidencia disponible sobre la premisa de la regla. Veamos un ejemplo simple. Supongamos que tenemos la siguiente regla:

  SI el conocimiento del alumno sobre el concepto i es *bastante bueno,* y
  el concepto i es prerrequisito para el concepto j, y
  el concepto j no es *demasiado difícil*,
  ENTONCES
  el concepto j *debe ser* el próximo objetivo instructor.

  Diferentes técnicas de inferencia, como por ejemplo la *técnica máx-mín* o la *técnica del producto máximo* pueden ser aplicadas para determinar el resultado, que será un conjunto difuso que se llama *conjunto difuso inducido.*

- Una vez que tenemos el resultado del cálculo difuso, necesitamos convertir el resultado en un resultado *nítido*. Los métodos más usados para este proceso de paso de difuso a nítido son: *el método del máximo*, que selecciona el punto del dominio en el que se alcanza el grado máximo del conjunto difuso y el *método del centroide*, que selecciona el punto del dominio para el cual una perpendicular al eje de abscisas pasaría por el centro del conjunto.

Estos elementos del razonamiento difuso pueden ahora combinarse entre sí: podemos tener los conjuntos difusos como entrada, usarlos en las reglas difusas que sean apropiadas, y después combinar la salida de las diferentes reglas usadas. Finalmente, el conjunto de salidas difusas se convierte en un conjunto de salidas nítidas mediante un proceso de paso de difuso a nítido. Es decir, la configuración básica de un sistema experto basado en lógica difusa es la que se muestra en la siguiente figura:

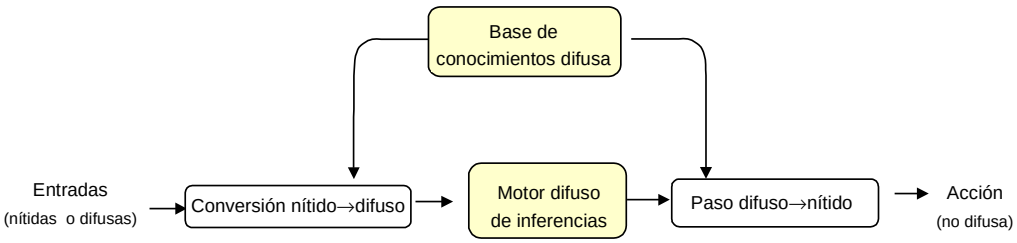

Figura 2.2 Estructura básica de un sistema experto basado en lógica difusa

2.2.3 Redes bayesianas

Una red bayesiana[2] (Pearl, 1988) es un grafo acíclico dirigido en el que los nodos son variables y los arcos representan relaciones de influencia causal entre ellos. Los parámetros usados para representar la incertidumbre son las probabilidades condicionadas de cada nodo dado los diferentes estados de sus padres, es decir, si las variables de la red son $\{X_i, i = 1, \ldots, n\}$ y $pa(X_i)$ representa el conjunto de los padres de X_i para cada i = 1,..., n, entonces los parámetros de la red son $\{P(X_i / pa(X_i), i = 1, \ldots, n\}$. Este conjunto de probabilidades define la distribución de probabilidad conjunta asociada mediante la expresión:

$$P(X_1,...,X_n) = \prod_{i=1}^{n} P(X_i / pa(X_i)) .$$

Por tanto, para definir una red bayesiana tendremos que especificar:

- Un conjunto de variables, $X_1, \ldots, X_n$.
- Un conjunto de enlaces entre esas variables, de forma que la red formada con estas variables y enlaces sea un grafo acíclico dirigido.
- Para cada variable, su probabilidad condicionada al conjunto de sus padres, es decir, $\{P(X_i / pa(X_i)), i = 1, \ldots, n\}$.

Las variables pueden representar el conocimiento del alumno, o el grado alcanzado en la habilidad correspondiente, o si ha sido capaz de resolver determinado

[2] En este capítulo haremos solamente una breve introducción a las redes bayesianas, puesto que al ser el modelo de razonamiento aproximado que hemos elegido dedicaremos un capítulo de la tesis a hacer una descripción más detallada. Una introducción sencilla a las redes bayesianas es (Charniak, 1991). Para una presentación más detallada y actualizada, véase (Castillo, Gutiérrez et al., 1997).

problema. Tomarán valores binarios (*sabido/no_sabido)*, discretos (*mal/bastante mal/regular/bastante bien/bien)* o continuos (el conocimiento del alumno es un número entre 0 y 1), según el nivel de detalle requerido. Una vez que el curriculum y el comportamiento del alumno se han representado mediante variables, utilizamos los enlaces para describir diferentes tipos de influencias: relaciones de prerrequisito, relaciones de agregación, relaciones entre el conocimiento que posee un alumno y las acciones que realiza, etc. Para terminar de definir la red es necesario especificar las probabilidades condicionadas, y a partir de ahí es posible utilizar la red definida para establecer conclusiones a medida que se va obteniendo nueva información o evidencia acerca del alumno. El mecanismo que permite establecer dichas conclusiones se llama propagación de evidencia o simplemente propagación, y consiste en actualizar las distribuciones de probabilidad de las variables según la nueva evidencia disponible.

Las redes bayesianas permiten hacer dos tipos de inferencia distintos:

- *Inferencia abductiva*: Sabiendo que el alumno ha resuelto correctamente una situación, ¿cuál es la probabilidad de que domine cierta parte del curriculum?
- *Inferencia predictiva*: Sabiendo que el alumno domina cierta parte del curriculum, ¿cuál es la probabilidad de que sea capaz de resolver cierto problema P?

2.2.4 La teoría de Dempster-Shafer

La teoría de Dempster-Shafer (Dempster, 1967; Shafer, 1976) se diseñó con objeto de tratar la diferencia entre la incertidumbre y la ignorancia. La teoría de Dempster-Shafer supone que hay un conjunto exhaustivo fijo de elementos mutuamente excluyentes $\Theta = \{\theta_1,\ldots,\theta_n\}$ que se llama el *marco de discernimiento.* Al conjunto de partes de Θ lo denotaremos por 2^{Θ}. Para indicar el grado de creencia, la teoría de Dempster-Shafer utiliza una función *M* que se llama *función básica de asignación de probabilidades*, y asigna a cada elemento de 2^{Θ} un número entre 0 y 1. La función *M* es tal que $M(\varnothing) = 0$ y $\sum_{x \in 2^{\Theta}} M(x) = 1$.

Las tareas de diagnóstico en la teoría de Dempster-Shafer se realizan de forma incremental e iterativa. En este proceso, la evidencia adquirida en una iteración (M_1) se combina con la adquirida en la iteración siguiente (M_2) mediante *la regla de combinación de Dempster*:

$$M_1 \oplus M_2\,(A) = \frac{\sum_{\substack{B,C \subseteq \theta \\ B \cap C - A}} M_1(B) \cdot M_2(C)}{\sum_{\substack{B,C \subseteq \theta \\ B \cap C \neq \varnothing}} M_1(B) \cdot M_2(C)}$$

A partir de la asignación básica de probabilidades, la teoría de Dempster-Shafer define otras tres medidas:

- La *creencia* en un conjunto *A* de 2^{Θ}, que se define como la suma de todas las asignaciones básicas de probabilidad para todos los subconjuntos posibles de *A*, es decir:

$$Bel(A) = \sum_{x \subseteq A} M(x)$$

  Por tanto, la creencia en un conjunto representa la mínima creencia basada en la evidencia disponible.

- La *medida de la duda* de *A*, que se define como $D(A) = Bel(\neg A)$.

- La *credibilidad* (plausibility) de *A*, que se define mediante $Pl(A) = 1 - D(A)$. Esta medida también se llama *función de creencia superior o función de probabilidad superior*, y representa la creencia máxima basada en la evidencia disponible.

El intervalo entre la creencia y la credibilidad se llama *intervalo de creencias* o *intervalo de evidencia*, y longitud tiende a disminuir conforme se va añadiendo nueva evidencia, como se muestra en la Figura 2.3:

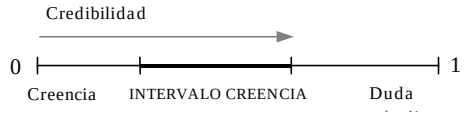

Figura 2.3 Representación gráfica de medidas

Veamos un ejemplo sencillo de cómo esta teoría puede aplicarse al modelado del alumno: supongamos que queremos clasificar a un alumno en una de las siguientes categorías: {1 (*novato*), 2 (*principiante*), 3 (*avanzado*), 4 (*experto*)}. La evidencia disponible para clasificar al alumno son las respuestas que ha dado a un conjunto de preguntas que se le han formulado acerca de ciertos conceptos que previamente se han clasificado como *fáciles*, de *nivel medio* y *difíciles*. Es decir, en este ejemplo, $\theta = \{1,2,3,4\}$ y 2^{θ}. En la Figura 2.4 damos una representación gráfica de 2^{θ}:

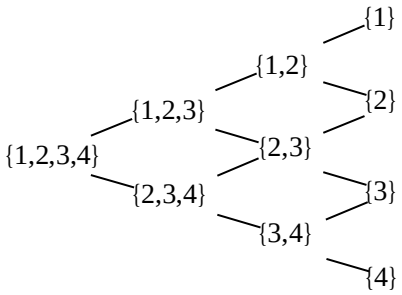

Figura 2.4 Conjuntos de 2^{θ}

El siguiente paso consiste en asignar funciones de creencia a los subconjuntos de θ de acuerdo a las acciones que vaya realizando. Supongamos por ejemplo que el alumno contesta correctamente a una pregunta, demostrando conocer un concepto que previamente hemos clasificado como *fácil*. Como el concepto es fácil, podemos decir que tenemos una creencia del 70% de que el alumno pertenece al conjunto {2,3,4}, en el cual es altamente probable que se posean conocimientos sobre los conceptos de nivel *fácil*. Esta información se representa asignando una función de creencia de 0.7 al conjunto {2,3,4} y una función 0.3 al conjunto {1,2,3,4}.

2.3 Revisión de la aplicación de técnicas de razonamiento aproximado al problema del modelado del alumno

En esta sección revisaremos aplicaciones de los diversos formalismos de razonamiento aproximado al problema del modelado del alumno. Para algunas de ellas no ha sido sencillo encontrar ejemplos de aplicación: para la teoría de Dempster-Shafer hemos tenido que buscar en el campo de modelado del usuario, estrechamente relacionado con el modelado del alumno al que contiene como caso particular. Por esta razón, no todas las secciones son igual de extensas, y, al ser las redes bayesianas el formalismo elegido, la sección dedicada a ellas es la más amplia.

2.3.1 Sistemas basados en reglas y factores de certeza

Hasta finales de los ochenta, los diseñadores de STI sólo disponían de un número limitado de técnicas para tratar con la incertidumbre. Tenían que elegir entre técnicas carentes de fundamentos teóricos sólidos como MYCIN y técnicas generales que en realidad se ajustaban poco al tratamiento de los problemas de este dominio. Muchos investigadores prefirieron desarrollar sus propios heurísticos para resolver este problema, buscando enfoques robustos y fáciles de implementar. Incluso hoy en día, algunos investigadores, más preocupados por otros aspectos de sus STI,

implementan sus propios heurísticos para actualizar su modelo del alumno[3], como por ejemplo ocurre en el tutor web de LISP basado en reglas ELM-ART[4], descrito en (Weber & Spechlt, 1997) y en el tutor de derivación simbólica TUDER (Millán, Vázquez et al, 1996).

Los únicos sistemas que hemos encontrado en la literatura que usan el modelo de factores de certeza son los derivados de MYCIN: NEOMYCIN (Clancey, 1984; Clancey & Letsinger, 1984) y GUIDON (Clancey, 1987). GUIDON es un intento de explorar la posibilidad de transformar sistemas expertos ya existentes en STI. El sistema GUIDON se construyó a partir de MYCIN. Los objetivos perseguidos al desarrollar el sistema GUIDON fueron a) explorar la utilidad pedagógica de la base de conocimientos de un sistema experto, b) determinar qué conocimiento adicional requiere un sistema tutor, y c) expresar estrategias instructoras en términos independientes del dominio. Para ello, la base de conocimientos se mantuvo, añadiéndole nueva información, y se construyó un módulo tutor independiente, con lo cual GUIDON fue uno de los primeros sistemas en los que el conocimiento pedagógico aparecía separado del conocimiento del dominio.

Aunque este enfoque no parece haber sido muy utilizado, queríamos presentarlo aquí por dos razones:

a) Aunque como ya hemos comentado el modelo basado en factores de certeza carece de fundamentos teóricos sólidos, la validación del sistema MYCIN (Yu, Fagan et al., 1984) demostró que funcionaba razonablemente bien. Evidentemente, unos resultados empíricos no son suficientes para validar el modelo de factores de certeza en general, pero al menos demuestran que el enfoque funciona bien para *diagnóstico*, que es una de las componentes clave en el problema del modelado del alumno.

b) El modelo de factores de certeza es muy fácil de utilizar e implementar, de forma que puede ser usado en una primera etapa para evaluar y validar los primeros

---

[3] Como consecuencia, la descripción de estas técnicas no se ha considerado relevante en las publicaciones acerca de estos sistemas, con lo que con frecuencia es difícil saber qué técnicas de razonamiento aproximado se han utilizado.

[4] En (Weber & Spechlt, 1997) no se detalla cómo se hace el tratamiento de la incertidumbre. En comunicación personal mediante correo electrónico, Gerard Weber explicó que utilizaban un heurístico muy simple que habían desarrollado ellos mismos. Para medir el conocimiento del alumno (que organizaban en conceptos) utilizan preguntas tipo test. A cada concepto se le asocia un valor de confianza y cada pregunta un nivel de dificultad (que es un número entre 0.5 y 1.5). Si la respuesta a la pregunta es correcta, el valor de confianza del concepto correspondiente se aumenta en el nivel de dificultad estimado para la pregunta, y si es incorrecta se disminuye. Se considera que el concepto se domina cuando el valor de confianza asociado supera cierto valor fijado inicialmente.

prototipos del sistema, siendo siempre preferible a utilizar técnicas ad-hoc que pueden tener comportamientos imprevisibles al no haber sido debidamente evaluadas.

2.3.2 Sistemas basados en la teoría de Dempster-Shafer

Para encontrar ejemplos de sistemas que han usado la teoría de Dempster-Shafer, hemos tenido que estudiar un campo muy estrechamente relacionado con modelado del alumno: modelado del usuario. Aun cuando el número de sistemas que han utilizado este enfoque es muy pequeño comparado con el número de sistemas que han utilizado redes bayesianas, hemos querido presentar algunos de ellos para después discutir en qué casos creemos que su aplicación debería al menos ser considerada. Los trabajos en los que esta teoría ha sido utilizada son:

2.3.2.1 Inferencias por defecto en identificación de objetivos

El sistema descrito en (Carberry, 1990) utiliza la teoría de Dempster-Shafer para el identificación de objetivos en una herramienta de consultoría que asesora a los alumnos sobre qué carrera elegir. Dada la información acerca de qué asignaturas ha elegido el alumno, el sistema intenta determinar cuál es la carrera que le interesa. Para ello, espera a tener varias observaciones que constituyan evidencia, integrando después estas observaciones usando la regla de combinación de Dempster-Shafer. Los criterios que el sistema usa son: a) el objetivo debe tener una credibilidad que exceda cierto umbral prefijado y b) la diferencia entre esta credibilidad y la siguiente mayor debe ser también superior a cierto umbral. Una vez que se determina el objetivo del alumno, se determina con certeza total, es decir, ningún hecho posterior puede cambiar esta creencia. Carberry utilizó este procedimiento basándose en evidencia psicológica que demuestra que es así como lo hacen los humanos cuando deben hacer inferencia en varias etapas (y no propagando la incertidumbre de una etapa a la siguiente).

2.3.2.2 El sistema PHI

PHI (Bauer, 1995) es un sistema de ayuda inteligente para usuarios del correo electrónico. Usa la teoría de Dempster-Shafer para identificación de objetivos, procesando la evidencia existente sobre los objetivos que pueda tener un usuario de correo electrónico. Se distingue entre dos tipos diferentes de planes: básicos y abstractos. Por ejemplo, una observación puede sugerir que el usuario está intentando almacenar mensajes (plan abstracto) pero no si lo que planea hacer es editarlos o grabarlos (plan básico). Bauer también usa la información sobre el usuario recogida en sesiones anteriores como evidencia para predecir cuáles son sus planes en la sesión actual, enfoque que parece funcionar muy bien cuando el número de sesiones es grande. Para evitar que el sistema se cree expectativas muy definitivas

en las primeras sesiones, Bauer introduce como primera sesión de cada usuario una sesión ficticia en la cual el sistema no pudo hacer ninguna inferencia. Conforme el número de sesiones aumenta, el impacto de esta primera sesión ficticia va perdiendo importancia, imitando así el comportamiento de las redes bayesianas. En versiones más recientes, Bauer introduce una nueva forma de interpretar el comportamiento del usuario en sesiones anteriores: el sistema no sólo graba las acciones del usuario, sino también el contexto en el que fueron realizadas, de forma que el sistema también analiza cómo depende de la situación el plan elegido.

2.3.2.3 Esquemas de inferencias para modelos de errores jerárquicos

Una característica importante de la teoría de Dempster-Shafer es que trata con conjuntos de hipótesis, y por tanto con un conjunto de tamaño mucho mayor que si se trata con las hipótesis individuales. Este hecho puede conducir a problemas de complejidad computacional en casos como el tratado en (Tokuda & Fukuda, 1993): imaginemos que un alumno resuelve problemas de restas, y que se supone que tiene exactamente uno de los 36 errores catalogados en la librería. Si utilizamos la teoría de Dempster-Shafer para intentar determinar cuál de estos errores está cometiendo, tenemos que tratar con 2^{36}-1 subconjuntos no vacíos del conjunto de los 36 errores posibles, lo cual puede resultar demasiado costoso computacionalmente, especialmente si se realizan muchas observaciones. Para reducir esta complejidad computacional, Tokuda y Fokuda dividen los 36 errores en 3 clases básicas, donde cada clase contiene errores que producirían respuestas incorrectas en cada tipo particular de problema. Cuando el usuario da una respuesta incorrecta A a un problema P, se asigna una función de creencia a la clase básica de errores que producen respuestas incorrectas a P (no necesariamente la A) y a cada uno de los subconjuntos de un sólo elemento (una única hipótesis) cuyo error produciría la respuesta A a la pregunta P. Este procedimiento fue utilizado con respuestas incorrectas generadas artificialmente y el sistema era capaz de diagnosticar el tipo de error que las generaba. Pero, aún cuando parece trabajar mucho mejor que la aplicación directa de reglas erróneas en términos de la complejidad computacional, no parece claro si funcionaría bajo circunstancias más reales.

2.3.3 Sistemas basados en lógica difusa

Para revisar las aplicaciones de la lógica difusa al modelado del alumno, hemos adoptado un enfoque bidimensional que nos permitirá clasificar los sistemas de acuerdo con dos puntos de vista diferentes:

- Desde el punto de vista de modelado del alumno, revisaremos las estructuras elegidas, los procedimientos de inicialización y diagnóstico y los diferentes usos que se han dado al modelo (identificación de objetivos, asistencia al alumno, etc.)

- Desde el punto de vista de la lógica difusa, estudiaremos la naturaleza de las entradas del sistema (nítidas o difusas), el proceso de conversión de nítido en difuso (si es que la entrada original era nítida), el tipo de razonamiento que se efectúa en el motor de inferencias difuso (reglas, etc.) y el proceso mediante el que los valores difusos obtenidos como resultado de estos procesos se convierten en nítidos para después tomar decisiones.

Describimos a continuación los sistemas analizados, y concluimos la sección con una comparativa de estos sistemas.

2.3.3.1 El sistema KNOME

KNOME (Chin, 1989) es la componente que realiza el modelado del usuario en el sistema UNIX CONSULTANT (UC), una herramienta de consulta en lenguaje natural para el sistema operativo UNIX. Durante la interacción con el usuario, KNOME crea y mantiene un modelo del usuario que usa para proporcionar ayuda al nivel de detalle adecuado según el conocimiento que posee el usuario. La Figura 2.5 muestra dos ejemplos de sesiones con el sistema UC.

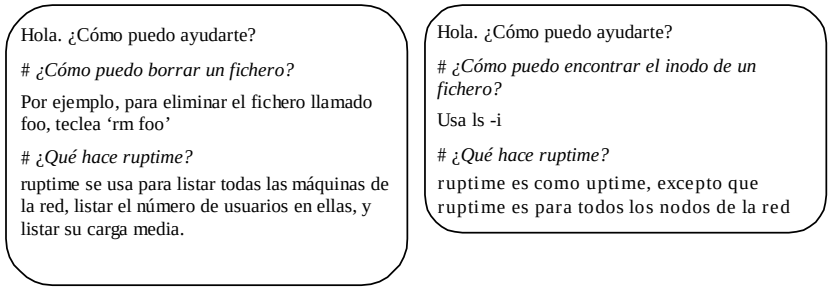

Figura 2.5 Sesiones con el sistema UC del usuario 1 (izquierda) y del usuario 2 (derecha).

En este ejemplo, KNOME ha sido capaz de inferir que el usuario 2 tiene un nivel más avanzado que el usuario 1 (ya que demuestra conocer el concepto de inodo), y por tanto la explicación que le proporciona de ruptime es más concisa que la que da al usuario 1.

Para hacer esto, KNOME representa la verosimilitud y los cambios en ella en términos de una variable difusa con nueve valores discretos: *falso, muy poco probable, ..., muy probable, verdad*. Los usuarios se clasifican en cuatro niveles: *NOVATO, PRINCIPIANTE, INTERMEDIO, EXPERTO* y los conceptos en cuatro niveles de dificultad: *SIMPLE, USUAL, COMPLEJO Y AVANZADO*. KNOME utiliza 16 reglas difusas para predecir el conocimiento del usuario, y 32 reglas para diagnosticar el nivel alcanzado por el usuario. Ejemplos de tales reglas son:

Regla de predicción

SI el usuario *U* es *EXPERTO* y el concepto *C* es *SIMPLE*,
ENTONCES es *muy probable* que U conozca C

Regla de diagnóstico

SI el usuario *U* conoce el concepto *C* y *C* es *COMPLEJO*
ENTONCES parece *más probable* que *U* sea *EXPERTO*

El nivel de un usuario se representa asignando valores difusos a cuatro categorías posibles. Por ejemplo, el nivel al que se inicializa el modelo del usuario para un usuario nuevo es:

NOVATO	*incierto*	*PRINCIPIANTE*	*algo probable*
INTERMEDIO	*incierto*	*EXPERTO*	*incierto*

Estos valores se actualizan según las acciones del usuario. En el ejemplo presentado en la Figura 2.5, una vez que el usuario demuestra conocer el concepto *COMPLEJO inodo*, KNOME elimina la categoría *NOVATO*, disminuye el grado de verosimilitud de PRINCIPIANTE e INTERMEDIO y aumenta el grado de verosimilitud de EXPERTO. Una vez que se ha aceptado una hipótesis, es decir, que la hipótesis alcanza el valor *verdadero*, se asigna el valor *falso* al resto de las hipótesis, y ninguna evidencia futura podrá cambiar esta creencia. El modelo del usuario que se infiere en una sesión no se guarda para sesiones posteriores del mismo usuario, ya que el propósito de este sistema no es la enseñanza, sino proporcionar una ayuda de forma adaptativa.

2.3.3.2 El sistema SPYROS

SPYROS (Herzog & Zierl, 1994) es un STI sobre programación paralela. Como muchos otros sistemas tutores en dominios basados en procedimientos, SPYROS usa un conjunto de objetivos y planes estructurados en forma de árbol para representar el conocimiento del dominio. Este árbol de objetivos y planes se completa con planes y objetivos incorrectos, para que el sistema tenga la capacidad de interpretar la solución del alumno.

El proceso de diagnosticar la solución de un alumno consiste en tres pasos:

- Un algoritmo de reconocimiento encuentra, dada una sentencia en un programa, todos los planes que concuerdan con dicha sentencia $\{P_1, \ldots, P_n\}$.

- Un algoritmo de interpretación selecciona exactamente uno de los planes determinados por el algoritmo de reconocimiento en la etapa anterior.

- Dado el conjunto de planes que determinan todas las sentencias del programa del alumno y el conjunto de objetivos que se corresponden con

esos planes, un *algoritmo de diagnóstico* asigna un grado de corrección a cada uno de los objetivos (*correcto*, *incorrecto*, *innecesario*, etc.) y usando esta información determina el grado de corrección del programa escrito por el alumno.

En este sistema se utilizaron técnicas difusas para realizar la interpretación, es decir, para seleccionar del conjunto de los posibles planes $\{P_1, \ldots, P_n\}$ asociados a cada sentencia un plan P_i. La información que el algoritmo de interpretación usa para seleccionar el plan P_i consiste en diez tipos diferentes de evidencia (procedentes de tres fuentes distintas), de las cuales solamente mencionaremos algunas a modo ilustrativo:

- pasos que los alumnos suelen dar,
- errores más comunes cometidos por los alumnos,
- dificultad relativa de los objetivos G_i asociados con cada plan P_i,
- nivel del alumno,
- explicaciones que se han dado al alumno hasta ese momento,
- hasta qué punto contribuye el plan P_i a alcanzar el objetivo que se sabe que persigue el alumno.

Esta información es procesada por un sistema experto con reglas difusas como la que aparece en el siguiente ejemplo simplificado, tomado de (Jameson, 1996), donde los conceptos y operadores difusos aparecen en letras mayúsculas:

SI P_i tiene el *HISTORIAL ADECUADO*
  Y el objetivo asociado con P_i tiene el *GRADO DE DIFICULTAD ADECUADO*
  O P_i no está asociado con los *ERRORES MÁS COMUNES*

ENTONCES P_i es la *HIPÓTESIS CORRECTA*

Para los operadores O y NO se utiliza la definición más habitual (máximo y resta de 1, respectivamente), y para el operador Y se utiliza la media aritmética en lugar del mínimo, para evitar que una sola condición con un valor muy bajo bloquee la aplicación de la regla.

El proceso de asignar una función de pertenencia a un plan P_i dada una sentencia S se realiza de la siguiente forma: se ordenan los planes de acuerdo al grado en que la sentencia S los apoya; entonces se usa una función no monótona para asignar valores de pertenencia difusos a cada plan: el plan con el grado mayor recibe una pertenencia 1 y el resto una pertenencia que es un valor del intervalo [0, 1).

Los tres sistemas que describimos a continuación se basan en la misma idea.

2.3.3.3 El sistema SHERLOCK II

SHERLOCK II (Katz, 1994) es una simulación realista que permite tutorizar a los alumnos en el diagnóstico de averías en aviación. En este sistema se asocia a cada variable de conocimiento una distribución de probabilidad difusa (dpd) con cinco valores, que van desde *ningún_conocimiento* a *conocimiento_total*. Esta dpd se actualiza, aumentándola o disminuyéndola en diferentes grados según factores como el tipo de evidencia disponible (acciones del alumno, pistas que se le han presentado hasta ahora, etc.). Ejemplos de tales variables son: *habilidad para usar aparatos de medición*, *habilidad para interpretar los resultados de una prueba*, etc. La regla de actualización para la dpd asociada a una variable de conocimiento $F = (f_1, \ldots, f_5)$ se especifica dando dos parámetros: un *vector de escala*, $V = (v_1, \ldots, v_5)$, donde $v_1 = 0$, y un *porcentaje de cambio*, *c*, que controla la medida del cambio. Por ejemplo, la regla de disminución de la dpd es:

$$f_i = f_i - f_i \cdot v_i \cdot c + f_{i+1} \cdot v_{i+1} \cdot c \text{ para } i = 1, \ldots, 4$$
$$f_5 = f_5 - f_5 \cdot v_5 \cdot c$$

Esta regla se basa en la idea de desplazar la distribución hacia la izquierda, de forma que una parte de la probabilidad asignada a cada valor de la variable se transfiere al valor anterior. El vector de escala se utiliza para controlar la velocidad de actualización de los vectores, de forma que por ejemplo se pueda aplicar una actualización más lenta cuando el alumno se considere de nivel muy avanzado (con objeto de no equivocarnos al clasificarlo como experto demasiado pronto). El porcentaje de cambio c se utiliza para controlar la razón de actualización, de forma que indicadores débiles que ocurran muy frecuentemente actualicen la variable lentamente, e indicadores fuertes que ocurran con poca frecuencia actualicen la variable rápidamente. Una expresión muy similar se utiliza como regla de aumento, desplazando en este caso la distribución a la derecha, transfiriendo parte de la probabilidad asignada a cada valor al valor superior. Estos procesos de actualización no se relacionan con ninguna regla de teoría de la probabilidad y por tanto carecen totalmente de fundamento teórico y se basan en ideas intuitivas de cómo debe evolucionar la creencia conforme se adquiere nueva evidencia. El uso del vector de escala y el porcentaje de cambio permite ajustes para conseguir efectos particulares, como la actualización más lenta en los niveles avanzados.

Estas variables de conocimiento se denominan variables locales, y se usan para evaluar habilidades específicas. Las dpds se inicializan con la distribución uniforme (1/5, ..., 1/5), para representar ignorancia sobre el estado de conocimiento del alumno, pero si se tuviese información sobre dicho estado podrían inicializarse con otros valores para representar dicha información. Hay también otro tipo de variables, llamadas variables globales, que representan abstracciones sobre grupos de esas variables locales. Por ejemplo, hay una variable global *X = habilidad para usar los equipos de medición* que se relaciona con las variables locales X_1 = *habilidad para*

usar el osciloscopio, X_2 = *habilidad para usar el multímetro* y X_3 = *habilidad para usar el termómetro* mediante la expresión $X = 0.6X_1 + 0.2X_2 + 0.2X_3$.

Pero este sistema presenta serias anomalías, como demuestra el ejemplo propuesto en (Jameson, 1996): si el alumno demuestra conocimiento total de las variables X_1 y X_2 (es decir, $X_1 = X_2 = (1,0,0,0,0)$) y una ausencia total de conocimiento para las variable X_3 (es decir, $X_3 = (0,0,0,0,1)$), entonces $X = (0.4,0,0,0,0.6)$, es decir, o bien habilidad totalmente desarrollada (con 0.6) o bien carencia total de la habilidad (con 0.4) para utilizar los aparatos de medición, lo cual no parece reflejar correctamente la relación entre las variables. Esta anomalía ilustra el peligro de desarrollar aproximaciones ad-hoc para procesar la evidencia, en lugar de usar métodos teóricos con validez demostrada.

2.3.3.4 El sistema ML-MODELER

ML-MODELER (Gürer, desJardins et al., 1995) es el módulo del alumno de un sistema adaptativo para la enseñanza de Química, que modela dinámicamente el proceso de aprendizaje de un alumno y es capaz de proporcionar tutorización adaptativa. ML-MODELER compara la traza de la solución del alumno con la traza de la solución experta, genera hipótesis sobre los errores del alumno e infiere (utilizando razonamiento basado en casos) los métodos de aprendizaje que el alumno ha utilizado para alcanzar el estado actual de conocimiento. De esta forma, ML-MODELER es capaz de modelar no sólo qué errores y qué áreas conceptuales están siendo problemáticas para el alumno, sino también el posible uso incorrecto de técnicas de aprendizaje como analogía, generalización y especificación.

La estructura usada para representar tanto el conocimiento experto como el conocimiento del alumno es una red conceptual (los autores la llaman MOP) que representa el problema, su solución y los conceptos usados para resolverlo. Cada red conceptual de un alumno representa un episodio de resolución de problemas que consiste en una red conceptual de características, hechos, conceptos y un conjunto de ecuaciones y procedimientos. El modelo del alumno consiste en su estado de conocimiento y sus mecanismos de aprendizaje y se representa también mediante una red conceptual que incluye los conceptos, procedimientos y mecanismos de aprendizaje que ML-MODELER cree que está usando el alumno.

La lógica difusa se utiliza en este sistema para seleccionar los heurísticos y conceptos que mejor explican el comportamiento del alumno. Para describir cada concepto y enlace en la red del alumno, se usan siete valores que van desde *definitivamente_no* a *definitivamente_sí* (en lugar de los cinco valores que se usaban en SHERLOCK II). Estos valores se actualizan con las mismas reglas usadas en SHERLOCK II, pero con un vector de escala $V = (0,1,1,1,1,1,1)$. Como en este caso no

existen variables globales, el sistema no hereda las anomalías que presenta el sistema anterior.

2.3.3.5 El sistema MDF

MFD (Beck, Stern et al., 1997) es un tutor de matemáticas desarrollado para enseñar operaciones básicas con diferentes tipos de números (números enteros, fracciones, números mixtos y decimales). En MDF, cada tipo de problema se considera un tema, y hay relaciones de prerrequisito entre ellos. Cada tema tiene asociada una serie de habilidades, que son pasos en el proceso de resolución del problema. Por ejemplo, el tema *sumar fracciones* tiene asociadas las siguientes habilidades {*encontrar el mínimo común múltiplo, calcular fracciones equivalentes, sumar numeradores y simplificar fracciones*}.

El modelo del alumno de MDF contiene dos tipos de información distintos: un *nivel de conocimiento* para cada tema y *factores generales* relativos a cada alumno, en concreto la capacidad de adquisición de nuevos conocimientos y la capacidad de recordar conocimientos antiguos, a los que llamaremos, respectivamente, factores de *adquisición* y *recuerdo*. La estructura que se usa para representar la incertidumbre en cada tema es un vector de creencias de siete componentes que suman 1. Estos valores indican la posibilidad aproximada de que el alumno haya alcanzado el nivel de conocimiento correspondiente. Así, por ejemplo un vector de creencias de (0.2,0.3,0.5,0,0,0,0) para un tema dado significa que hay una posibilidad de 0.2 de que el alumno tenga nivel 1, 0.3 que tenga nivel 2, 0.5 de que tenga nivel 3 y que estamos seguros de que no tiene un nivel superior a 3 para cada tema. Estos valores se actualizan de acuerdo fórmulas basadas en las utilizadas en (Gürer, desJardins et al., 1995), pero en lugar de desplazar la distribución hacia la derecha o hacia la izquierda el sistema también tiene en cuenta otros factores, como las pistas que se han mostrado al alumno y sus factores de adquisición y recuerdo. Cada pista tiene un índice asociado que describe cuánta información da al alumno. Para índices menores que el índice de la pista de mayor información presentada al alumno hasta el momento, se usa como regla de actualización la regla de aumento con $C = A$, donde A es función del factor de adquisición), y para índices mayores que el índice de la pista, se usa la regla de disminución con $C = B$ (donde B es función del factor de recuerdo), es decir, que las reglas de actualización son:

- Para i menor que el índice de la pista: $f_i = f_i - f_i \cdot A + f_{i-1} \cdot A$.
- Para i mayor que el índice de la pista: $f_i = f_i - f_i \cdot B + f_{i+1} \cdot B$.

El modelo del alumno así construido se usa de varias formas diferentes: para seleccionar el tema objetivo, para generar el problema al nivel adecuado de dificultad y para proporcionar información al alumno de forma adaptativa. Sin embargo, como reconocen los propios autores en (Beck, Stern et al., 1997) se trata de

un trabajo en una etapa temprana y aún hay muchas cuestiones que deben investigarse más a fondo:

- El sistema no usa ningún marco teórico sólido, por lo que no hay una comprensión formal de cómo funciona.

- En lugar de utilizar el vector de creencias se colapsa dicho vector a un único valor mediante una suma ponderada de sus componentes, en la que cada valor del vector se pondera utilizando el nivel correspondiente. Este número se usa como medida del conocimiento del alumno en el tema. Los autores consideran prioritario encontrar un mejor uso para este vector, sin embargo en una publicación posterior sobre el mismo sistema (Beck & Woolf, 1998) no se mencionan avances en este tema.

2.3.3.6 El sistema ALLEN

ALLEN (González, Iida et al., 1994) es un STI sobre Análisis de Circuitos. A los alumnos se les enseña en dos fases diferentes: una primera etapa de adquisición de conocimientos conceptuales, que conlleva el estudio de la teoría y ejemplos en un entorno basado en hipertextos, y una segunda etapa de adquisición de habilidades que mejora las habilidades del alumno mientras que éste resuelve problemas en los que debe aplicar la teoría aprendida. La interacción con el alumno en esta etapa de resolución de problemas es adaptativa, en el sentido de que se puede llevar a cabo bajo tres estrategias instructoras diferentes.

Durante el aprendizaje de conceptos, el sistema usa reglas difusas para inferir el conocimiento del alumno y seleccionar la estrategia instructora apropiada para la sesión de resolución de problemas a partir de los patrones que ha seguido en la navegación a través del hipertexto, utilizando reglas como:

SI el tiempo empleado en estudiar los marcos sobre cierto tema es *bajo*
Y no ha habido *muchos* saltos entre esos marcos
ENTONCES
el nivel de comprensión del alumno es C_i,

donde C_i es un conjunto difuso.

Una vez que todas las posibles reglas en la base de conocimientos difusa se han aplicado a un alumno particular, se le asigna un identificador nítido (*bueno, medio, por debajo de la media*, etc.) según el valor de pertenencia más alto para todos los conjuntos difusos que han sido inferidos como consecuentes de las reglas aplicadas. Este identificador nítido se usa para seleccionar la estrategia instructora más adecuada, que se aplicará en la fase de resolución de problemas. Sin embargo, la estrategia puede cambiarse si el comportamiento del alumno durante la fase de

resolución de problemas sugiere que el alumno necesita un tipo de interacción diferente.

2.3.3.7 Comparativa de los sistemas basados en lógica difusa

Una vez terminada la descripción de los sistemas, empezaremos con la comparativa entre las diferentes aplicaciones de la lógica difusa. Para ello, hemos resumido en la Tabla 2.2 las principales características de cada sistema según el modelo bidimensional definido.

Como puede apreciarse en la Tabla 2.2, KNOME es el único de los sistemas revisados que intenta evitar el uso de representaciones numéricas. Aún cuando este modelo de usuario es muy simple, es capaz de proporcionar la funcionalidad necesaria para los propósitos con los que fue diseñado, y como consecuencia el sistema UNIX CONSULTANT es capaz de proporcionar ayuda adaptativa a sus usuarios. Modelos de usuario simples como este pueden dar mejores resultados en estudios preliminares y en la implementación de prototipos que otros heurísticos ad-hoc carentes de fundamento y no suficientemente comprobados.

		SISTEMAS					
		KNOME	SPYROS	SHERLOCK II	ML-MODELER	MFD	ALLEN
MODELADO ALUMNO	Estructura	Un vector de palabras que describe las vero-similitudes de los niveles de conoc.	Conjunto (árbol) de *objetivos / planes* (habilidad para re-solver problemas).	*Variables de conocimiento* y dpds asociadas *.	*Redes conceptuales:* - Conceptos, procedimientos (representados como en *). - Mecanismos aprendizaje.	- *Variables de conocimiento.* y dpds asociadas. - Factores de *adquisición* y *recuerdo.*	Nivel de *conocimiento*, representado por una variable lingüistica.
	Inicialización	Estereotipos.	--	Uniforme.	Uniforme.	Uniforme.	--
	Diagnóstico	Reglas de diagnóstico actualizan el nivel de conocimiento a partir de las preguntas del usuarios.	Determinación de planes del alumno en su programa y asignación de grado corrección.	Se usan ecuaciones de aumento / disminución de creencia para procesar las acciones del alumno.	- Se diagnostica la solución del alumno (comparando ecuaciones). - Las ecuaciones de aumento / disminución actualizan las pdps.	Se usan ecuaciones de aumento / disminución de creencia para procesar las acciones del alumno.	Reglas difusas determinan nivel de compresión a partir del tiempo empleado en los marcos.
	Usos	- Reglas predicción determinan si el usuario conoce el concepto. - Se presenta ayuda.	Generación adaptativa de explicaciones.	Selección de problemas.	No descrita.	Selección del próximo tema, generación de problemas, asistencia.	Selección de la estrategia instructora.
LÓGICA DIFUSA	Entradas	Preguntas que son interpretadas.	Programas que se descomponen en sentencias.	Acciones tomadas, pistas proporcionadas.	Ecuaciones que representan soluciones a un problema de química.	Números (soluciones a problemas de álgebra básica).	Tiempo empleado y comportamiento en la lectura del marco.
	Paso nítido→difuso	Reglas de diagnóstico asignan al usuario a categorías difusas.	Asignación de fun-ciones de pertenen-cia a los planes.	– –	– –	– –	Conjuntos difusos en *U*=tiempo de estudio de los marcos.
	Motor inferencias	Reglas de predicción determinan lo que sabe el usuario.	Reglas difusas realizan el proceso de interpretación.	Ecuaciones aumento / disminución y variables locales combinadas en variables globales.	Ecuaciones aumento / disminución.	Ecuaciones aumento / disminución.	Reglas difusas.
	Paso difuso→nítido	Se selecciona el nivel de verosimilitud máx.	– –	Dpd máxima.	Dpd máxima.	Se colapsa el vector de Dpd (suma con pesos).	Método del centroide.
	Acciones	Ayuda al nivel de dificultad adecuado.	Explicaciones adaptadas.	Selección de problemas (no descrita).	No descrita.	Selección del siguiente problema.	Selección estrategia tutora.

Tabla 2.2 Clasificación bidimiensional de los sistemas revisados.

Tres de los sistemas analizados se basan en el uso de ecuaciones ad-hoc para actualizar el modelo del alumno. Dichas ecuaciones se basan en la idea intuitiva de

desplazar la distribución a la derecha o a la izquierda según la evidencia disponible. Aún cuando esta técnica parece imitar bastante bien la forma en que razonan los humanos, las inconsistencias del modelo pueden hacer que el comportamiento del modelo del alumno sea impredecible, especialmente en situaciones que no hayan sido previstas por sus autores, como hemos discutido en el ejemplo presentado en el sistema SHERLOCK II.

Solamente dos de los sistemas (SPYROS y ALLEN) usan reglas SI-ENTONCES difusas y las tratan numéricamente con definiciones y técnicas estándar en lógica difusa. Si se usan reglas difusas, es necesario seleccionar los significados de los operadores Y, O y NO de la librería de funciones que proporciona la lógica difusa para la interpretación de dichos operadores. Como se señala en (Jameson, 1996):

"la tarea de determinar las representaciones apropiadas está pendiente, y puede requerir estudios empíricos de envergadura considerable y/o ingeniería del conocimiento".

De acuerdo a estos últimos tres criterios que hemos mencionado, los sistemas revisados pueden clasificarse como se muestra en la Tabla 2.3.

	Representaciones numéricas	Uso de ecuaciones ad-hoc	Uso de reglas difusas
KNOME	No	No	Sí
SPYROS	Sí	No	Sí
ALLEN	Sí	No	Sí
SHERLOCK II	Sí	Sí	No
ML-MODELER	Sí	Sí	No
MFD	Sí	Sí	No

Tabla 2.3 Características de los sistemas analizados.

Como conclusión, creemos que la lógica difusa ha sido usada en el modelado del alumno tan sólo como una alternativa de bajo coste (en términos del esfuerzo de ingeniería del conocimiento requerido). Pero una aplicación más consistente, detallada y cuidadosa de estas técnicas podría producir mejores resultados y, como consecuencia, modelos del alumno más precisos.

2.3.4 Sistemas basados en redes bayesianas

La primera propuesta de usar redes bayesianas en el modelado del alumno aparece en (Villano, 1992). En este artículo se discute la aplicación de dos modelos teóricos distintos al problema del modelado: la *teoría del espacio de conocimiento* y las redes bayesianas. Es aquí donde se pueden encontrar las primeras ideas acerca de cómo construir y usar tales modelos. Desde entonces se han desarrollado varios sistemas en los que las redes bayesianas se han utilizado con éxito para construir y actualizar

el modelo del alumno. Vamos por tanto a describir los principales trabajos y aportaciones que desde entonces se han hecho a este campo.

2.3.4.1 Sistemas OLAE, ANDES y POLA

Los sistemas OLAE (Martin & VanLehn, 1995a; Martin & VanLehn, 1995b), POLA (Conati & VanLehn, 1996a; Conati & VanLehn, 1996b) y ANDES (Conati, Gertner et al., 1997; Conati, Larkin et al., 1997; Gertner, 1998; VanLehn, 1996; VanLehn, Niu et al., 1998) son el resultado de una década (la de los noventa) de investigación del equipo liderado por Kurt Vahn Lehn en la Universidad de Pittsburgh. POLA (1996) es el módulo de diagnóstico del alumno en ANDES (1997) (Sistema Instructor Inteligente para Física Newtoniana), y representa una mejora respecto a OLAE (1995), puesto que permite construir el modelo del alumno con la técnica de *traza del modelo*. Por tanto, describiremos primero el sistema OLAE, y después el sistema POLA.

OLAE (Martin & VanLehn, 1995b) es una herramienta que recopila información sobre alumnos que resuelven problemas a nivel introductorio de física, analiza esos datos con métodos probabilísticos (redes bayesianas) y determina lo que sabe el alumno. OLAE genera automáticamente para cada problema una red bayesiana que relaciona el conocimiento (representado en forma de reglas de primer orden) con acciones concretas, como por ejemplo ecuaciones escritas. Usando la red resultante, OLAE observa el comportamiento del alumno y calcula las probabilidades de que el alumno conozca y use cada una de las reglas.

En la red bayesiana de OLAE, se consideran cuatro tipos de nodos: *nodos de regla*, para recoger si el alumno conoce o no una regla del dominio; *nodos de aplicación de la regla*, para saber si el alumno usó determinada regla durante la resolución del problema propuesto; *nodos de hecho*, que recogen si el alumno sabe determinado hecho acerca del problema y *nodos de acción*, que recogen si el alumno ha realizado determinada acción.

Estos nodos se conectan mediante arcos dirigidos en la red. Los diferentes caminos que se pueden seguir a través de la red representan la multitud de formas que un alumno puede utilizar para resolver determinado problema. Una vez que el alumno da una respuesta, los algoritmos de propagación actualizan las probabilidades a través de los arcos para determinar la probabilidad a posteriori de que el alumno conozca determinada regla.

El grafo de resolución de problemas es una red dirigida de unos 150 nodos, que se va generando de forma automática de la siguiente forma: siempre que se pueda usar una regla para producir una conclusión a partir de ciertos antecedentes, se introduce un nodo en la red para representar la aplicación de la regla. Asimismo se introduce

un arco desde el nodo de aplicación de la regla hasta un nodo de hecho que represente su conclusión (dicho nodo se crea en ese momento si es que no existe). Para cada antecedente (hechos usados para justificar que la regla se dispare) se introduce un arco desde su nodo de hecho hasta el nodo de la aplicación de la regla. También se introduce un arco desde el nodo de la regla hasta el nodo de aplicación de la regla. Si un hecho tiene una acción observable correspondiente, se crea un nodo de acción y se coloca un arco desde el nodo de hecho hasta el nodo de acción. De esta forma OLAE *genera automáticamente* la red bayesiana a partir del modelo del dominio. Una vez la red bayesiana está generada el alumno resuelve el problema y OLAE propaga esta información a través de la red actualizando las probabilidades de cada uno de los nodos.

Otra característica importante de OLAE es que proporciona un segundo tipo de red bayesiana que está diseñada específicamente para el profesor, que consulta el sistema una vez terminado el proceso descrito anteriormente. Esta red para el profesor contiene los siguientes nodos: (a) los nodos de regla de la red bayesiana original que representan el resultado del proceso de inferencias del sistema y (b) nodos dimensionales que almacenan la información de variables más abstractas que representan el dominio que tiene el alumno sobre partes específicas del curriculum, como *Cinemática* o *Dinámica*. En nuestra opinión, estos nodos podrían incluirse directamente en la red, de forma que sus probabilidades se fuesen actualizando a medida que evolucionan las otras probabilidades de la red[5]. Esto permitiría además que, si por cualquier circunstancia adquirimos conocimiento acerca de que el alumno domina determinada parte del curriculum, este conocimiento afectaría también a la probabilidad de que domine las reglas que lo componen.

Cabe resaltar que el sistema OLAE actúa cuando el alumno ha terminado de resolver el problema, puesto que su propósito no era servir de soporte a una enseñanza interactiva, sino simplemente diagnosticar de una forma precisa qué partes del dominio eran conocidas por el alumno.

POLA (Probabilistic On-Line Asessment) (Conati & VanLehn, 1996a) es una extensión del sistema OLAE para determinar no sólo las reglas que sabe el alumno sino el camino seguido por el mismo para la resolución del problema, tratando la incertidumbre en la interpretación de las acciones del alumno de forma consistente utilizando probabilidades. Es decir, mientras que OLAE sólo realiza lo que Anderson y otros (Anderson, Corbett et al., 1995) llaman *traza del conocimiento*

[5] Si la actualización de las probabilidades se hace cuando el alumno termina de resolver el problema, incluir los nodos dimensionales en la red no es más costoso que construir una segunda red bayesiana. Sin embargo, si la actualización se hace cada vez que escribe una ecuación, mantener las dos redes separadas puede resultar en menos cálculos y, por tanto, en más eficiencia.

(determinación de qué sabe el alumno, incluyendo conocimiento correcto y errores), POLA realiza también la traza del modelo (seguimiento de la forma de resolver un problema). En particular, cuando existan varios caminos de resolución que sean consistentes con la acción que ha tomado el alumno, POLA tendrá la capacidad de decidir qué camino es más probable que haya sido el seguido por el alumno. A partir de tal información se dota al sistema de nuevas capacidades, como contestar preguntas formuladas por el alumno o generar pistas a un nivel adecuado, y también se pueden tomar decisiones pedagógicas como proporcionar una ayuda, presentar cierto material o elegir el siguiente problema a proponer.

Con este objeto, es preciso que el módulo de diagnóstico del sistema conozca las posibles líneas de razonamiento que los alumnos pueden seguir. El conjunto de tales líneas se denomina *espacio de soluciones*, y a la estructura de datos usada para representarlo *grafo solución*. El grafo solución se construye automáticamente a partir de una base de conocimientos de reglas de producción y contiene tres tipos de información: a) todos los planes para resolver el problema que se pueden derivar de las reglas de la base de conocimiento; b) todos los caminos algebraicos de resolución que desarrollan dichos planes, y c) el razonamiento que subyace a dichos planes.

Para ilustrar el procedimiento consideremos el siguiente problema: un chico que pesa 75 kilogramos sostiene una bolsa de harina que pesa 40 Newtons. Calcular la fuerza normal que ejerce el suelo sobre el chico. La Figura 2.6 muestra el grafo solución para este problema.

Los nodos de aplicación son nodos de tipo AND (ya que para que una regla se aplique es necesario que la regla y todos sus antecedentes sean conocidos) y los nodos de hecho son nodos de tipo OR (modelando el hecho de que a ellos se puede llegar por varios caminos diferentes). Así, el sistema genera un grafo AND/OR que codifica todas las formas conceptualmente distintas en las que se pueden combinar las reglas y los datos dados para llegar a la solución final.

Para determinar cuáles de los posibles caminos solución ha escogido el alumno, es necesario distinguir entre las reglas que el alumno ha utilizado ya y las que pertenecen a su camino solución pero aún no han sido utilizadas. Para ello, Conati y VanLehn adoptan la estrategia de ir construyendo la red bayesiana de una forma incremental conforme el alumno va resolviendo el problema, de forma que las reglas que aún no han sido usadas no forman parte de la red bayesiana que se utiliza para la inferencia.

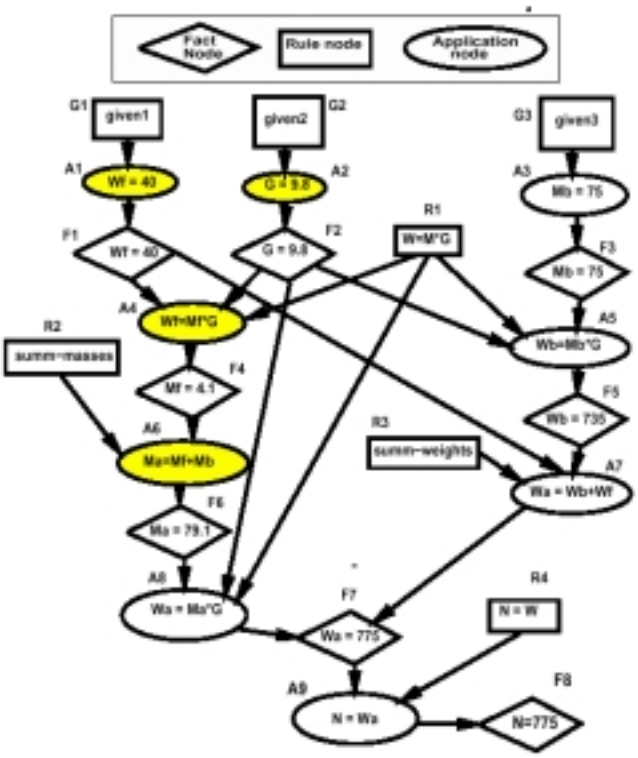

Figura 2.6 Gráfico AND/OR de solución (tomado de (Conati & VanLehn, 1996b)).

Uno de los artículos relativos a ANDES (VanLehn, Niu et al., 1998) merece especial mención por su relación con nuestro trabajo. El objetivo de esta investigación era determinar las probabilidades a priori que tiene un alumno de conocer o no cada una de las 350 reglas (items elementales de conocimiento) en las que se ha dividido el dominio en el sistema ANDES. Para ello, los profesores de Física asociados al proyecto desarrollaron un examen de 34 preguntas (con respuestas cortas o tipo test multirespuesta) que se evaluaban como correctas o incorrectas y que utilizaban 66 de las 350 reglas. El problema era entonces encontrar un algoritmo de diagnóstico, es decir, un algoritmo que dadas las respuestas de un alumno a las preguntas y las relaciones entre preguntas y reglas, determinase el subconjunto de reglas que eran conocidas por el alumno que ha hecho el examen.

Para evaluar dicho algoritmo VanLehn usa alumnos simulados, en los que modela también los aciertos casuales sin poseer conocimiento (adivinanzas, en inglés *guesses*) y los errores no intencionados (descuidos, en inglés *slips*), utilizando las siguientes expresiones:

- P(respuesta correcta / domina todas las reglas) = 1 – P(descuido)
- P(respuesta correcta / al menos una de las reglas no es conocida) = P(adivinanza) / número de posibles respuestas.

En nuestra opinión la segunda regla para asignación de probabilidades puede mejorarse, porque, especialmente en preguntas tipo test, contra más conocimiento posea el alumno más fácil es que dé la respuesta correcta (aunque sea descartando las alternativas incorrectas), y por tanto creemos que no se debe dar la misma probabilidad de responder correctamente si al alumno no conoce una de las reglas que si no conoce ninguna de ellas. Este tema se discutirá más ampliamente en el capítulo 5.

Las medidas que utilizan para evaluar la bondad del algoritmo de diagnóstico son: la *precisión*, que definen como la proporción entre el número de reglas que el sistema ha diagnosticado correctamente como dominadas por el alumno simulado y el número de reglas que fueron diagnosticadas como dominadas, y la *cobertura*, que definen como la proporción entre el número de reglas que el sistema ha diagnosticado correctamente como dominadas por el alumno simulado y el número de reglas que el alumno domina. Por tanto, ambos parámetros deben tomar (idealmente) valores próximos a 1.

En este trabajo se probaron varios esquemas alternativos para modelar con redes bayesianas las relaciones entre preguntas y reglas. Básicamente, estas alternativas se pueden reducir a dos: a) conocer una regla tiene influencia causal en contestar correctamente un problema (relación $R \rightarrow P$), y b) responder correctamente a un problema es informativo para saber que el alumno domina una regla (relación $P \rightarrow R$). La segunda opción tuvo que ser descartada porque las relaciones de independencia que implica no se corresponden con las relaciones de independencia que se dan en la vida real.

La conclusión fue por tanto que el modelo que más se ajustaba era el a), el único problema era que las probabilidades a posteriori (una vez evaluadas las respuestas) parecían depender fuertemente de las probabilidades a priori que se especificaran. La solución al problema fue basar el diagnóstico en la *medida del cambio* en la probabilidad (es decir, la diferencia entre la probabilidad a posteriori y la probabilidad a priori) en lugar de los valores de las probabilidades a posteriori.

2.3.4.2 El sistema HYDRIVE

El sistema HYDRIVE (Mislevy & Gitomer, 1996) modela la habilidad que tiene un alumno para diagnosticar averías en el sistema hidráulico de los aviones F-15. El problema empieza con un vídeo en el que un piloto, que está a punto de aterrizar o de despegar, describe algún problema en el funcionamiento. La interfaz de HYDRIVE permite que el alumno intente diagnosticar la avería por los procedimientos usuales y le permite consultar tanto videos de las componentes como material de apoyo técnico. El comportamiento del alumno es observado por el sistema con el propósito de evaluar cómo el alumno hace uso de la información

disponible para dirigir las acciones que le permiten diagnosticar la avería. El sistema de diagnóstico de HYDRIVE evalúa la calidad de acciones de diagnóstico de averías concretas y caracteriza el conocimiento del alumno mediante el uso de variables más generales como conocimiento del sistema y estrategias y procedimientos usados. Así, el conocimiento del alumno se divide en tres partes: conocimiento del sistema, conocimiento de estrategias y conocimiento de procedimientos de resolución. Cada una de estos conocimientos se divide a su vez en otros nodos y variables. De esta forma, el diagnóstico producido por el sistema es lo bastante preciso para saber qué partes del conocimiento no tiene el alumno, pero también lo bastante general como para guiar las estrategias instructoras básicas, como por ejemplo si dada su respuesta procede presentarle una ayuda o proponerle una situación más complicada. En la Figura 2.7 mostramos una parte de la red bayesiana utilizada en HYDRIVE[6] como ejemplo de los tipos de nodos que se definen y de la forma de modelar las relaciones entre ellos.

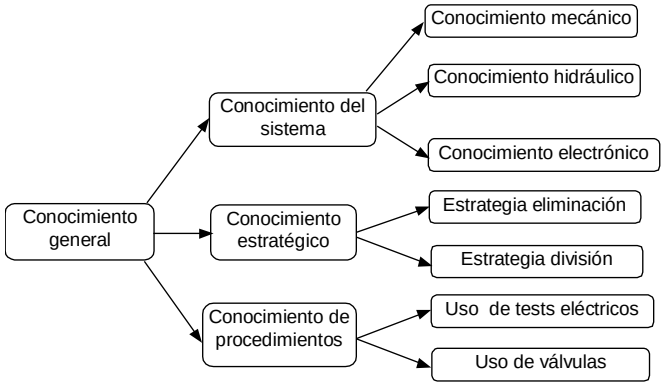

Figura 2.7 Red bayesiana de HYDRIVE.

El sistema de inferencias en HYDRIVE es mixto. Así, existe lo que los autores llaman un *intérprete de estrategias*, que emplea un número relativamente pequeño de reglas (unas 25) para caracterizar la estrategia de resolución que está utilizando el alumno.

2.3.4.3 Modelado del alumno con redes bayesianas dinámicas

Los trabajos de Jim Reye (Reye, 1996, Reye, 1998) en modelado bayesiano del alumno se basan en la hipótesis de que el dominio de conocimiento se puede

[6] Los parámetros del sistema fueron inicialmente especificados por expertos en la materia, y posteriormente modificados utilizando técnicas de simulación.

estructurar en una colección abstracta de temas que representan conocimiento conceptual o habilidades que el alumno debe adquirir, y que esos temas admiten una estructuración en forma de relaciones de prerequisitos. Reye propone que la estructura de la red bayesiana apropiada para este problema se base en: a) una parte central, que conecte a todos los nodos de conocimiento (que Reye denomina nodos "*student-knows*") ordenándolos en términos de una lista de relaciones de prerrequisito, y b) un grupo de nodos para cada uno de los nodos conocimiento de la parte central, en el que aparezcan el nodo de conocimiento y un conjunto de nodos adicionales relacionados con él, como por ejemplo nodos para medir el interés del alumno en el tema particular (lo que Reye denomina nodos "*student-interested-in*"). Este tipo de estructura tiene la ventaja de que las actualizaciones sucesivas de la red conforme se va adquiriendo nueva evidencia se llevan a cabo de forma local y sólo afectan a las partes de la red correspondientes a otros temas a través de la parte central, permitiendo aumentar la eficiencia al realizarse los cálculos localmente. Pero en nuestra opinión esta estructura resulta demasiado restrictiva, puesto que como hemos visto en otros sistemas discutidos en esta sección, los nodos en la red pueden utilizarse para representar muchos factores diferentes que no tienen cabida en este tipo de enfoque.

Otro trabajo sobre la utilización de redes bayesianas dinámicas para modelado del alumno es (Reye, 1998). La red bayesiana dinámica que utiliza Reye en este trabajo es muy simple, puesto que el concepto de *dinámico en el tiempo* se mide en función de interacciones con el sistema en lugar de en función de intervalos de tiempo. De esta forma, para cada $i = 1, ..., n$ se define el nodo L_i = "estado del conocimiento del tema que posee el alumno después de la *i*-ésima interacción con el sistema", y este nodo se hace depender del nodo L_{i-1} y del nodo O_{i-1} (resultado de la interacción *n*-ésima, que a su vez dependerá también de L_{i-1}). De este modo, las redes bayesianas tienen la siguiente estructura:

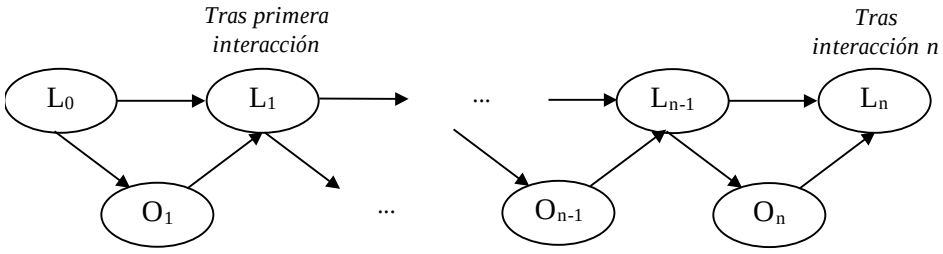

Figura 2.8 Redes bayesianas dinámicas para modelado del alumno.

Con este esquema, la probabilidad a posteriori $P(L_n / O_n)$ resulta ser función de $P(L_{n-1})$ y de tres parámetros del sistema:

$\rho = P(L_n / \neg L_{n-1}, O_n)$,
$\lambda = P(L_n / \neg L_{n-1}, O_n)$, y
$\gamma = P(O_n / L_{n-1}) \ / \ P(O_n / \neg L_{n-1})$.

Al analizar la forma de tales funciones según los valores de ρ, λ, y γ se observa que la forma de las curvas explica perfectamente las funciones utilizadas por Shute para actualización del modelo del alumno en el sistema SMART (Shute, 1995a; Shute, 1995b) y también las fórmulas utilizadas por Corbert y Anderson para calcular la probabilidad de que una regla sea conocida por el alumno dada su respuesta correcta o incorrecta a la pregunta planteada en el sistema ACT (Anderson, Corbett et al., 1995; Corbett & Anderson, 1992).

2.3.4.4 Test adaptativos y redes bayesianas

El trabajo del grupo ARIES (James Greer, Gordon Mc Calla, Sherman Huang, Jason Collins y otros) es quizás el más directamente relacionado con el nuestro, puesto que investiga el uso de redes bayesianas en test adaptativos (Collins, Greer et al., 1996). Se basa en la aplicación de redes bayesianas y jerarquías de granularidad (McCalla & Greer, 1994) para, a partir de un conjunto de preguntas tipo test, evaluar al alumno. En este trabajo se parte de un dominio de conocimiento estructurado en: *objetivos a aprender* (learning objectives) con niveles de logro específicos y un conjunto de *preguntas (*que no son necesariamente tipo test, sino que pueden ser cualquier tipo de preguntas siempre que aseguremos que podemos comprobar si la respuesta que da el alumno es correcta o incorrecta). Los tipos de relaciones considerados son: *relaciones de agregación* (que permiten descomponer un objetivo en subobjetivos y que garantizarán tests de contenido equilibrado), *relaciones de prerequisitos* (que permiten una estructuración del dominio y que ayudan a establecer el orden de las preguntas en el test) y *relaciones objetivos-pregunta*, entre objetivos de aprendizaje alcanzados y preguntas, que son las que permitirán realizar el diagnóstico.

En cuanto a la selección de preguntas, la propuesta de este grupo es elegir la pregunta *más informativa* (que maximiza cierta medida de utilidad). La medida de utilidad de una pregunta *Q* para un objetivo *O* la definen como:

$$\text{utilidad}(Q) = \left| P(O / Q) - P(\neg O / \neg Q) \right|$$

Es decir, la probabilidad de que se domine el objetivo *O* dado que la pregunta se responde correctamente menos la probabilidad de no dominarlo dado que la pregunta se responde incorrectamente. Para calcular dichos valores, cada vez que queremos elegir una pregunta deberemos actualizar la red 2*n* veces (donde *n* es el número total de preguntas), construir las diferencias y elegir la máxima. Para nosotros, esta medida de utilidad es muy discutible, puesto que el objetivo debería ser maximizar ambas probabilidades y por tanto no tiene mucho sentido maximizar

la diferencia en valor absoluto. En el capítulo 6 propondremos medidas de utilidad alternativas.

Como criterio de finalización del test proponen dos alternativas: a) que el nivel de conocimiento del objetivo instructor más grueso caiga por encima o por debajo de ciertos valores o b) en el caso de que se necesite una evaluación más completa, utilizar los niveles de conocimiento de cada uno de los subobjetivos instructores, es decir, de cada uno de los hijos del nodo objetivo instructor.

Sin embargo, no parecen tener claro cuál es la estructura más adecuada de la red bayesiana, cuál es el efecto que tiene añadir o no nodos dimensionales ni cuál es la dirección adecuada de las relaciones de agregación (todas estas cuestiones se discutirán en más detalle en el capítulo 5). Por ello, realizan simulaciones con tres estructuras diferentes que se representan en la Figura 2.9: la estructura (A), en la que aparecen nodos dimensionales y las relaciones de agregación son de *parte-a-todo*, la estructura (B), que es la (A) sin nodos dimensionales, y la estructura (C), que es la (B) en la que cambian la dirección de los arcos (ahora son de *todo-a-parte*, en un intento de reducir las probabilidades requeridas).

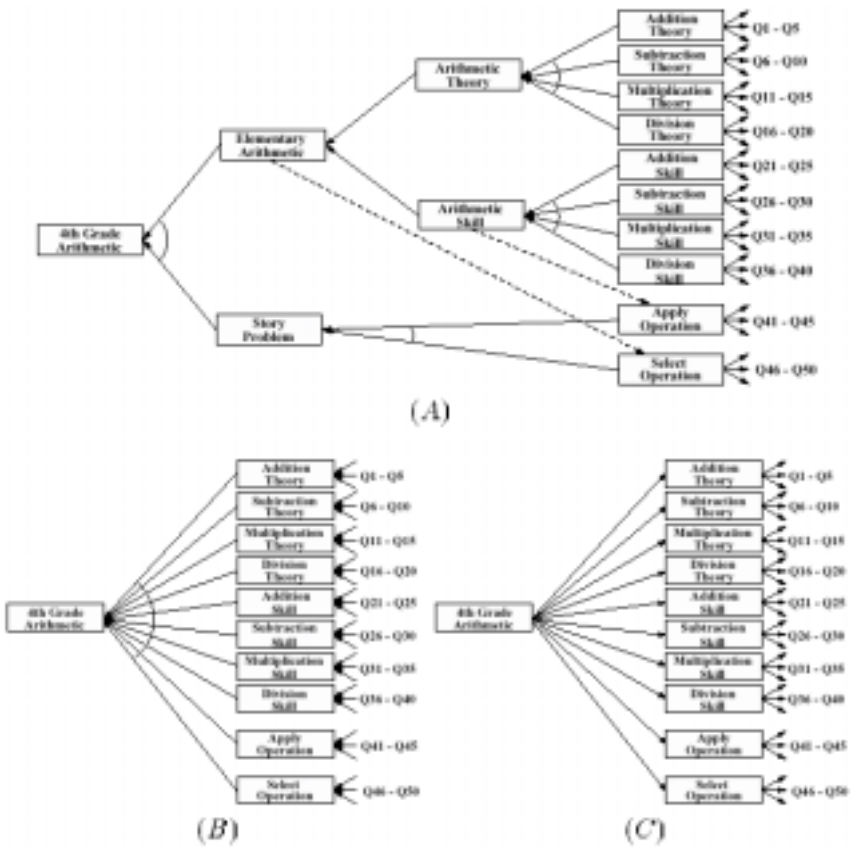

Figura 2.9 Diferentes estructuras de la red. Tomada de (Collins, Greer et al., 1996).

Las pruebas se realizaron con alumnos simulados. Nótese sin embargo que el funcionamiento del test adaptativo tan sólo fue evaluado con alumnos cuya probabilidad de responder correctamente era 0, 0.2, 0.8 y 1, es decir, no se consideraron alumnos de nivel intermedio que probablemente son los más difíciles de diagnosticar puesto que su comportamiento es más impredecible. Los resultados aparecen en la Figura 2.10, donde las *probabilidades condicionadas requeridas* son las probabilidades que tendría que dar el experto; la *probabilidad de respuesta correcta* indica las probabilidades dadas para el alumno simulado; la *longitud del test* indica el número de preguntas que es necesario hacer para que se complete el test y la *cobertura* es la proporción objetivos preguntados en el test / total de objetivos. A la vista de los datos de la Figura 2.10, los autores concluyen que: a) la jerarquía A parece ser mejor tanto en términos de longitud del test como de cobertura de contenidos; b) la jerarquía B es casi igual de buena en términos de cobertura, pero mucho más complicada de generar puesto que requiere más probabilidades condicionadas y además da tests más largos y c) la jerarquía C es más fácil de especificar que A y la duración de sus tests es prácticamente la misma, pero no nos garantiza la cobertura.

Hierarchy Tested	Conditional Probabilities Required	Probability of Correct Answer	Test Length	Coverage
A	155	1.0	20	1.00
		0.8	21	1.00
		0.2	44	1.00
		0.0	14	0.83
		Average	**24.8**	**0.96**
B	1345	1.0	30	0.90
		0.8	36	0.90
		0.2	39	0.90
		0.0	31	0.90
		Average	**34.0**	**0.90**
C	122	1.0	41	1.00
		0.8	50	1.00
		0.2	12	0.70
		0.0	4	0.40
		Average	**26.8**	**0.78**

Figura 2.10 Resultados de la simulación. Tomada de (Collins, Greer et al., 1996).

2.3.4.5 El sistema Desktop Associate

El sistema Desktop Associate (Murray, 1998; Murray, 1999) evalúa las habilidades de un usuario que usa un procesador de textos. Los nodos que utiliza Murray son básicamente de dos tipos:

- *Nodos de habilidades*, que miden si el alumno es capaz de hacer algo, como dar formato a un párrafo, cambiar el tipo de letra, etc. Dentro de estos nodos, se distinguen las *habilidades básicas* (habilidades que no admiten descomposición),

como cambiar el tipo de letra, y las habilidades más generales (*nodos dimensionales*), como dar formato a un párrafo. Ambos tipos de nodos aparecen simultáneamente en la misma red, a diferencia del enfoque adoptado por Conati y Van Lehn en POLA.

- *Nodos evidencia*, que son los nodos encargados de recoger la información sobre el alumno que después servirá para determinar su nivel de conocimiento. Dicha información se puede recoger de tres formas distintas (que se corresponden con tres clases distintas de nodos evidencia): realizando preguntas al alumno, pidiéndole que realice cierta tarea o, si el profesor observa de forma directa que el alumno tiene cierta habilidad, introduciendo esta información en la red.

En cuanto a las relaciones de causalidad, Murray considera que tener una habilidad tiene influencia causal en ser capaz de realizar una tarea o contestar una pregunta, y que poseer una habilidad general tiene influencia en poseer las habilidades en las que se descompone.

En (Murray, 1998) se propone una simplificación para el problema de la obtención de los parámetros de la red. Partiendo de una red en la que sólo hay nodos de habilidad (que pueden tomar n valores) y nodos pregunta (que son binarios, es decir, se considera que cada pregunta se responde correcta o incorrectamente). En general, si para medir una habilidad tenemos q preguntas, se deben especificar $n \cdot q \cdot k$ probabilidades condicionadas, y n-1 probabilidades a priori. Si quisiéramos modelar k habilidades, necesitaríamos $k \cdot n \cdot q$, que es un número muy grande incluso para valores pequeños de n, q y k. Para reducir el número de datos precisos, Murray propone a) agrupar las preguntas por niveles de dificultad, y utilizar los mismos parámetros para preguntas del mismo nivel, lo cual reduce el número de parámetros necesarios de $k \cdot n \cdot q$ a $k \cdot n \cdot c$, donde c es el número de niveles de dificultad considerados y b) asociar estos niveles de dificultad a los valores de los niveles utilizados para las habilidades, es decir, si por ejemplo para cada habilidad se tienen cinco valores {*novel, principiante, intermedio, avanzado, experto*}, podemos considerar cuatro categorías de preguntas {*nivel-principiante, nivel-intermedio, nivel-avanzado, nivel-experto*} (no necesitamos nivel novel puesto que el alumno se clasificará como novel cuando no pueda contestar bien ni siquiera a las preguntas de nivel principiante). La última reducción en el número de parámetros es resultado de la naturaleza transitiva de esta clasificación de las habilidades en categorías: si un alumno alcanza cierto nivel, entonces debe ser capaz de responder a todas las cuestiones correspondientes a este nivel y a niveles inferiores, y probablemente no responderá correctamente a las preguntas de niveles más avanzados que el suyo. Para modelar las *adivinanzas* (respuestas correctas sin tener conocimiento) y los *descuidos* (errores debidos no a una falta de conocimiento, sino a otros factores difícilmente controlables como despistes, errores al teclear, etc.), que pueden modificar las hipótesis anteriores, se usan dos probabilidades: s (probabilidad de

error) y g (probabilidad de adivinanza), y las probabilidades condicionadas se construyen como en el siguiente ejemplo:

- P(respuesta correcta a pregunta nivel intermedio / alumno principiante) = *g*.
- P(respuesta incorrecta a pregunta nivel principiante / alumno principiante) = 1-*s*.

De esta forma, las $k \cdot n \cdot c$ probabilidades condicionadas se pueden obtener de *s* y *g*. Esta forma de calcular el número de parámetros tiene otra ventaja adicional: el proceso de propagación de probabilidades cuando se adquiere evidencia tiene lugar en tiempo lineal. Como contrapartida, la principal desventaja de esta aproximación es su limitado alcance: sólo permite diagnosticar una habilidad cada vez, y sólo permite usar nodos evidencia binarios. Y por último, la gran limitación de este enfoque es que su validez se restringe a redes con forma de árbol, es decir, en redes cuyos nodos tienen un único padre, lo cual es, una restricción muy fuerte. Basándonos en esta idea, hemos realizado unas extensiones de las puertas AND y OR clásicas (Pearl, 1988) que permiten simplificar el problema de la especificación de los parámetros en redes con cualquier tipo de estructura. Dichos resultados aparecen publicados en (Millán, Agosta et al., 2000).

Como continuación de este trabajo Murray propone en (Murray, 1999) una implementación del algoritmo clásico de propagación en árboles (Pearl, 1988) que garantiza la actualización en tiempo lineal.

2.3.4.6 Otros trabajos

Hay otros trabajos (Greer, Zapata-Rivera et al., 1999; Henze & Nedjl, 1999; Madigan & Almond, 1995; Madigan, Hunt et al., 1995; Madigan, Raftery et al., 1995; Mislevy & Almond, 1997; Mislevy, Almond et al., 1998; Mislevy, Steinberg et al., 1999; Möbus & Schröder, 1997; Petrushin & Sinitsa, 1993; Schäfer & Weyrath, 1997; Sime, 1993) que tratan sobre el desarrollo de modelos del alumno usando redes bayesianas, pero nosotros hemos descrito en detalle sólo aquellos más directamente relacionados con el nuestro. Asimismo, las redes bayesianas también se han usado en modelado del usuario (Horvitz, Breese et al., 1998; Wolverton, 1999). Quizás el más conocido de estos trabajos sea el asistente de Microsoft Office, desarrollado por el grupo de Eric Horvitz y David Heckerman en Microsoft (para una descripción detallada, véase (Horvitz, Breese et al., 1998) o http://research.microsoft.com/~horvitz/lum.htm), en el que se usan redes bayesianas para emular el comportamiento de expertos humanos en la tarea de intentar dilucidar qué problemas está teniendo el usuario con el software a partir de su comportamiento con objeto de poder proporcionarle la ayuda adecuada. Una excelente revisión de aplicación de técnicas de inteligencia artificial (teoría de Dempster-Shafer, lógica difusa, redes bayesianas) al modelado del usuario y al modelado del alumno es (Jameson, 1996).

2.4 Conclusiones

En este apartado justificaremos los motivos por los que hemos decidido utilizar las redes bayesianas en lugar de las otras alternativas existentes. En principio, las redes bayesianas parecen ser útiles en toda clase de situaciones, tienen gran versatilidad en modelado del alumno y constituyen una herramienta muy potente para realizar inferencias abductivas y predictivas. Sin embargo, su uso en modelado del alumno no está todo lo extendido que cabría esperar, puesto que a cambio de su solidez teórica y su potencia tienen otras desventajas, principalmente a) el esfuerzo que supone especificar el modelo (variables y relaciones causales) y estimar los parámetros (probabilidades condicionadas), b) la complejidad computacional de los algoritmos de propagación, y c)la dificultad que supone la implementación de los mismos. Con nuestro trabajo pretendemos paliar estos inconvenientes, con objeto de dotar al modelo del alumno de la solidez teórica de la que hasta ahora carecen muchos de los sistemas existentes, reduciendo la complejidad computacional y disminuyendo el número de parámetros que es necesario especificar. Vamos por tanto a analizar cómo se pueden solventar estas dificultades:

- *Especificación de los parámetros.* Este problema parece en principio el más difícil de solventar. Al utilizar las redes bayesianas se supone que tanto la estructura de dependencias como los parámetros son proporcionados por el experto humano; sin embargo, para un profesor puede resultar imposible especificar el gran número de probabilidades condicionadas que se requieren. Todo ello ha motivado que se haya investigado mucho en técnicas de simplificación de los parámetros, o de obtención de los mismos a partir de bases de datos existentes (también para aprender las estructuras, es decir, las relaciones causales, a partir de datos). Por ejemplo, en (Druzdel, 1995) se describe un método general para derivar las probabilidades a partir de conjuntos de datos (de diferentes formas). Para una introducción sencilla a los métodos de aprendizaje (tanto de estructuras como de parámetros) en redes bayesianas, se puede consultar el libro de Castillo (Castillo 1997). Estas técnicas se han aplicado ya con mucho éxito en dominios como por ejemplo diagnóstico médico, en los que existen grandes bases de datos procedentes de hospitales y otras fuentes. Sin embargo, la aplicación de estas técnicas está condicionada a la existencia de estas bases de datos, que probablemente sean escasas en modelado del alumno. Otras posibilidades para paliar el esfuerzo de adquisición del conocimiento son el uso de modificaciones de las puertas AND y OR, como las propuestas en (Millán, Agosta et al., 2000) y de simplificaciones como las que se proponen y discuten en el capítulo 5.

  Por todo ello en principio puede parecer más sencillo utilizar otros métodos, como la lógica difusa o la teoría de Dempster-Shafer, puesto que normalmente resulta más fácil hacer que el experto describa sus opiniones en términos de reglas difusas o creencias que pedirle que las cuantifique en forma de

probabilidades. Pero utilizar estos modelos no nos libra de utilizar *números*: una variable difusa tiene asociada una función de pertenencia, y para combinar la información obtenida con las reglas difusas se necesita utilizar operadores. También será preciso trabajar con números si utilizamos la teoría de Dempster-Shafer, puesto que es necesario asignar funciones de creencia a las diferentes hipótesis. En cualquier caso, la exactitud de estos valores es tan incierta y cuestionable como la de las probabilidades en las redes bayesianas.

- En cuanto al esfuerzo de *implementación de los algoritmos*, hoy en día existe software comercial y de dominio público que facilita grandemente la construcción y actualización de las redes, como por ejemplo HUGIN (http://hugin.dk) y NETICA (http://www.norsys.com/netica.html). También existen librerías desarrolladas en diferentes lenguajes de programación que pueden ser integradas en los sistemas tutores desarrollados, como por ejemplo las librerías HUGIN, SMILE (http://www2.sis.pitt.edu/~genie/), y JAVABAYES (http://www.cs.cmu.edu/~javabayes/index.html/). Para una revisión de los programas y librerías existentes y de sus características, véase http://bayes.stat.washington.edu/almond/belief.html.

- En cuanto a la *complejidad computacional* de los algoritmos, antes de descartar el uso de redes bayesianas en un sistema deberían considerarse las siguientes cuestiones:
  - *Tamaño y estructura* de la red que tendremos que utilizar para representar nuestro dominio. Si las redes bayesianas son lo suficientemente pequeñas o poseen una estructura especial (árbol, redes simplemente conexas, o con número de padres de cada nodo pequeño) entonces la complejidad computacional no supondría ningún problema.
  - *Posibilidad de usar enfoques mixtos* si es que hay partes específicas del sistema que se prestan especialmente a ser modelados utilizando las redes bayesianas de tamaño aceptable.
  - *Uso de técnicas especiales para reducir la complejidad.* Por ejemplo, los algoritmos orientados a un objetivo (Castillo, 1997) determinan la parte de la red que debe utilizarse según los nodos que nos interesen en cada momento y la propagación se efectúa en este conjunto reducido, disminuyendo así los tiempos de computación
  - Por último, siempre que aparezcan problemas de complejidad debe considerarse el *uso de algoritmos de propagación aproximados*, menos costosos que los exactos.

Una vez realizado el esfuerzo de especificación e implementación de las redes bayesianas podremos disfrutar de sus ventajas:

- *Consistencia*. Si el sistema se comporta de una forma incorrecta o inesperada sabremos que este mal funcionamiento no es debido al mecanismo de inferencias utilizado, y por tanto deberemos revisar las hipótesis del modelo. Puede que los resultados obtenidos por el sistema sean inexactos, pero no serán nunca inconsistentes.

- *Explicaciones*. Si el diseñador del sistema tiene que explicar el papel de las inferencias, el objetivo esencial sería explicar la naturaleza de las relaciones causales de las variables representadas en la red.

- *Comunicación*. Al utilizar una técnica de razonamiento aproximado ampliamente difundida, será mucho más sencillo que otros colegas entiendan y sean capaces de evaluar nuestro sistema.

Por último daremos las razones que nos llevaron a descartar las otras alternativas utilizadas:

- Como ya hemos discutido en la sección 2.3.1, no recomendaríamos el uso de factores de certeza en modelado del alumno, sobre todo por su falta de una base teórica sólida. Cuando se usan modelos carentes de fundamentos teóricos las inconsistencias pueden hacer que el comportamiento del modelo del alumno sea impredecible, especialmente en situaciones que no han sido consideradas previamente por sus autores. Sin embargo, consideramos que el modelo de factores de certeza es un procedimiento sencillo de entender e implementar en las primeras versiones de un sistema, permitiendo así hacer una primera evaluación antes de utilizar modelos mejores desde el punto de vista teórico pero que exigen un esfuerzo mucho mayor de implementación como las redes bayesianas.

- La lógica difusa ha sido considerada seriamente como alternativa a las redes bayesianas por su capacidad para procesar datos de entrada expresados verbalmente de forma imprecisa, y no descartamos su uso en nuestro trabajo futuro. La lógica difusa debería ser considerada en aquellas situaciones en que:

  - El razonamiento que hay que realizar se pueda describir de forma natural en términos de conceptos, operadores o reglas imprecisas. Este razonamiento puede ser el relativo al alumno cuyo comportamiento estamos intentando anticipar, o al tutor humano cuyo conocimiento estamos intentando transferir al sistema tutor.
  - Necesitamos procesar datos de entrada imprecisos, como por ejemplo en el caso de un tutor que deba procesar afirmaciones en lenguaje natural.

  Hay que tener en cuenta que si utilizamos lógica difusa nos veremos obligados a elegir entre diferentes interpretaciones para algunos de sus conceptos, como

por ejemplo entre diferentes procesos de paso de difuso a nítido o diferentes significados para los operadores AND, OR y NOR.

- Para la aplicación de la teoría de Dempster-Shafer encontramos principalmente dos problemas: a) basar una decisión en los resultados del análisis es más complicado que cuando se utiliza una red bayesiana, puesto que con redes bayesianas cada hipótesis se asocia con una única probabilidad, mientras que en la teoría de Dempster-Shafer para cada hipótesis existen tres medidas diferentes para explicar la compatibilidad de la hipótesis con la evidencia existente y se necesitan criterios adicionales, y b) la teoría de Dempster-Shafer realiza inferencia abductiva, pero no predictiva, con lo cual no permite realizar predicciones, que tan útiles son en modelado del alumno. Sin embargo, esta teoría parece especialmente recomendable en aquellas situaciones en las que tengamos informaciones no totalmente fiables sobre el alumno, pero que aún puedan tener cierto interés: supongamos por ejemplo que queremos clasificar a un alumno como *novato*, de *nivel medio* o *experto*, y el profesor de dicho alumno nos dice que cree que sabe quién es ese alumno (con un 80% de fiabilidad), y que, si el alumno es el que él cree, no se trata de un alumno novato. En este caso asignaríamos una creencia de 0.8 al conjunto de hipótesis *{medio, experto}*, pero aún tenemos un 20% de posibilidades de que el profesor esté equivocado, así que asignaríamos 0.2 al conjunto {*novato, medio, experto*}. Este enfoque parece más adecuado que asignar probabilidades a priori P(novato) = 0.2 y P(intermedio o experto) = 0.8, porque la incertidumbre que tenemos no es sobre el nivel que tiene el alumno, sino sobre si el profesor sabe de qué alumno estamos hablando o no. Pero ninguno de los sistemas discutidos aplica la teoría de Dempster-Shafer en este sentido, sino que utilizan como evidencia las acciones directamente observables que realiza el alumno.