

PROPUESTA DE UN MODELO DE  
RED BAYESIANA  
PARA LA DEFINICIÓN DE TESTS  
ADAPTATIVOS



Nuria Gil Díaz  
Departamento de Inteligencia Artificial  
U.N.E.D.

Tesis presentada para el  
Máster en Inteligencia Artificial Avanzada  
y dirigida por  
Elena Gaudioso Vázquez  
Septiembre 2012

---

# Resumen

El uso de Internet y las nuevas tecnologías ha traído consigo nuevas posibilidades en el acceso a la información por parte de los usuarios. Los usuarios acceden a la web para la búsqueda de recursos con un objetivo claramente educativo. No sólo cuentan con el material disponible en el aula, sino que Internet les abre un campo de posibilidades con multitud de recursos y puntos de vista distintos de los diferentes usuarios.

Desde que a finales de los 80, surgen los sistemas hipermedia, los alumnos pueden evaluarse vía web de una forma mucho más productiva para el proceso de enseñanza-aprendizaje a la que se tenía con información meramente textual y en papel. Un tipo de sistemas hipermedia son los sistemas hipermedia adaptativos, que se componen de páginas web y desarrollan técnicas para mostrar al usuario en cada momento un material u otro en función de sus acciones con dichas páginas.

Las técnicas de adaptación que incorporan los sistemas hipermedia adaptativos pueden ser simples (un conjunto de reglas de navegación) o ser motores de decisión que implementen técnicas de la Inteligencia Artificial. Los tests adaptativos inteligentes (TAI) utilizan estas técnicas y centran nuestro trabajo porque los consideramos eficaces en la evaluación de los alumnos y en la propia instrucción.

Una parte fundamental de los TAI es el módulo del estudiante que tiene como objetivo que el sistema sea capaz de adaptar la instrucción de forma individualizada a cada estudiante. El modelo del estudiante, como veremos más adelante, contiene información sobre el estudiante como el conocimiento que tiene del sistema y sus gustos y preferencias.

El presente trabajo realiza un estudio del arte de los distintos métodos y técnicas utilizadas en el modelado del estudiante en cada uno de estos sistemas. Partiendo de un sistema TAI que implementa el modelado del estudiante con el uso de redes bayesianas, el presente trabajo realiza una propuesta de sistema que también se basa en las redes bayesianas pero presenta, como novedad, una nueva forma de considerar las probabilidades implicadas en la red, con el objetivo de mejorar la precisión de la estimación en el nivel de

---

conocimiento del alumno.

# Índice de figuras

3.1. Red POLA con nodos solución y redundancia [13] . . . . .	37
3.2. Red bayesiana dinámica en cada interacción del alumno con el sistema [45] . . . . .	39
4.1. Curvas CCI de 3 parámetros para 3 ítems realizadas con ELVIRA [18] . . . . .	50
4.2. Graph Author Tool de un curso adaptativo AHA [1] . . . . .	61
4.3. Relaciones entre conceptos y preguntas en Millán [37] . . . . .	63
4.4. Valores de la función $G(x)$ . . . . .	67
5.1. Paso de mensajes entre un nodo X y sus vecinos [43] . . . . .	80
5.2. Propagación de evidencia en HUGIN [32] . . . . .	85
5.3. Red bayesiana de conceptos y preguntas realizada con ELVIRA [18] . . . . .	87
5.4. Red bayesiana de conceptos y preguntas . . . . .	89
6.1. Relaciones entre conceptos y preguntas realizada con ELVIRA [18] . . . . .	97
7.1. Preguntas de la 0 a la 9 . . . . .	111
7.2. Preguntas de la 10 a la 19 . . . . .	111
7.3. Preguntas de la 20 a la 29 . . . . .	111
7.4. Preguntas de la 30 a la 39 . . . . .	112
7.5. Preguntas de la 40 a la 49 . . . . .	112
7.6. Preguntas de la 50 a la 59 . . . . .	112
7.7. Preguntas de la 60 a la 69 . . . . .	112
7.8. Preguntas de la 70 a la 79 . . . . .	112
7.9. Preguntas de la 80 a la 89 . . . . .	113
7.10. Preguntas de la 90 a la 99 . . . . .	113
7.11. Número de conceptos dominados, no dominados y no diagnosticados por ambas redes . . . . .	114

7.12. Probabilidades de los conceptos tras 2 ítems. Parte izquierda (modelo de red bayesiana de Millán). Parte derecha (modelo de red bayesiana con pesos propuesto) . . . . .	134
8.1. Estructura ASI Selection and Ordering [28] . . . . .	141
8.2. Ejemplo de test definido con la especificación QTI LITE [29] .	144
8.3. Estructura XML de un ítem QTILite [29] . . . . .	145
8.4. Estructura de un ítem de respuesta múltiple en QTILite [29] .	145
8.5. Estructura de un ítem con realimentación <i>feedback</i> en la especificación QTILite [29] . . . . .	146
8.6. testPart de la herramienta ADASAT [20] con precondiciones y reglas de bifurcación . . . . .	149

# Indice de tablas

4.1. Tabla de probabilidades condicionadas de la red bayesiana de Millán . . . . .	66
5.1. Tabla probabilidades conjuntas calculadas por el algoritmo de HUGIN . . . . .	90
5.2. Ejemplo de código para propagación de evidencia con API JAVA de Hugin Lite [26] . . . . .	91
6.1. Valores de las probabilidades condicionadas de $G(x)$ en ambos modelos bayesianos . . . . .	106
7.1. Resultados de 6 alumnos reales tras administrarles 10 ítems . . . . .	117
7.2. Resultados de 6 alumnos reales tras administrarles 20 ítems . . . . .	118
7.3. Resultados de 6 alumnos reales tras administrarles 30 ítems . . . . .	118
7.4. Resultados de 6 alumnos reales tras administrarles 40 ítems . . . . .	118
7.5. Error estándar para un alumno con Tipo de alumno 1.0 tras 40 ítems . . . . .	120
7.6. Error estándar para un alumno con Tipo de alumno 0.9 tras 40 ítems . . . . .	121
7.7. Error estándar para un alumno con Tipo de alumno 0.8 tras 40 ítems . . . . .	122
7.8. Error estándar para un alumno con Tipo de alumno 0.6 tras 40 ítems . . . . .	124
7.9. Error estándar para un alumno con Tipo de alumno 0.4 tras 40 ítems . . . . .	125
7.10. Error estándar para un alumno con Tipo de alumno 0.2 . . . . .	127
7.11. Evolución del Error estándar para un alumno con Tipo de alumno 1.0 . . . . .	128
7.12. Evolución del Error estándar para un alumno con Tipo de alumno 0.9 . . . . .	129

7.13. Evolución del Error estándar para un alumno con Tipo de alumno 0.8 . . . . .	130
7.14. Evolución del Error estándar para un alumno con Tipo de alumno 0.6 . . . . .	131
7.15. Evolución del Error estándar para un alumno con Tipo de alumno 0.4 . . . . .	131
7.16. Evolución del Error estándar para un alumno con Tipo de alumno 0.2 . . . . .	132

# Índice general

Resumen . . . . .	III
<b>Lista de figuras</b>	<b>VI</b>
<b>Lista de tablas</b>	<b>VIII</b>
<b>1. INTRODUCCIÓN AL TRABAJO</b>	<b>1</b>
1.1. Introducción . . . . .	1
1.2. Objetivos del trabajo . . . . .	2
1.3. Avance del trabajo . . . . .	3
1.4. Estructura del trabajo . . . . .	5
<b>2. LOS TESTS ADAPTATIVOS INFORMATIZADOS</b>	<b>9</b>
2.1. Introducción . . . . .	9
2.2. Los sistemas hipermedia . . . . .	9
2.2.1. Arquitectura de un sistema hipermedia . . . . .	10
2.2.2. Características de adaptación de un sistema hipermedia . . . . .	10
2.3. Los tests adaptativos informatizados (TAI) . . . . .	11
2.3.1. Antecedentes de los sistemas de tests . . . . .	12
2.3.2. Definición de test adaptativo informatizado . . . . .	14
<b>3. EL MODELADO DEL ESTUDIANTE</b>	<b>17</b>
3.1. Introducción . . . . .	17
3.2. Introducción al modelado del estudiante . . . . .	17
3.3. Tratamiento de la incertidumbre en el problema del modelado del alumno . . . . .	19
3.4. Paradigmas de modelado . . . . .	20
3.4.1. Modelado del estudiante con reglas de inferencia . . . . .	20
3.4.2. Lógica difusa . . . . .	23
3.4.3. Redes conceptuales . . . . .	27
3.4.4. Modelado del estudiante con redes bayesianas . . . . .	32
3.4.5. Comparativa entre formas de modelado . . . . .	45

<b>4. ESTADO DEL ARTE DE LOS TESTS ADAPTATIVOS</b>	<b>47</b>
4.1. Introducción . . . . .	47
4.2. Tests adaptativos modelados con la Teoría de Respuesta por Item . . . . .	47
4.2.1. Modelo logístico TRI de un parámetro . . . . .	54
4.2.2. Modelo logístico TRI de dos parámetros . . . . .	54
4.2.3. Modelo logístico TRI de 3 parámetros. El sistema SIET- TE . . . . .	55
4.3. Modelos de regla de inferencia para tests adaptativos . . . . .	60
4.4. Modelos de red bayesiana para tests adaptativos . . . . .	61
4.4.1. Redes bayesianas con modelo respuesta del ítem . . . . .	61
4.5. Conclusiones . . . . .	67
<b>5. LAS REDES BAYESIANAS</b>	<b>69</b>
5.1. Introducción . . . . .	69
5.2. Nociones básicas . . . . .	69
5.2.1. Teoría de grafos . . . . .	69
5.2.2. Conceptos básicos . . . . .	70
5.2.3. Características de los grafos dirigidos . . . . .	71
5.2.4. Formas de representación de un grafo . . . . .	72
5.2.5. Teorema de Bayes . . . . .	73
5.2.6. Definición formal de red bayesiana . . . . .	74
5.2.7. Construcción de redes bayesianas . . . . .	75
5.2.8. Inferencia en redes bayesianas . . . . .	77
5.3. Algoritmos del modelo de red bayesiana con pesos propuesto . . . . .	86
5.3.1. Algoritmo de agrupamiento para redes bayesianas . . . . .	86
5.3.2. Algoritmo de HUGIN . . . . .	88
5.3.3. Algoritmo orientado a un objetivo . . . . .	91
5.4. Justificación del uso de Redes Bayesianas . . . . .	92
<b>6. MODELO BAYESIANO DEL ALUMNO CON PESOS PRO- PUESTO</b>	<b>95</b>
6.1. Introducción . . . . .	95
6.2. Modelado del dominio . . . . .	95
6.3. Modelo del estudiante . . . . .	96
6.3.1. Nodos de la red bayesiana . . . . .	96
6.3.2. Relaciones entre los nodos . . . . .	97
6.4. Proceso de diagnóstico . . . . .	98
6.5. Diagnóstico de cada ítem . . . . .	99
6.6. Criterio de selección de preguntas . . . . .	104
6.7. Comparación de dos modelos bayesianos . . . . .	106

## ÍNDICE GENERAL

---

<b>7. EXPERIMENTACIÓN DEL MODELO Y RESULTADOS</b>	<b>109</b>
7.1. Objetivos de los experimentos . . . . .	109
7.2. Experimentación con alumnos simulados . . . . .	110
7.3. Experimentación con alumnos reales . . . . .	116
<b>8. CONCLUSIONES Y TRABAJOS FUTUROS</b>	<b>137</b>
8.1. Principales aportaciones . . . . .	137
8.2. Trabajos futuros . . . . .	139
8.2.1. Mejora del entorno real . . . . .	139
8.2.2. Uso de estándares . . . . .	140
8.2.3. Relaciones de prerequisite . . . . .	151
8.2.4. Equilibrio de contenido . . . . .	151
<b>A. Anexo I</b>	<b>153</b>
<b>Bibliografía</b>	<b>163</b>

# Capítulo 1

## INTRODUCCIÓN AL TRABAJO

### 1.1. Introducción

El trabajo presentado tiene como objetivo definir un sistema de tests adaptativos capaz de determinar con la mayor exactitud posible cuáles son los conocimientos de un alumno en una determinada materia.

Para abordar el problema, se ha visto esencial la necesidad de hacer una revisión de los distintos modelos existentes para entender qué modelo de estudiante es el que mejor puede modelar un test adaptativo.

Después de ver las características y ventajas e inconvenientes de los métodos de modelado del estudiante, hemos escogido para el modelado del estudiante, un modelo basado en redes bayesianas.

Tras hacer un estudio del arte de los modelos de redes bayesianas aplicados a los sistemas de enseñanza de tests adaptativos, hemos visto la idoneidad de este modelo en estos sistemas por ser una estructura que además de reflejar los conceptos de la materia y las relaciones entre ellos, permite hacer razonamientos sobre la propia estructura, con el fin de conocer los conceptos de la materia que el alumno domina y los que no domina. Este conocimiento de los conceptos, es uno de los objetivos de nuestro trabajo, puesto que se considera una información muy útil de cara a insistir con el estudiante en el aprendizaje de aquellos conceptos que la red bayesiana diagnostica que no han quedado lo bastante claros.

El modelo de red bayesiana de este trabajo se basa en uno que ya se había propuesto y probado anteriormente que se puede consultar ampliamente en [37], y que hemos elegido como trabajo de partida por considerar que el modelo es preciso en la estimación del nivel de conocimiento del alumno y

también por su eficacia computacional en cuanto a los cálculos necesarios para determinar los parámetros de la red bayesiana.

Debido a que el modelo de red bayesiana de este trabajo realiza solamente una modificación en el cálculo de los parámetros de la red, de ahora en adelante, en este trabajo, haremos referencia a nuestro modelo como “modelo de red bayesiana con pesos propuesto” y al modelo de Millán [37], “modelo de red bayesiana de Millán”, con el único fin de diferenciar cuando hablamos de uno y de otro.

Es sabido que los tests adaptativos informatizados (TAI) basados en la Teoría de Respuesta al Ítem (TRI) proporcionan una herramienta de diagnóstico fiable y eficiente porque dan una medida cuantitativa del conocimiento del alumno sobre el sistema [11].

La TRI es una teoría de los tests adaptativos que sustituye a la teoría clásica de tests y que presenta varias posibilidades de modelos matemáticos de cada ítem de un test adaptativo, donde cada modelo permite caracterizarlo con una serie de parámetros y adaptarlo al nivel de conocimiento del estudiante.

Como el modelo de red bayesiana se corresponde con el de un test adaptativo informatizado TAI en el que se busca la mayor precisión en el nivel de conocimiento del alumno, dicho modelo ya incorporaba en Millán [37] el modelo TRI como modelo de respuesta al ítem. El “modelo de red bayesiana con pesos propuesto” incorpora el mismo modelo de TRI que el “modelo de red bayesiana de Millán”.

## 1.2. Objetivos del trabajo

El modelo de tests adaptativos informatizados que se propone en el siguiente trabajo pretende alcanzar los siguientes objetivos:

1. Determinar el nivel de conocimiento del alumno en cada uno de los conceptos en que se divide una materia de estudio con la mayor precisión posible mejorando un modelo bayesiano basado en la TRI ya existente.
2. Ofrecer un modelo bayesiano con un método sencillo y rápido para el cálculo de los parámetros de la red.

El primer objetivo se lleva a cabo intentando mejorar la precisión de la estimación del nivel de conocimiento de los modelos de redes bayesianas presentados hasta el momento. La mejora se centra en el “modelo de red bayesiana de Millán” basado en la TRI antes comentado que se puede consultar en [37] y que se desarrolla en la sección 4.4.1

El segundo objetivo se centra en presentar un modelo de red bayesiana que además de mejorar la precisión del “modelo de red bayesiana de Millán” es una solución eficaz computacionalmente. El cálculo más importante de un modelo de red bayesiana reside en el cálculo de los parámetros de la red. Pues bien, en ambos modelos bayesianos se persigue que la determinación de los parámetros sea un proceso sencillo y automático.

### 1.3. Avance del trabajo

El “modelo de red bayesiana con pesos propuesto” tiene la misma estructura gráfica o cualitativa que el “modelo de red bayesiana de Millán”, que es una red bayesiana donde los nodos son de dos tipos: los nodos *concepto* en que se divide el conocimiento y los nodos *pregunta* que componen el test adaptativo. Como nuestro trabajo tiene como objetivo principal determinar el nivel de conocimiento del alumno en cada uno de los conceptos, y un test adaptativo muestra al usuario una serie de preguntas, serán las relaciones entre preguntas y conceptos las que nos interesen para determinar la parte cuantitativa de la red.

La parte cuantitativa de la red serán las probabilidades condicionadas de cada pregunta al conjunto de conceptos que están relacionados con esa pregunta, o lo que es lo mismo, que se consideran que el alumno debe dominar para contestar correctamente la pregunta. Los nodos de tipo concepto de la red son binarios, es decir, adoptan dos posibles estados que denotaremos como “conocido” y “desconocido”, de modo que un nodo concepto en estado “conocido” implica que el alumno domina el concepto asociado, y un nodo concepto en estado “desconocido” significa lo contrario. Así, por cada pregunta del test adaptativo hay que especificar el conjunto de probabilidades correspondiente a todas y cada una de las combinaciones de estado de cada uno de sus conceptos relacionados. En mi opinión, el hecho de que las redes bayesianas empleen probabilidades para caracterizar la incertidumbre, la convierte en una estructura que puede representar la incertidumbre que se tiene sobre el conocimiento del alumno de forma sencilla con un conjunto de probabilidades.

Como acabamos de comentar, son las probabilidades condicionadas de cada pregunta dados sus conceptos, las que la persona que entrene la red, que en nuestro caso, será un profesor, las que el profesor debe especificar al principio del sistema y que se consideran que son los parámetros de la red bayesiana. Dada la característica de que los nodos de tipo concepto son binarios, el número de probabilidades condicionadas por cada nodo de tipo pregunta de la red es un número exponencial con base binaria 2 elevado al número

de conceptos de la pregunta. Otro de los objetivos fundamentales de este trabajo es determinar un proceso de cálculo de estos parámetros que sea lo más sencillo posible y eficiente computacionalmente.

Parece razonable pensar que los conceptos que intervienen en una pregunta, no tienen todos la misma importancia sobre la pregunta; este hecho se considera en el “modelo de red bayesiana de Millán” presentando este modelo un método sencillo y rápido para el cálculo de la probabilidad de responder correctamente a una pregunta dados los conceptos que conoce y los que desconoce, teniendo en cuenta su importancia sobre la pregunta. El “modelo de red bayesiana de Millán” lo consigue con una simple ordenación de los conceptos dentro de la pregunta, de forma que el primer concepto será el menos importante de la pregunta, el segundo el siguiente más importante y así sucesivamente hasta llegar al último concepto que será el más importante sobre la pregunta. Por su parte, el “modelo de red bayesiana con pesos propuesto” ordena los conceptos mediante un factor *peso*, que habrá que indicar para cada concepto dentro de una pregunta, y que no es otra cosa que una proporción del concepto sobre la pregunta, un número entre 0 y 1.

Después de calcular para los dos modelos, “modelo de red bayesiana de Millán” y “modelo de red bayesiana con pesos propuesto”, el conjunto de parámetros de la red, en este trabajo se concluye que los dos modelos permiten calcular una probabilidad de responder correctamente la pregunta distinta para cada caso del conjunto exponencial de probabilidades. Y además, el cálculo es sencillo en los dos casos y por tanto, computacionalmente eficiente.

Una vez visto que los dos modelos permiten un cálculo rápido y sencillo de las probabilidades de la red, el siguiente y objetivo final de nuestro trabajo es demostrar que el “modelo de red bayesiana con pesos propuesto” obtiene una estimación del nivel de conocimiento del alumno cometiendo un menor error. Voy a explicar este punto.

El conocimiento del sistema que tiene el alumno se divide en una serie de conceptos. Pues bien, el conocimiento de cada concepto o en términos estadísticos hablando, la certeza que se tiene sobre cada concepto, lo calcula la red bayesiana como la probabilidad a posteriori de ese nodo. Existen algoritmos que actualizan la información que se tiene sobre los nodos de la red y cuya salida es la probabilidad a posteriori de cada nodo dada la información que influye directamente en él; son los algoritmos de propagación de evidencia de las redes bayesianas. Dependiendo del algoritmo seleccionado, también se puede mejorar la eficiencia en los cálculos. El “modelo de red bayesiana de Millán” utiliza un algoritmo de agrupamiento de los nodos de la red orientado a objetivo, cuyo propósito es reducir el subconjunto de la red sobre el que hay que propagar la información. Como la información en una red bayesiana veremos que se propaga hacia adelante y hacia atrás con respecto a cada no-

do, el algoritmo orientado a objetivo, empleará en cada momento, los nodos de tipo concepto que se estén considerando.

Nuestro trabajo también parte de este algoritmo, tras hacer un estudio del estado del arte en los distintos algoritmos de propagación existentes para redes bayesianas.

Pasamos a comentar, en el siguiente apartado, la estructura del trabajo.

### 1.4. Estructura del trabajo

El trabajo presentado se divide en ocho capítulos. Pasamos a ver una introducción de cada uno de ellos.

- En este primer capítulo, hemos comenzado con un resumen que sintetiza los objetivos y temática del presente trabajo y el estado del avance en cada uno de los objetivos marcados. Termina con una descripción breve del contenido de cada capítulo.
- En el segundo capítulo, se presentan las características principales de un sistema hipermedia adaptativo. Se explican las características que definen un sistema hipermedia y lo distinguen del resto de sistemas de aprendizaje. También se explica la arquitectura de un sistema hipermedia haciendo especial hincapié en el módulo del alumno como uno de los componentes que merecen su estudio. En este mismo capítulo, se introducen los sistemas de tests adaptativos informatizados TAI, como un caso especial de los sistemas hipermedia. Se da la definición de un TAI y de sus componentes esenciales.
- El tercer capítulo está dedicado al problema del modelado del estudiante por ser el componente más importante de un sistema tutorial y del que más depende las características de adaptación de un TAI. Se introduce el problema del modelado del estudiante y se destaca el tratamiento de la incertidumbre sobre el conocimiento que del alumno tiene el sistema como una característica que deben incluir todas las técnicas de modelado del estudiante. A continuación, se repasan las principales técnicas de modelado del estudiante y su método de tratamiento de la incertidumbre. Y en la última sección del capítulo, se revisan las diferentes aplicaciones en el contexto de los sistemas tutoriales que de las técnicas de modelado presentadas se han realicado hasta el momento.
- El cuarto capítulo hace un estudio del estado del arte en los tests adaptativos informatizados TAI. Se presentan los antecedentes de los sistemas de tests desde la teoría clásica de tests hasta los tests adaptativos

de hoy en día. Se explican las bases de la Teoría de Respuesta al Ítem como un fundamento teórico sólido que permite caracterizar un test adaptativo en base a un nivel de rasgo del alumno. Acto seguido, se hace un estudio del arte de los sistemas de tests adaptativos informatizados haciendo una clasificación en: sistemas de TAIs modelados con la TRI, sistemas de TAIs modelados con sistemas de reglas y sistemas de TAIs modelados con redes bayesianas.

- En el capítulo quinto, se presentan las características de una red bayesiana, por ser el modelo probabilístico seleccionado en el modelo del estudiante de un test adaptativo informatizado que proponemos en este trabajo. El capítulo comienza con algunos conceptos sobre grafos y las características fundamentales de una red bayesiana, que son importantes conocer, para adentrarnos después en los algoritmos de propagación de redes bayesianas que han sido útiles para definir el algoritmo de propagación propuesto para el modelo de red bayesiana presentado en este trabajo.
- En el capítulo seis, se propone un modelo de red bayesiana, al que haremos referencia a lo largo del presente trabajo como “modelo de red bayesiana con pesos propuesto”, que basándose en un modelo de red bayesiana sobre tests adaptativos informatizados con base en la TRI, pretende mejorar la precisión de la información sobre el alumno que se obtenía en relación al número de conceptos no dominados, dominados y no diagnosticados por el sistema bayesiano. El “modelo de red bayesiana con pesos propuesto” consiste en una modificación de los valores que se asignan a los parámetros de la red bayesiana del modelo bayesiano de partida.
- En el capítulo siete, se presentan los resultados de los experimentos realizados con alumnos simulados, y después con alumnos reales. Los resultados de los experimentos consisten en el número de conceptos que cada una de las redes diagnostica como dominados o no dominados. Nuestro interés siempre va a estar en obtener información acerca de los conceptos que el alumno aún no es capaz de dominar completamente para poder insistir en su aprendizaje, pero también se trata de ver qué red da la información positiva o negativa sobre los conceptos con el menor error de precisión.
- Por último, en el capítulo ocho, se proponen trabajos futuros que a partir de este modelo de red bayesiana con pesos propuesto, se pueden desarrollar con vistas a obtener más información sobre los alumnos y

también propuestas orientadas a realizar una futura investigación sobre las ventajas de estos sistemas de redes bayesianas frente a las especificaciones estándares definidas para la creación de tests adaptativos como es la especificación IMS Question and Test Interoperability.



## Capítulo 2

# LOS TESTS ADAPTATIVOS INFORMATIZADOS

### 2.1. Introducción

El trabajo presentado se centra en el estudio del modelo del estudiante de los tests adaptativos informatizados que se pueden englobar dentro de los llamados sistemas hipermedia educacionales, que son los sistemas educacionales que suscitan mayor interés en la actualidad debido al avance de las tecnologías en la web.

En la primera sección, hacemos una breve introducción general a los sistemas hipermedia para pasar ya en la segunda sección a hacer un repaso histórico y la definición de un test adaptativo informatizado.

### 2.2. Los sistemas hipermedia

Como explican Boticario y Gaudioso en [4], una de las áreas de desarrollo de los sistemas interactivos de enseñanza/aprendizaje que surgieron a finales de los años 80, son los sistemas hipermedia adaptativos.

Los sistemas adaptativos en educación son sistemas que tratan de suplir las carencias de los sistemas tradicionales de tests mediante la adaptación de su comportamiento a las metas, tareas, intereses y otras características individuales de los usuarios y de grupos de usuarios como explica Brusilovsky en [3]. Muy importante en estos sistemas, será, por tanto, su capacidad de adaptación.

Un sistema hipermedia se basa en presentar al usuario los contenidos en forma de material hipertexto. El hipertexto es básicamente un método de organización de la información de naturaleza no lineal, cuya estructura con-

siste en una red de nodos y enlaces. El usuario del sistema puede acceder a la información de forma interactiva navegando a través de los nodos de la red.

Los enlaces de un sistema hipertexto son las conexiones entre un nodo origen y un nodo destino. En un sistema hipertextual, la información contenida en los nodos es de tipo texto. En un sistema hipermedia, la información contenida en los nodos puede ser de tipo multimedia (imágenes, secuencias de imágenes, sonidos, vídeos).

### 2.2.1. Arquitectura de un sistema hipermedia

La arquitectura de un sistema hipermedia se divide en varios niveles [4].

En el primer nivel, están los dos tipos de usuarios: el usuario que accede consultando el material disponible y el autor del hiperdocumento. El primero puede consultar y navegar por la base de la información.

El software para estructuras del tipo hipertexto está formado, por un lado, por un módulo de búsqueda y, por otro lado, por un módulo de operaciones de mantenimiento, diseñados específicamente para los dos tipos de usuarios. A estos módulos se les denomina, respectivamente, módulo del alumno y módulo del profesor.

Es necesario el módulo del alumno que se construye con las características personales de cada estudiante y se actualiza durante la interacción del sistema con el mismo. Gracias a este modelo, será posible adaptar el contenido y el recorrido por el sistema a las necesidades y preferencias de cada usuario [4].

Según Boticario y Gaudioso [4], la información contenida en un sistema hipertexto puede ser recuperada por navegación o interrogación, siendo la navegación considerada como la forma primaria de acceso a la información. Una herramienta importante de la navegación es el navegador.

En una navegación por la red, la posibilidad de la desorientación es importante. Con el fin de solventar este problema, se introdujeron en estos sistemas capacidades de adaptación, de manera que pudieran guiar al alumno en su navegación por los contenidos disponibles.

### 2.2.2. Características de adaptación de un sistema hipermedia

Según Brusilovsky en [3], en los distintos sistemas adaptativos los efectos de adaptación pueden ser distintos. Por ejemplo, en los sistemas adaptativos web, la adaptación se divide en tres tipos: adaptación del contenido, adaptación de la navegación de contenidos y adaptación en la presentación al usuario.

Los tipos de adaptación que se consideran principalmente son dos [4]:

- Adaptación del contenido.
- Adaptación de la navegación.

En la adaptación del contenido, la idea es adaptar tanto el tipo de materiales multimedia utilizados para construir las páginas presentadas en cada momento a un estudiante como los contenidos de las mismas, teniendo en cuenta las características personales del estudiante, incluidas en el modelo del estudiante.

En la adaptación de la navegación, se trata de ofrecer ayuda a los usuarios de forma que puedan guiarse por sí mismos por los enlaces, adaptando la presentación de los enlaces a los objetivos, conocimientos y otras características de los estudiantes.

Boticario y Gaudioso en [4] citan 4 técnicas de adaptación de contenido:

- Texto condicional: la información relacionada con un concepto se divide en varias porciones de texto, cada una de ellas lleva asociada una condición relacionada con el nivel de conocimiento requerido para acceder a ella y que está representado en el modelo del usuario (como permite configurar el sistema AHA (Adaptive Hipermedia for All)).
- Texto expansible: cuando se selecciona una palabra resaltada o una frase, ésta se reemplaza por una porción de texto. Este texto puede volverse a comprimir y viceversa.
- Variantes de fragmentos y páginas: el sistema mantiene variaciones del mismo documento o fragmento con distintas presentaciones del mismo contenido. Cada presentación es válida para cada uno de los posibles estereotipos de usuarios.
- Técnica basada en marcos: la información de un concepto se presenta como un marco.

En resumen, el modelo del estudiante debe incluir toda la información necesaria sobre el mismo, y posteriormente, esta información se utiliza para generar cursos adaptativos utilizando (si se desea) distintas técnicas de adaptación.

### 2.3. Los tests adaptativos informatizados (TAI)

En esta sección, mostramos los tests adaptativos informatizados (TAI) como un caso particular de los sistemas hipermedia adaptativos. Los TAI son sistemas hipermedia que presentan una serie de preguntas o ítems a un

alumno para que las responda. Veremos en esta sección, la importancia de determinar una secuencia adecuada de los ítems con el objetivo de adaptar el nivel de dificultad de las preguntas al nivel de habilidad del alumno y así orientarlo y mejorar su aprendizaje.

### 2.3.1. Antecedentes de los sistemas de tests

Según I. Béjar [46], que es Investigador Científico Principal (Principal Research Scientist) y Director del Centro para el Diseño de Evaluación y Sistemas de Puntuación (Center for Assessment Design and Scoring), perteneciente a la División de Investigación (Research Division) del Servicio de Medición Educativa (Educational Testing Service, a partir de ahora ETS), y que es reconocido dentro del campo de los TAI como una de las figuras más importantes y que trabajan actualmente en ello, los primeros TAI se remontan a Frederick Lock en el ETS, quien en los años 80 empezó un amplio programa de investigación sobre los tests adaptativos.

Los primeros sistemas de tests fueron obra de Catell en el 1980 [42]. Según los autores, los tests tradicionalmente han sido utilizados por psicólogos como herramienta de trabajo. Se denomina “psicometría” a la ciencia que estudia la medición de los tests, además de otros aspectos.

La relación entre la psicología y los sistemas de tests se debe a que los tests permiten medir múltiples variables que además de con la habilidad del individuo están relacionadas con la personalidad y la genética. Asimismo, debido a la importancia de que las medidas resultado de los tests sean precisas y contengan el mínimo error, surgen las teorías estadísticas de los tests, lo cual es imprescindible para poder usar los tests de forma rigurosa y científica. Hay dos grandes teorías que sirven de base a la construcción y análisis de los tests, que son la Teoría Clásica de los tests (TCT) y la Teoría de Respuesta por Ítem (TRI).

La teoría clásica tiene sus orígenes en los primeros años del siglo XX y fue propuesta por Spearman [50]. Actualmente muchos tests psicológicos fundamentalmente se basan en esta teoría, por lo que se llevan utilizando aproximadamente 100 años; de ahí que sea necesario su estudio. Durante esta época, la evolución de los tests es rápida.

La elaboración de tests de acuerdo a la teoría clásica consistía en presentar los tests como un conjunto de preguntas consistentes entre sí y nada más, es decir, no se tenían en cuenta factores relativos a cada alumno. La consecuencia inmediata de la teoría clásica es que no existía una evaluación individualizada para cada alumno.

Se denomina Teoría Clásica de Tests, al conjunto de principios y técnicas de evaluación cuantitativa para la elaboración de tests que son aplicables de

forma estándar a una población.

Los principios en los que se basa la teoría clásica son relativamente simples y los tests clásicos han ido evolucionando en los métodos de análisis cuantitativos que empleaban.

En una primera etapa, conocida como “itemetría”, el foco de atención estaba en que los ítems tuvieran ciertas propiedades estadísticas hasta el punto de que si los ítems no cumplían con estas características, no se incluían en los tests. En esta etapa, se tomaba como medida estadística de un test su *confiabilidad*. Al principio, la confiabilidad estaba relacionada con la que correlación existente entre los ítems; si la correlación era alta, entonces se concluía que los ítems que componían el test eran adecuados. Esta forma de medida llevaba a errores por lo que se necesitaban otras técnicas para medir la confiabilidad. Un ejemplo de otras formas de medidas son las pruebas paralelas y la división por mitades. Surgieron también otras técnicas cuantitativas para la medida del error estándar, la fórmula Alfa de Cronbach, y en general, estadísticos basados en la correlación y curvas estadísticas. No obstante, todas estas medidas eran meramente cuantitativas y no tenían en cuenta las características de personalidad que distinguen a un individuo de otro y que afectan a la realización de los tests. Los psicólogos utilizan los tests para medir los atributos de las personas. Los tests son muestras de conducta que permiten llevar a cabo inferencias relevantes sobre el comportamiento de las personas.

En una segunda etapa, se utilizan otras herramientas estadísticas como el análisis factorial y las que se buscaban una relación entre los rasgos de personalidad y también poder explicar los distintos comportamientos del usuario. El resultado de esta etapa son tests estructurados que trataban de explicar mediante datos cuantitativos la relación entre componentes psicológicos del usuario, como por ejemplo, explicar cómo pueden afectar los factores ambientales a las respuestas del individuo, y también la influencia de la genética. En un test de esta etapa, cada pregunta del test estaba orientada a medir un rasgo psicológico.

Se puede afirmar, por tanto, que los tests clásicos evolucionaron desde medidas de correlación que medían la relación entre las respuestas del usuario hacia tests más elaborados que se usaban de una forma experimental para medir rasgos de la personalidad.

En cuanto a la calibración de los ítems, en el caso de los tests que medían la habilidad del alumno, se seleccionaban los ítems de acuerdo a su contenido, su nivel de dificultad y poder de discriminación. Los ítems que se buscaban eran los que poseían un mayor nivel de discriminación.

El nivel de dificultad de los ítems se elegía en función del propósito de la prueba y de la población o conjunto de individuos destino. La forma de

construir los ítems era seleccionarlos inicialmente a partir de una población inicial, lo que implicaba que si los ítems eran administrados a individuos con rasgos diferentes de la población inicial, no servían para determinar su comportamiento ante el test. Dicho de otra manera, los estadísticos que empleaba la Teoría Clásica de Tests eran dependientes de la población sobre la que se aplicaba el test. Para paliar este problema de dependencia de la población de prueba, se podía construir un banco de ítems procedente de la evaluación de grupos distintos de personas, pero esta solución no garantiza que los tests sean administrados a personas de grupos equivalentes, por lo que los estadísticos considerados en un principio no pueden extrapolarse a los nuevos grupos de personas. Por ejemplo, si los sistemas de medición de tests para dos estudiantes no tienen la misma escala, *¿cómo se puede hacer una comparación?*. Será necesario para poder comparar las dos medidas, contrastar la dificultad del test con el nivel aproximado de habilidad de los examinados.

Otra limitación de la Teoría Clásica de los Tests está en la precisión de la medición. Y es que en esta teoría, se mide la confiabilidad de un test como la correlación entre las puntuaciones obtenidas en el mismo. En la práctica, los valores de confiabilidad son más bajos de lo esperado y contienen error de medida, porque se supone que es una medida equivalente para un conjunto de examinados, cuando en realidad, las puntuaciones son distintas para alumnos con diferente nivel de habilidad.

Por otro lado, un ítem, en ocasiones, depende del resto de ítems, no estando esta relación reflejada en la teoría clásica con las medidas de confiabilidad y del error estándar. Por tanto, en el modelo clásico, no es posible aislar la contribución del ítem a la confiabilidad del test, y por consiguiente, no es posible medir su participación en la medida del error estándar.

Se concluye que a pesar de que los tests habían evolucionado con nuevos métodos estadísticos que iban más allá de la medida cuantitativa del ítem en el total del test, se hacía necesario un nuevo marco conceptual que se obtuvo con la Teoría de Respuesta por Ítem.

A continuación, vamos a ver en qué consiste un test adaptativo y la Teoría de Respuesta por Ítem como uno de sus fundamentos teóricos.

### 2.3.2. Definición de test adaptativo informatizado

Un TAI (Test Adaptativo Informatizado) o en inglés, un CAT (Computer Adaptive Test), es un test administrado por un ordenador, en el que la selección de la siguiente pregunta y la decisión de terminar el test se desarrollan dinámicamente en base al perfil del alumno que se crea y se actualiza durante la interacción con el sistema [42]. La principal ventaja de un TAI es

la disminución significativa en la longitud del test con las mismas y mejores estimaciones del nivel de conocimiento del alumno. Otra ventaja está derivada del uso del ordenador para la realización de los tests. De este modo, cualquiera que tenga acceso a Internet desde cualquier ordenador se podrá conectar al test. No podemos olvidar la gran capacidad de cómputo y almacenamiento de los ordenadores que pueden trabajar intensivamente con grandes conjuntos de datos.

Se puede definir un TAI como un algoritmo iterativo que comienza con un nivel inicial de estimación de conocimiento del alumno y que consta de los siguientes pasos:

1. Todas las preguntas (que aún no han sido presentadas en el test) se examinan para determinar cuál será la siguiente que se corresponda con el nivel actual de conocimiento del alumno.
2. La pregunta es presentada y el alumno responde.
3. De acuerdo a la respuesta dada por el alumno, se calcula para él un nuevo nivel estimado de conocimiento.
4. Los pasos 1 a 3 se repiten hasta que se cumple el criterio de parada.

De todos los pasos mostrados, se deduce que es importante determinar la respuesta del alumno a una pregunta en función de su nivel de habilidad.

No sería justo evaluar la pregunta sin tener en cuenta cuál es el nivel de conocimiento del alumno. También es importante calibrar bien la dificultad de la primera pregunta del test porque de ella derivan todas las demás. En este punto, se pueden usar distintos criterios como, por ejemplo, tomar el nivel medio de alumnos con un perfil similar que hayan pasado el test con anterioridad. Igualmente que es importante de la selección del primer ítem, será la selección de todos los demás por lo que será necesario un método para la selección de los ítems.

Por último, hay que decidir cuál es el criterio para terminar el test, es decir, para no mostrar más preguntas al alumno. También se pueden usar diferentes criterios de terminación en función del propósito del test. El test se puede terminar cuando al alumno haya alcanzado un determinado nivel de habilidad, cuando se haya presentado un número fijo de preguntas del test, etc.

Una de las teorías más sólidas usadas como fundamento en este tipo de tests es la Teoría de Respuesta por Ítem [12] que describimos en la sección 4.2 y de que puede consultarse más ampliamente en [22].

Ejemplos de TAIs usadas actualmente y que nos citan Olea y Ponsoda en [42] son: el Test of English as a Foreign Language (TOEFL), el Graduate

### 2.3. LOS TESTS ADAPTATIVOS INFORMATIZADOS (TAI)

---

Management Admissions Tests (GMAT), el Graduate Record Exam (GRE), el Differential Aptitude Tests (para la medición de aptitudes), tests para la selección de personal como el CAT-ASVAB para la selección de pilotos y controladores aéreos, tests para la admisión en centros educativos como el Law School Admission Test, para la evaluación educativa de niños y jóvenes como el COMPASS, el NWEA y el CARAT, para la evaluación de conocimientos de matemáticas de personas adultas como la prueba MATHCAT y también tests que se aplican para la obtención final de un certificado o incluso una licenciatura como medicina o enfermería.

# Capítulo 3

## EL MODELADO DEL ESTUDIANTE

### 3.1. Introducción

En el presente capítulo, se trata en primer lugar el problema del modelado del estudiante para así poder diseñar tests adaptativos que sean eficaces en su capacidad de adaptación al alumno y en la mejora de su aprendizaje. Después se introduce la necesidad de modelar la incertidumbre relativa al alumno en el modelo para tener una información completa que nos permita saber con mayor detalle su nivel de conocimiento.

Tras estas dos introducciones en las secciones 3.2 y 3.3, en la sección 3.4, se presentan las formas de modelado del estudiante más comunes que tratan con la incertidumbre del alumno, cada una de ellas de manera distinta y con menor o mayor eficacia. Dentro de cada forma de modelado del estudiante, se hace un estudio del arte de los distintos sistemas educacionales que utilizan el modelo. Los modelos del estudiante considerados son: reglas, lógica difusa, mapas cognitivos y redes bayesianas. La separación entre mapas cognitivos y redes bayesianas se hace para mostrar que hay modelos con redes conceptuales que para manejar la incertidumbre utilizan otros métodos distintos a las probabilidades de Bayes (modelo de factores de certeza, lógica difusa, otros heurísticos).

### 3.2. Introducción al modelado del estudiante

El modelado del estudiante tiene como objetivo principal proporcionar información acerca del objeto que está siendo modelado, en este caso, el estudiante que utiliza el sistema de enseñanza/aprendizaje, como afirma Self

[48]. Como dice este autor, el sistema usa el modelo del alumno para determinar las acciones más apropiadas para ese estudiante. Se puede deducir, por tanto, que es un proceso dinámico que se va construyendo con el tiempo, según Self [48], en tiempo de ejecución, durante el cual el alumno va dando datos de entrada al sistema.

Sin un modelo de estudiante [48], un sistema de aprendizaje se comportaría del mismo modo para todos los estudiantes. Nosotros sabemos que cada estudiante tiene un perfil distinto en cuanto a su nivel de conocimiento previo, sus aptitudes o habilidades, etc.

El modelo del estudiante debe tener acceso al conocimiento previo del alumno como se indica en [47]. Según este texto, la experiencia del alumno en una herramienta influye en su conocimiento en otras similares. También se afirma en este trabajo que no sólo es importante el conocimiento previo del alumno sino también el contexto de aprendizaje, así como sus habilidades, intereses, etc. En el “modelo bayesiano propuesto” del capítulo 6 presentado en este trabajo, el modelado del alumno va a contener inicialmente únicamente el conocimiento previo del alumno que va a ir cambiando a lo largo del proceso de enseñanzaaprendizaje.

Se puede deducir de esta complejidad, que el modelado del alumno es una tarea ardua que implica el conocimiento de aspectos personales del alumno, más bien cognitivos [47].

El modelo del estudiante puede ser enriquecido con los posibles pasos que el alumno debe dar para la solución de un problema y estos pasos pueden mostrarse en una estructura en forma de árbol. Con este tipo de sugerencias, el alumno va a estar mejor tutorizado lo que redundará positivamente en su aprendizaje. El estudiante va a ser quien aporte la información de cómo va realizando el trabajo a medida que interactúa con el sistema.

Para obtener mejoras en el aprendizaje del alumno, Self [47] propone que se incluya también en la resolución de un problema información al alumno sobre los objetivos que debe ir cumpliendo para guiarle mejor en lo que está haciendo. El autor presenta una herramienta que es un tutor lógico que presenta al usuario como información operadores que son las reglas de inferencia del sistema y una ventaja con las metas.

En el problema del modelado del estudiante, hay que tener en cuenta las interacciones que tienen lugar entre el conocimiento del sistema y el conocimiento que el sistema cree que tiene el alumno, como apunta Self [48]. Según el autor, el sistema no trata directamente con el conocimiento del alumno, sino que hace suposiciones acerca de lo que sabe y va aprendiendo.

Un sistema colaborativo como GUIDON (Clancey, 1979 citado en [9]) debería ser capaz de dar explicaciones desde la tarea de más bajo nivel a una explicación de mayor nivel de abstracción. Los autores comentan que es im-

portante también definir bien el nivel de “granularidad” del dominio, o lo que es lo mismo, la parte más pequeña representable del conocimiento del dominio. Y continúan diciendo que para permitir que el sistema sea capaz de dar explicaciones que se corresponden con las necesidades de los usuarios, es preciso conocer también sus metas.

Un problema que existe también en el modelado del estudiante es el modelado de los errores de concepto en el alumno. Por un lado, hay sistemas basados en el enfoque de reglas erróneas que tratan de distinguir los errores simples de los más graves como se hace en Brown y Burton, 1978 (citado en [9]). Hay muchas formas de modelar los malentendidos (Ridway, 1988, citado en [9]).

Vamos a hacer una revisión de las distintas técnicas de modelado del estudiante desarrolladas hasta la actualidad.

### 3.3. Tratamiento de la incertidumbre en el problema del modelado del alumno

Pearl [43] afirma que todos los hechos que no conocemos y situaciones que no consideramos constituyen la incertidumbre. La autora presenta los modelos probabilísticos como capaces de calcular los cambios en un hecho o evidencia al cambiar información parcial y no relevante gracias a la teoría de la probabilidad. Los modelos probabilísticos se basan en dos propiedades fundamentales:

- Relevancia: es una relación que indica un posible cambio en una variable debido al cambio de otra. La relevancia no es necesario cuantificarla numéricamente.
- Causalidad: es la representación de la relevancia mediante probabilidades.

Pearl [43] concluye que además de permitir los modelos probabilísticos representar la relevancia entre las variables definen un esquema de prioridades entre ellas para poder hacer predicciones.

De entre los modelos probabilísticos existentes, Castillo et al., [5] destacan las redes de Markov y las redes bayesianas como los dos modelos más importantes. Según los autores, con la aparición de las redes probabilísticas, la probabilidad es hoy en día, la medida de incertidumbre más intuitiva y aceptada. También señalan que las redes bayesianas presentan ventajas sobre las redes de Markov porque la factorización de la función de probabilidad se

obtiene de forma sencilla a partir del grafo dirigido considerando un conjunto de funciones de probabilidad condicionada que involucran a cada nodo con sus padres, mientras que en una red de Markov son necesarios más pasos.

A continuación mostramos cómo tratan la incertidumbre los distintos métodos de razonamiento que se aplican al problema del modelado del alumno. Damos mucha importancia a la incertidumbre presente en el conocimiento sobre el alumno, porque nos interesa en todo momento, saber con la mayor exactitud posible cuál es su grado de avance en la materia y los motivos por los que puede acertar o fallar una pregunta.

## 3.4. Paradigmas de modelado

### 3.4.1. Modelado del estudiante con reglas de inferencia

Self [50] afirma que se distinguen entre proposiciones de alto nivel que son aquéllas que afirman hechos que el sistema desea que el alumno aprenda y otras proposiciones que representan el supuesto conocimiento que del sistema tiene el alumno. Si las proposiciones no son independientes entre sí, es posible representar las relaciones entre ellas para deducir conocimiento. Estas relaciones serán un conjunto de reglas de inferencia, que puestas en un determinado orden son eficaces para determinar el nivel de conocimiento del alumno.

Las restricciones en los sistemas de reglas son de la forma *if e then h*, que representa que dada la evidencia *e* la hipótesis *h* es verdadera. A cada regla del sistema se le asigna un factor de certeza CF, que no es un valor de probabilidad, sino un *grado de creencia* en la hipótesis dada la evidencia. Los valores del factor de certeza pueden ser negativos y positivos; un valor de -1 a 0 indica un factor bajo en la creencia de que dada la evidencia la hipótesis sea verdadera y un valor de 0 a 1 indica por el contrario una creencia alta.

Cuando tenemos más de una regla en el sistema, pueden combinarse sus factores de certeza para obtener un factor de certeza final como resultado de analizar todas las reglas. Por ejemplo, si dos reglas comparten la misma hipótesis, se combinan mediante la que se denomina *función de combinación*. Y si la consecuencia de una regla es la causa de otra se combinan mediante la *función de serialización*.

En el problema del modelado del alumno, podemos tener, por ejemplo las reglas:

Si el alumno responde correctamente a la pregunta P1 entonces el alumno conoce el concepto C1. (CF = 0.9) Si el alumno responde correctamente a la pregunta P2 entonces el alumno conoce el concepto C1. (CF = 0.5)

Como ambas reglas comparten la misma hipótesis, podemos combinarlas mediante la función:

$$CF = CF_1 + CF_2 - CF_1CF_2 = 1,4 - 0,45 = 0,95 \quad (3.1)$$

En este caso, hemos deducido que las respuestas correctas las preguntas P1 y P2 respectivamente producen un factor de creencia alto en que el alumno conozca el concepto C1.

También es posible combinar evidencia mediante las operaciones de conjunción y disyunción del tal manera que en la conjunción de hipótesis, se toma como factor de creencia final el factor de la regla multiplicado por el mínimo de los valores y en la disyunción, el factor de la regla multiplicado por el máximo de los factores de las hipótesis.

Pasamos a analizar el estado del arte.

El conocimiento del sistema MYCIN [9] se ha construido a lo largo de cuatro años mediante interacciones con físicos. Este conocimiento es una colección de sentencias condicionales llamadas “reglas de producción”. Hay un total de 450 reglas. Cada regla contiene un conjunto de precondiciones llamado *premisa* que, en caso de ser cierta, se lleva a cabo la conclusión que aparece en la parte *acción* de la regla.

En MYCIN, el hecho de utilizar reglas de producción para tener una base de conocimiento rica y completa para resolver problemas complejos, lleva a pensar en las ventajas que puede tener la forma de representar el conocimiento del dominio con un sistema de reglas de inferencia. El formalismo de reglas de producción se puede aplicar a otros dominios aparte del médico por ser un lenguaje universal. GUIDON es un programa tutorial multidominio.

El sistema NEOMYCIN [8] basado en MYCIN que no solamente enriquece la base de reglas de MYCIN sino que define reglas de inferencia a distintos niveles del proceso de diagnóstico de una enfermedad, añade taxonomía y permite navegar por las reglas definidas hacia adelante y hacia atrás en función de la nueva información que recibe el sistema de consulta. MYCIN realiza esta exploración por las reglas de forma aleatoria.

Otro sistema que modela el estudiante con reglas de inferencia es el SQL-Tutor [41]. El autor presenta la herramienta SQLTutor que define un modelo del estudiante basado en restricciones. El modelado basado en restricciones se explica en (Ohlsson, 1994, citado en [41]).

En SQLTutor, el conocimiento del dominio se representa con un conjunto de restricciones donde una restricción define un conjunto de problemas equivalentes o clase de problemas ante los cuales el tutor devuelve la misma acción. Los estados en un problema son pedagógicamente equivalentes.

En este sistema, una restricción es un par ordenado  $(C_r, C_s)$  donde  $C_r$  es

la condición relevante y  $C_s$  es la condición de satisfacción.  $C_r$  representa los estados del problema donde  $C_r$  es relevante y  $C_s$  identifica la clase de estados relevantes en los que  $C_s$  se satisface. En otras palabras, si  $C_r$  se cumple en el estado de un problema, para concluir que el problema es correcto, debe cumplirse también  $C_s$ . Las condiciones de cumplimiento deben ser expresiones lógicas.

El funcionamiento del modelado del estudiante en el SQLTutor es el siguiente: En un primer momento, todos los patrones relevantes son comparados con el patrón que define el estado del problema. En un segundo paso, el patrón de satisfacción de las restricciones cuyas condiciones relevantes coincidan con el patrón del problema se considera que se cumplen también. Por lo tanto, si un patrón de satisfacción mapea el estado, la restricción se cumple y el ITS no toma ninguna acción; por el contrario, la restricción es violada. El modelo del estudiante va a contener todas las restricciones violadas.

La base de restricciones del SQLTutor es de 352 restricciones. Cada restricción tiene asignado un único número y se compone de los patrones de relevancia (parte izquierda de la restricción) y de satisfacción (parte derecha de la restricción). Por cada restricción, se guarda información de cuántas veces se encontró relevante para resolver un problema, cuántas veces la usó el alumno y cuántas veces la usó correctamente. Esta información se almacena en forma de tres indicadores (relevante, usada, correcta) que es utilizada por el módulo pedagógico del sistema para seleccionar nuevos problemas para el estudiante.

El módulo pedagógico selecciona los problemas para el alumno y lleva a cabo acciones de instrucción (realimentación) de acuerdo con el modelo del estudiante.

El nivel de realimentación indica cuánta información se proporciona al alumno. Hay 5 niveles de realimentación: positiva/negativa, de error, de pista, de solución parcial y de solución completa.

Puede suceder que la solución de un alumno viole varias restricciones a la vez. En este caso, el tutor examina las restricciones y selecciona la que al ser violada hace referencia a una confusión de concepto, que será aquella que tenga la mayor diferencia entre los indicadores “usada” y “correcta”. La razón de seleccionarla de esta manera es que esta restricción habrá sido usada de la misma forma errónea varias veces, y por tanto, la instrucción debe comenzar por esa restricción.

Al alumno se le informa de todas las restricciones que ha violado pero sólo se le indica el error de una de ellas con el criterio pedagógico de que será capaz de corregir un error en ese momento y no varios errores al mismo tiempo.

Otro criterio para seleccionar restricciones es elegir la que parezca que el

alumno por su comportamiento no está seguro de ella, que será aquella con el valor máximo en el indicador “correcta”. Otro criterio seguido es presentar al alumno un problema que requiere una restricción que no ha sido usada todavía. En este punto, el tutor permite al estudiante seleccionar el problema por sí mismo, lo cual introduce aleatoriedad en el proceso de aprendizaje de las restricciones del sistema.

El modelado basado en restricciones se usa también en el sistema KERMIT [51], un tutor para el aprendizaje de los modelos entidadinterrelación en los sistemas de bases de datos. Este tutor utiliza un modelado basado en restricciones para modelar el conocimiento del dominio y generar los modelos de los estudiantes.

Los tutores basados en restricciones modelizan los dominios de aprendizaje en un nivel de abstracción que simplifica el desarrollo de los sistemas tutoriales inteligentes [52]. Se han desarrollado sistemas basados en este enfoque que se han comercializado con éxito debido a su efectividad para ser entornos de aprendizaje adaptativos.

Las reglas de producción han funcionado bien en varios dominios como afirma Feigenbaum, 1977 (citado en [52]). Sin embargo, los autores señalan que otros enfoques para construir sistemas basados en conocimiento son posibles. Y además el enfoque basado en reglas presenta inconsistencias en el razonamiento debido a que corrompe los principios de unión y localidad, como señala Heckerman [23].

### 3.4.2. Lógica difusa

En la sección anterior, hemos tratado la representación del conocimiento en forma de hechos y relaciones, es decir, en la certidumbre o de lo que se tiene constancia del conocimiento del sistema y del alumno.

Los “conjuntos difusos” fueron introducidos por Zadeh [55] para tratar con la incertidumbre y datos no conocidos o imprecisos.

Para entender la necesidad de disponer de una lógica difusa, Mendel [36] comienza distinguiendo entre dos tipos de conocimiento que se pueden dar en cualquier problema en el mundo real: conocimiento objetivo y el conocimiento subjetivo. El conocimiento objetivo es aquél que podemos cuantificar numéricamente porque no hay duda de su valor, por ejemplo, en una variable numérica que toma valores discretos o continuos, el valor que tenga la variable en cada momento es único. En cambio, el conocimiento subjetivo hace referencia a aquellos valores que se aproximan o acercan a cada uno de los valores certeros que toma una variable en la lógica clásica. Este conocimiento subjetivo suele expresarse con lenguaje natural y se le conoce también como lingüístico. Un ejemplo es si consideramos la variable “altura” de una perso-

na, podemos decir expresiones del tipo: “María es lo bastante alta”, “María es muy alta”, para comparar su altura con una situación del mundo real como puede ser que María llegue a ser jugadora de baloncesto. Se usan, por tanto, variables lingüísticas para analizar y modelar sistemas.

En el ejemplo del baloncesto, será necesario también un método que permita asociar los valores lingüísticos con los valores numéricos que forman parte del dominio original de la variable. Tal asociación la realiza un componente fundamental en un sistema de lógica difusa que es la función de membresía.

Dicho de una forma general, la función de membresía o de pertenencia, asigna a cada elemento de un conjunto su *grado* de pertenencia a uno de los valores del conjunto. El rango de pertenencia de cada elemento al conjunto puede variar en el intervalo  $[0,1]$ , es decir, no se sabe con exactitud si el elemento pertenece o no al conjunto difuso, como sucedía en la teoría clásica de conjuntos. Es por esto que podemos considerar un conjunto más grande que contiene al subconjunto difuso.

Los conjuntos sobre los que se definen las funciones de membresía se les denomina *conjuntos difusos*. Un conjunto difuso  $F$  en otro  $U$  se representa con un conjunto ordenado de pares de un elemento genérico  $x$  y su función de pertenencia:  $F = (x, \mu_F(x)) \mid x \in U$ . Por tanto, no todos los elementos  $x$  forman parte del conjunto  $F$  sino que solamente los que verifican la función de pertenencia  $\mu_F(x)$ .

Análogamente a la teoría clásica de conjuntos, se definen operadores para los conjuntos difusos. Entre las operaciones más importantes destacan la unión, la intersección y el complemento, que se pueden definir de manera distinta en función del dominio de aplicación y que no coinciden con las operaciones de los conjuntos clásicos.

La función de membresía la define el usuario del sistema de lógica difusa según sus necesidades, aunque las más habituales son: triangular, trapezoidal, lineal a trozos y gaussiana.

Además, las funciones de membresía pueden ser traducidas directamente a reglas de producción en las que el antecedente y consecuente contienen respectivamente, el valor numérico y el valor difuso. Por ejemplo, una regla difusa puede ser la siguiente: SI la persona mide 1.80 ENTONCES la persona es una persona ALTA

que se corresponde con el valor  $\mu_F(0,8)$ .

Por lo que hemos comentado, los elementos fundamentales que son necesarios definir en un sistema de lógica difusa son:

- Variables lingüísticas. En realidad, es una fase de conversión de variables numéricas a difusas.
- Conjuntos difusos. Los conjuntos difusos son los elementos del dominio

de cada variable lingüística.

- Reglas difusas. Las reglas difusas permiten relacionar los conjuntos numéricos con los conjuntos difusos.
- Método de conversión de conjuntos difusos a conjuntos numéricos. Normalmente la salida de los sistemas es numérica, aunque también puede asociársele otro valor que no sea difuso.

En el problema del modelado del alumno, podríamos considerar como variable difusa  $X = \text{niveldeconocimientodelalumnoenunconceptodelamateria}$ , que puede tomar como valores  $X = \text{bajo, medio, alto}$ . La variable  $X$  se determina en función del porcentaje sabido del concepto por el alumno, por lo que la función de membresía es:  $\mu(\text{porcentajeconcepto}) = \text{bajo, medio, alto}$ .

Con esta función de membresía, podemos tener como regla difusa:

SI el conocimiento del alumno en el concepto  $i$  es *bajo* ENTONCES se presenta al alumno con *casi seguridad* otra pregunta del concepto  $i$

Las reglas difusas contienen dos o más afirmaciones difusas. Se pueden aplicar diferentes técnicas de inferencia sobre las reglas difusas como la técnica máx-min o la técnica del producto máximo que dan como resultado un conjunto difuso. Es necesario transformar el conjunto difuso en un conjunto nítido como salida del sistema, para lo que se pueden emplear varias técnicas como el método del máximo (que selecciona el punto del dominio donde la función de membresía alcanza su valor máximo) y el método del centroide (que selecciona el punto del dominio que divide verticalmente por la mitad al conjunto).

Pasamos a analizar el estado del arte.

KNOME [7] es el módulo de usuario del sistema UC, un sistema de consulta de lenguaje natural sobre el sistema operativo UNIX. El propósito del sistema es ofrecer un sistema adaptativo que detecte el nivel del usuario y sea capaz de adaptar sus respuestas al nivel.

El autor discute distintas formas de modelado del usuario para concluir que el modelo del usuario de un sistema de consulta como es UC, no puede ser un subconjunto del modelo del dominio (modelo de superposición), ni un sistema que tome acciones en base a las preferencias y/o necesidades del usuario que se indican al principio de la sesión, ni un sistema que siga planes de acción y metas en función de las acciones del usuario, sino que debe ser un sistema capaz de inferir lo que el usuario conoce y no conoce de UNIX, en este caso, en cada momento en el tiempo.

KNOME representa lo que cree que sabe el usuario. Para ello, utiliza una categoría o prototipo para el usuario y así después poder hacer inferencias

según su valor. La categoría de usuario es una variable lingüística que toma como valores: novicio, principiante, intermedio y experto.

Además de categorizar a los usuarios, KNOME clasifica la información en niveles de dificultad. De este modo estereotipa los comandos, los formatos de los comandos, la terminología en función de su nivel de dificultad. El nivel de dificultad es otra variable lingüística que toma como valores: simple, mundano, complejo y esotérico. El nivel simple incluye las características que se suponen que se aprenden de forma rápida cuando se ve por primera vez el sistema UNIX. El resto de valores representan un valor de dificultad mayor que el inmediato anterior, excepto el valor esotérico que se utiliza para usuarios con necesidades especiales; por ejemplo, que necesiten el uso específico de un comando que tenga una aplicación específica como es el comando *spice* usado en simulación de circuitos.

Una vez que la información se clasifica con un nivel de dificultad, las inferencias basadas en los niveles se representan como relaciones entre la categoría del usuario y el nivel de dificultad.

El uso de un doble sistema de estereotipado le permite a KNOME agrupar las inferencias que van por defecto para cada grupo juntas y de este modo, se ahorra tener que representar cada modelo de usuario individual junto con sus inferencias específicas, lo que es eficiente computacionalmente.

Además, para tratar con la incertidumbre, KNOME utiliza variables lingüísticas para indicar las medidas de certeza de sus inferencias en lugar de variables numéricas. Son valores de certeza: probablemente, improbablemente, muy probablemente, muy improbablemente, algo probable, algo improbable, verdadero, falso. Dentro de las inferencias pueden haber los predicados: unos pocos, la mayor parte, todos y ninguno. Por ejemplo, si un alumno conoce *todos* los conceptos asociados con un nivel de dificultad, entonces el sistema infiere como *verdadero* cualquier hecho de este mismo nivel que se le presente.

El tutor de matemáticas MDF que trata de guiar al alumno en el aprendizaje de números mixtos, fracciones y decimales. Cada tipo de problema del dominio es considerado un *tópico*. Hay 14 tópicos en el tutor y cada tópico tiene asociado una lista de pretópicos que deben ser dominados por el alumno antes de presentarle un problema del tópico. Además cada tópico tiene asociado una lista de habilidades que son los pasos en la resolución de un problema del tópico considerado. Por ejemplo, para sumar matrices, el estudiante debe saber calcular el mínimo común múltiplo, convertir las fracciones en equivalentes, sumar numeradores, simplificar y formatear el resultado.

El modelo del estudiante del tutor MDF almacena la variable *rendimiento* (proficiency) por cada tópico del dominio. Los autores de la herramienta consideran que un valor numérico para el rendimiento del alumno no aporta suficiente información sobre su progreso en cada uno de los tópicos. MD

utiliza elige para representar la incertidumbre *vectores de certeza* por cada tópico. Cada vector contiene 7 puntos y todos ellos suman 1. El valor de cada punto indica la probabilidad aproximada de que el estudiante tenga ese nivel de rendimiento. De este modo, el valor más bajo del vector es (1000000) y el más alto (0000001).

Para actualizar los vectores de creencia, el tutor utiliza otros factores como son las pistas que se han mostrado al alumno y los factores de adquisición y de retención. Cada pista tiene asociado un índice de información 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 retención.

El tutor también determina por cada tópico el número de habilidades implicadas y su nivel de dificultad en función del nivel de rendimiento del alumno. Así el tutor proporciona contenido adaptado al nivel del alumno y selecciona los temas objetivo.

A pesar de estar resuelto el problema de la adaptación, los autores en reconocen que el esquema de vectores propuesto no tiene ninguna base teórica sólida, con lo que su validez está pendiente de comprobar. Más aún, el tutor internamente en lugar de emplear el vector de creencias lo colapsa en un único valor numérico que se calcula multiplicando cada índice del vector por su componente y después se suman todos los valores. Es después este valor numérico el que se usa para medir el nivel de rendimiento del alumno en cada tópico. Parece razonable pensar que se podría aprovechar mejor tener un vector como medida del rendimiento.

En cuanto a la efectividad de los sistemas basados en lógica difusa, hemos de comentar que no han sido probados de forma rigurosa lo cual puede llevar a situaciones imprevisibles. Además, solamente se ha demostrado que solamente son más eficaces cuando se trata de un sistema completo que trabaja además de con conocimiento difuso con reglas difusas [38].

### 3.4.3. Redes conceptuales

Hemos podido encontrar literatura extensa acerca de sistemas que modelan el conocimiento del alumno y las estrategias de aprendizaje en una red conceptual que contiene relaciones causales entre los distintos elementos del dominio relacionados. Desde mi punto de vista, todos los sistemas basados en esta estructura pretenden representar más situaciones que pueden darse durante el aprendizaje y también mejorar la toma de decisiones. En la im-

plementación de más bajo nivel de estos sistemas, puede que utilicen algún heurístico para evaluar la información disponible, factores de certeza e incluso lógica difusa. A continuación presentamos ejemplos de redes conceptuales para el modelado del alumno con una lógica subyacente en algunos casos distinta, con la intención de resaltar los mapas cognitivos como una técnica propia de modelado.

Weiss, Kulikowski y Safir [52] presentan el sistema CASNET como un sistema que utiliza una red con relaciones causales y de asociación para representar las situaciones de un paciente con el objetivo de determinar la diagnosis y mejor terapia recomendada en cada visita al médico.

El objetivo del sistema CASNET es la mejora del sistema de reglas MYCIN en cuanto a que la red causal proporciona un método capaz de razonar más allá de la mera asignación de un paciente a una categoría de enfermedad en un proceso dinámico que cambia el estado del paciente a medida que se va teniendo nueva información.

Según el estudio de CASNET, el modelo de una red causal es muy adecuado en la representación de las posibles causas de un glaucoma y las terapias a aplicar porque se conocen muy bien. El dominio está muy bien determinado.

El conocimiento del sistema se representa con tres tipos de datos: observaciones del paciente, estados de la red causal y categorías de la enfermedad (diagnóstico, pronóstico y terapia). Cada diagnóstico está asociado con un conjunto de terapias ordenadas como una lista de preferencias de tal modo que cada terapia en la lista está asociada con un grado más avanzado de la enfermedad.

Así tenemos que las observaciones son la evidencia directa obtenida de un paciente. Los estados fisiopatológicos son constructos intermedios que describen las condiciones que resultan de las observaciones. Y en un nivel de abstracción mayor están las categorías. Se dan las siguientes relaciones entre elementos: observacionesestado, estadoestado y estadocategoría.

En CASNET interesan dos tareas fundamentales: la interpretación del diagnóstico y la recomendación de una terapia.

El diagnóstico es interpretado mediante las observaciones de un paciente referidas a los estados fisiológicos que le corresponden. Cada estado puede tomar tres valores (confirmado, denegado, indeterminado) y este valor se toma como hipótesis. Los diagnósticos son activados por la configuración de varios estados y denegados dentro de la red causal. La evaluación del estado de un paciente es un proceso dinámico. En este modelo no se especifican pesos a pesar de que cada hipótesis puede tener varias medidas.

En cuanto a la recomendación de una terapia, para elegir una terapia específica se tienen en cuenta los detalles de cada paciente. Cada terapia es asignada un peso derivado de las observaciones, indicando una medida de la

confianza en el éxito del tratamiento. El tratamiento con el peso más alto es seleccionado de la lista de tratamientos. Este esquema permite la codificación de condiciones para seleccionar una terapia.

En mi opinión, en el caso de CASNET, la representación en forma de árbol permite aunar en una representación única todas las posibles situaciones que pueden darse entre un paciente y sus posibles diagnósticos así como la influencia de las terapias seleccionadas en los estados de los pacientes.

Hay sistemas que organizan el conocimiento en una red jerárquica. Tal es el caso de un sistema tutorial inteligente de aprendizaje colaborativo como es LCS (Learning Companion System) [6] cuyo conocimiento es más complejo porque modela dos agentes de aprendizaje: el propio alumno y un agente de compañía que interactúa con el alumno con el objetivo de motivarle en una tarea de colaboración y competición al mismo tiempo.

En este sistema, cada agente es un conjunto de reglas, que se representan como reglas de producción. Sin embargo, las actividades de aprendizaje del dominio se pueden describir en tres niveles de abstracción:

- Nivel global, que es el nivel del curriculum. El curriculum es el conjunto entero de actividades que se corresponden con el conocimiento del dominio y los objetivos de aprendizaje.
- Nivel de protocolo, que organiza las actividades en un determinado formato.
- Nivel de episodio. Un episodio es una unidad básica de las actividades.

En resumen, un curriculum contiene un conjunto de protocolos, un protocolo es un conjunto de episodios y un episodio es una unidad básica de conocimiento. Se puede representar esta estructura en un árbol, donde cada nodo es un conjunto de reglas.

En el momento de aprendizaje de un episodio, las reglas de los diferentes agentes relacionadas con ese nodo se ejecutan. Después de ejecutar el episodio, se ejecutan las reglas de planificación que deciden qué nodo “episodio” se visita a continuación.

El árbol se llama también “mapa de situación”. Chan [6] define un mapa de situación como un grafo donde cada nodo representa una situación en la resolución de un problema y se rellena con los detalles de esa situación. Un enlace del grafo relaciona dos situaciones de un problema y por tanto cuál es la situación previa a la otra dada en el problema. El agente de compañía toma el grafo como guía para elegir una acción entre diferentes situaciones.

El estudio del sistema LCS concluye que un mapa de situación representa explícitamente las situaciones implícitas en la resolución de un problema con

lo que se puede aplicar de forma general a diferentes problemas. El mapa es capaz de capturar además la historia de situaciones pasadas.

No obstante, según Chan ningún problema del estado del arte es capaz de representar situaciones complejas (que LCS sí lo hace) e intuiciones humanas (que está pendiente) al mismo tiempo.

Esta organización jerárquica del conocimiento en forma de árbol se lleva a cabo en los cursos de aprendizaje web elearning.

El modelado del estudiante que se utiliza en los sistemas de aprendizaje web es el modelado de superposición multicapa [37]. En este método, afirma Weber [53] que el conocimiento de las unidades a ser aprendidas se organizan en una red conceptual. Las unidades se organizan jerárquicamente en lecciones, secciones, subsecciones y páginas terminales. Las páginas terminales pueden introducir nuevos conceptos, presentar listas de ítems y ofrecer problemas para su resolución. Cada unidad es un objeto que contiene trozos de texto de la unidad con la correspondiente página e información para relacionar las unidades y los conceptos.

El modelo de usuario se representa con un modelo multicapa. La primera capa describe si el usuario ya ha visitado la página asociada al concepto. La segunda capa contiene información sobre qué ejercicios e ítems de test relacionados con el concepto el usuario ya ha trabajado y si lo hizo bien. La tercera capa describe si un concepto puede inferirse mediante links de referencia desde otros conceptos que dominó previamente. La última y cuarta capa describe si el usuario ha marcado el concepto como ya sabido. La información en cada una de las capas es actualizada independientemente. Este modelo multicapa soporta la anotación adaptativa de links y el secuenciamiento del curriculum individualizado.

Un concepto es anotado como “ya aprendido” si se han realizado suficientes ejercicios o ítems de test relacionados con el concepto. Un concepto es anotado como “inferido” si el concepto no está todavía aprendido o si fue inferido como aprendido a partir de otros conceptos. Un concepto es anotado como “marcado como sabido por el usuario” si el usuario lo marcó como sabido y no hay información de que el concepto es “ya aprendido” o “inferido”. Finalmente, un concepto es anotado como “no preparado para ser visitado” si ninguno de los anteriores estados del concepto se cumple.

En este sistema, la anotación adaptativa de links es sólo usada como una pista desde el punto de vista de que un alumno puede visitar una página incluso cuando no es recomendable que lo haga. El secuenciamiento de contenidos significa que la página calculada por el sistema para ser mostrada a continuación se calcula dinámicamente en función del objetivo general de aprendizaje y del estado de los conceptos aprendidos. La siguiente página sugerida pertenecerá al concepto al que no se le asignan los tres primeros

estados y que será el siguiente preparado para ser aprendido.

Además de este conocimiento del dominio usado para el secuenciamiento de contenidos, la herramienta ELM-ART [53], utiliza un modelo llamado episódico del alumno para poder explicar los errores cometidos. La representación del conocimiento en el modelo episódico consiste en conceptos y reglas. Los conceptos se refieren a conocimientos del lenguaje de programación LISP (en concreto, procedimientos y conceptos semánticos). Las reglas describen diferentes formas de solucionar el problema planteado por el concepto. En el modelado episódico de Weber no se emplea un modelo de factores de certeza, sino que según Millán [37] en un correo electrónico Weber explicó que utilizaba un heurístico que habían desarrollado ellos mismos.

Así, el modelo del estudiante se compone de una colección de episodios que son descripciones de cómo el alumno ha resuelto el problema. Estas descripciones son estructuras de explicación de cómo una tarea de programación ha sido resuelta por el alumno.

Los episodios no se almacena como un todo, sino en fragmentos llamados *snippets*. Cada snippet describe un concepto y una regla. Estos snippets se almacenan como trozos con respecto a los conceptos del conocimiento del dominio. De este modo, el modelo episódico del alumno está interrelacionado con el conocimiento del dominio. Para construir el modelo del alumno, el código es analizado para determinar los conceptos y las reglas empleadas en la solución de un problema.

Según Weber y Spetcher [54], en los sistemas tutoriales web, los sistemas más avanzados son aquéllos que se basan en el modelo del estudiante episódico y también los que ofrecen ayuda individualizada. No realiza un diagnóstico del problema continuamente, sino que solamente cuando el usuario lo solicita, lo que hace el modelo más adecuado para entornos web. Esta última afirmación de Weber se refiere a que en un entorno web lo más importante es el tiempo de respuesta del sistema y si se estuvieran calculando enlaces de forma continua, el tiempo de respuesta aumentaría.

Un ejemplo de mapa cognitivo que utiliza lógica difusa es el tutor de química ML-Modeler [21], que se apoya en una red de conceptos abstractos, sus relaciones y estrategias para resolver problemas. En una única red, en realidad hay representadas más de una, denominándose cada una de ellas MOP. La nueva información que se añade a la red es en forma de MOP, extendiendo y contrayendo aquellas partes de la red que no son necesarias en cada momento. Cada MOP representa un problema, su solución y los conceptos usados para resolverlo y para ello contiene proposiciones de la materia, ecuaciones y procedimientos.

Los enlaces de la red conceptual indican estado de un atributo, conocimiento factual y conceptos de química.

El sistema ML-Modeler representa la incertidumbre con variables difusas que con valores de probabilidad. Gürer, desJardins y Schlager [21] descartan el uso de una red bayesiana por motivos de simplicidad y porque en un futuro el paso de una estructura a la otra es casi directo.

El sistema lógico usado por ML-Modeler se basa en las funciones de probabilidad difusas (fpds) que se utilizaron en el sistema Sherlock II. Estas funciones están asociadas a variables que se corresponden con conceptos de la red. También se asignan fpds a los enlaces y se usan los enlaces para calcular fpds.

ML-Modeler utiliza 7 valores (definitivamente, probablemente, posiblemente, quizás, posiblemente no, probablemente no, definitivamente no) en las fpds que están asociadas a cada concepto y enlace en la red. Una fpd se representa como una tupla  $(p_1, p_2, p_3, p_4, p_5, p_6, p_7)$  donde  $p_1$  es la probabilidad de masa asociada con el valor definitivamente no y  $p_7$  es la probabilidad de masa asociada con el valor definitivamente. Estos valores se actualizan con la información inicial del alumno y con la aplicación de hipótesis, utilizando funciones de actualización “upgrading” y “downgrading” para aumentar un valor de probabilidad de masa o disminuirlo respectivamente. Las funciones de “upgrading” y “downgrading” se muestran a continuación:

$$p_i = p_i - p_i c + p_{i-1} c \quad (3.2)$$

$$p_i = p_i - p_i c + p_{i+1} c \quad (3.3)$$

Por ejemplo, la hipótesis de que el alumno ha utilizado una determinada ecuación para resolver un problema haría que todos los conceptos y enlaces relacionados se actualizaran con la función “upgrading” y el resto se degradaran con la función “downgrading”.

#### 3.4.4. Modelado del estudiante con redes bayesianas

Una red bayesiana [5] es un par  $(D, P)$ , donde  $D$  es un grafo dirigido acíclico,  $P = p(X_i | pa(X_i))$ , con  $i = 1, \dots, n$ , es un conjunto de  $n$  funciones de probabilidad condicionada, una para cada variable, y  $pa(X_i)$  es el conjunto de padres del nodo  $X_i$  en  $D$ . Los parámetros de la red son  $p(X_i | pa(X_i))$ . El conjunto  $P$  define una función de probabilidad asociada mediante la factorización:

$$p(x) = \prod_{(1,n) \in i} p(x_i | pa(X_i)) \quad (3.4)$$

El conjunto de la red bayesiana será la probabilidad condicionada conjunta de todas las variables aleatorias del grafo. Existen diversos tipos de redes bayesianas dependiendo del tipo de variables que contenga el grafo (discretas, continuas o ambas) y del tipo de distribución que se considere para cada variable.

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

- Un conjunto de variables  $X_1, X_2, \dots, X_n$ .
- 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  $X_i$ , su probabilidad condicionada al conjunto de sus padres.

En el problema del modelado del alumno, las variables de la red bayesiana puede representar el grado de conocimiento del alumno en un determinado concepto de la materia o también, por ejemplo, si el alumno ha sido capaz de responder correctamente una pregunta o un ejercicio. Pueden tomar valores binarios (conocido/desconocido), (correcto/incorrecto), discretos (mal/regular/bien) o continuos donde el valor es un número entre 0 y 1.

Dadas las características de las redes bayesianas de representar todas las relaciones existentes entre las variables, son un mecanismo que permiten hacer deducciones (razonar sobre las variables que influyen en otras) y sacar conclusiones (razonar sobre los efectos producidos en unas variables debido a otras) a medida que se va obteniendo nueva información o evidencia acerca del alumno.

Pasamos a analizar el estado del arte.

La primera propuesta de usar redes bayesianas en el modelado del alumno aparece en Villano (1992, citado en [38]). En [33] se explica la necesidad de tener un modelo del estudiante probabilístico para poder representar la incertidumbre sobre el nivel de conocimiento del alumno. La incertidumbre está causada por errores de descuido o aciertos aleatorios en las respuestas del alumno y otras respuestas inesperadas por el sistema.

Vamos a ver los modelos basados en redes bayesianas de tres sistemas que están relacionados entre ellos: sistemas OLAE, ANDES y POLA.

El sistema OLAE [36] es una herramienta de evaluación que recopila datos de los estudiantes en la resolución de problemas en clases de física, analiza los datos con métodos probabilísticos que calculan el conocimiento del alumno y presenta los resultados del análisis de una forma muy flexible.

Por cada problema, OLAE crea automáticamente una red bayesiana que relaciona el conocimiento representado como reglas de primer orden hacia

acciones particulares, como pueden ser, la presentación de ecuaciones escritas por parte del sistema. El conocimiento del dominio en OLAE es un conjunto de reglas.

El modelo del estudiante es el conjunto de reglas que indican lo que el estudiante conoce del dominio. Cada regla se corresponde con una parte del dominio y pueden ser correctas e incorrectas. Varias reglas se pueden agregar para componer el modelo del alumno y el sistema tomar decisiones posteriormente en base a esas reglas.

Pero con este esquema simple de reglas, hay situaciones del aprendizaje no contempladas. En primer lugar, el alumno puede cometer errores no intencionados que simulan una regla del sistema que no debería haberse producido. En segundo lugar, el alumno puede acertar por azar respuestas. Y por último y en tercer lugar, en el caso de que haya varios caminos posibles para la solución de un problema, es muy difícil diagnosticar cuál es el camino que ha seguido que explica su conocimiento.

Para poder tratar con esta incertidumbre, el sistema OLAE utiliza las redes bayesianas. El enfoque bayesiano además de permitir una clasificación de las reglas, permite decidir en función de la frecuencia de las reglas.

OLAE presenta al alumno cinco tipos de actividades basadas en ordenador, una de las cuales es la resolución de problemas de física, que es en la que se centra el modelado de la red bayesiana. En base a estas actividades, OLAE recopila las acciones del alumno y la duración de las acciones.

Para poder razonar sobre el conocimiento del alumno en un nivel de abstracción superior a las reglas, el sistema define *factores*. Un factor es una función de las probabilidades de un conjunto específico de reglas y representa el dominio del alumno sobre ese conjunto. Por ejemplo, un factor puede ser el conjunto de reglas necesarias para apostar que se tiene un subconjunto del conocimiento. Entonces, el factor hace referencia a ese subconjunto.

En la red bayesiana de OLAE, hay cuatro tipos de nodos:

- Nodo de tipo regla, que representa si el alumno conoce una regla del dominio.
- Nodo de tipo aplicación, que representa que el alumno ha utilizado una regla en la resolución de un ejercicio.
- Nodo de tipo hecho, que representa si el alumno sabe un determinado hecho o concepto acerca del problema.
- Nodo de tipo acción, que representa que el alumno ha realizado una acción.

Los nodos de tipo hecho incluyen ecuaciones. Estos nodos se conectan con enlaces directos en la red. Los caminos que determinan los enlaces son las distintas vías que un alumno sigue en la resolución de un problema. Una vez que el alumno responde a un ejercicio, los algoritmos de propagación de la red utilizan los pesos de los enlaces para calcular la probabilidad a posteriori de que el alumno conozca una determinada regla del dominio. Así es como se lleva a cabo el proceso de inferencia en la red bayesiana.

La red bayesiana de OLAE tiene 150 nodos. Cada vez que una regla es aplicada por el alumno para dar una solución, se introduce una regla de aplicación en el grafo y un enlace que une el nodo de tipo aplicación y el nodo de tipo hecho implicado. Por cada antecedente (el hecho cuyo conocimiento dispara la regla), se introduce un enlace desde el nodo de tipo hecho al nodo de tipo aplicación. Si además un hecho tiene asociado directamente una acción, se añade un nodo de tipo acción y un enlace desde el nodo de tipo hecho al nodo de tipo acción. Los nodos de tipo acción son, por tanto, nodos terminales de los que no sale ningún enlace.

Las probabilidades a priori de la red son valores de una distribución uniforme, que para un conjunto grande de datos, se pueden estimar empíricamente.

Las probabilidades condicionadas se construyen a partir de funciones de puertas lógicas conocidas como AND y XOR que operan con las entradas de las reglas. La puerta AND representa que una regla se aplica si se cumplen todos y cada uno de sus antecedentes. Por su parte, la parte XOR se asocia con un nodo de tipo hecho para representar que un mismo hecho puede ser conocido a partir de reglas distintas que siga el alumno en su proceso de aprendizaje.

El modelo del estudiante en OLAE es una red que se genera a partir de la información de evaluaciones anteriores del alumno. El tamaño de esta red puede ser enorme y poco factible computacionalmente, por lo que se queda con una pequeña parte. El modelo del conocimiento del problema actual se utiliza para determinar el conjunto de ecuaciones que un alumno puede aplicar en la solución del problema. Estas ecuaciones se almacenan junto con los hechos a los que hacen referencia en una tabla hash.

Cuando el alumno introduce una ecuación como entrada en la interfaz del sistema, dicha ecuación es buscada en la tabla hash para recuperar el nodo de tipo hecho. Un nodo de tipo acción que representa la acción del usuario se añade a la red del dominio con un enlace que viene desde el nodo de tipo hecho. Al enlace que une ambos nodos se le asigna una probabilidad de 1.0. La red del dominio se ve, por tanto, aumentada como consecuencia de las acciones del alumno. El algoritmo de propagación de la red se ejecuta y se recalculan las probabilidades de cada nodo, dada la nueva evidencia o hecho conocido por el alumno.

El sistema OLAE tiene como principal ventaja que genera una red automáticamente con todas las características necesarias para llevar a cabo un proceso de evaluación. Pero tiene como inconveniente que está limitada al dominio científico ya que es capaz de tratar solamente con ecuaciones. Y finalmente, sólo permite diagnosticar el conocimiento del alumno con respecto al dominio pero no otras características del alumno como son sus hábitos de estudio.

Un sistema que está basado en OLAE es el sistema POLA [13] que además de mantener información probabilística del conocimiento del alumno, guarda la información del proceso seguido por el alumno en la resolución de un problema también de física.

Para que POLA sea capaz de saber los pasos del alumno, por cada ecuación que él introduce en el sistema como solución al problema, POLA debe inferir las reglas que se han utilizado y los hechos derivados de esas reglas que han llevado a esa ecuación. De hecho, para encontrar todas las reglas y conclusiones que se han usado para generar la ecuación, será suficiente con localizar el nodo de tipo hecho que se corresponde con la ecuación y todos sus nodos antecesores.

Para el caso de reglas incompletas, consideradas aquéllas que no tienen asignaciones de valores en todas sus variables implicadas, POLA calcula por cada ecuación introducida por el alumno, todas las reglas derivadas de la ecuación y selecciona los nodos de tipo aplicación afectados en la red.

La topología de la red OLAE tiene embebidos todos los posibles caminos que el alumno puede realizar al resolver el problema, por lo que es una estructura idónea para este tipo de problemas de seguimiento del progreso de los alumnos. Pero esta topología no es adecuada en POLA, porque cada vez que se instancia un nodo de tipo acción debido a que el alumno ha introducido una ecuación, la propagación de las probabilidades de la red se lleva a cabo a través de todos sus enlaces. Esto implica que aumentará el valor de las probabilidades en los nodos descendientes, indicando de forma errónea, las acciones y hechos derivados de las acciones de los alumnos.

Una revisión de la red bayesiana de OLAE es la adición de los nodos denominados “nodos de derivación”, cuyo significado es un posible camino de resolución a partir de un nodo de tipo acción. De esta manera, cada vez que el alumno introduce una nueva ecuación, la red bayesiana se extiende añadiendo el “nodo acción” asociado a la ecuación y un “nodo derivación” por cada camino alternativo que lleva la ecuación. Los nodos de tipo derivación son conectados con el nodo de tipo acción a través de una matriz leaky-XOR. Cada nodo de derivación es conectado con el conjunto de nodos de aplicación asociados con una matriz de enlaces AND. Asimismo, cada nodo aplicación es conectado con el correspondiente nodo de tipo regla con una

matriz leaky-AND.

Al tener los nodos de derivación, las alternativas para cada ecuación están representadas de forma separada en la red, con la subsiguiente propagación de la evidencia independiente para cada camino. El proceso de propagación se simplifica debido a que la red se construye incrementalmente.

A pesar de haber conseguido esta mejora en la eficacia del proceso de propagación, la red con los nodos de derivación se vio que tiene deficiencias en cuanto a la distribución de la última evidencia debido a esta separación en caminos. Por ejemplo, como cada camino de derivación tiene una estructura distinta en cuanto al número de nodos ancestrales que lo forman, la evidencia de cada nodo de derivación se distribuye inversamente proporcional al número de padres; esto conlleva que ante una evidencia, siempre se infiera como nodo de derivación aquel con el número menor de padres.

Estos problemas estructurales se resuelven con la adición de “nodos solución” que representan las soluciones diferentes para un problema. Además de los nodos de solución, se añaden “nodos de redundancia”, que indican la probabilidad de que el alumno esté siguiendo una solución o más de una. Cada nodo aplicación de una derivación se conecta con el correspondiente nodo solución y también con el nodo de tipo regla asociado, como puede apreciarse en la siguiente figura:

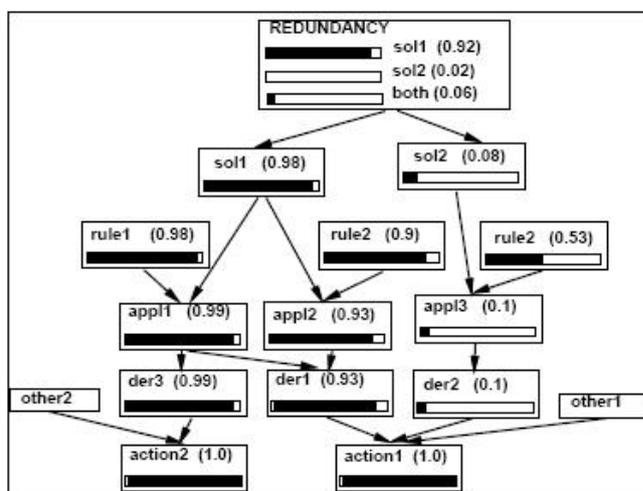


Figura 3.1: Red POLA con nodos solución y redundancia [13]

Con esta nueva red, se consigue dar más importancia a las acciones realizadas por el alumno porque cuando tiene lugar una acción del alumno, cuando la regla es conocida y el conjunto de soluciones a las que la aplicación per-

tenece es probable, entonces se deduce que la regla ha sido aplicada con eficacia. Con esto queda representada la intención del alumno de aplicar una regla que pertenece a un conjunto específico de soluciones. Dicho con otras palabras, se inferirá que una regla es probable que la haya elegido el alumno, solamente si es parte de una solución altamente probable o si existe evidencia explícita de que se haya aplicado esa regla.

El conocimiento del proceso llevado a cabo por el alumno en la resolución del problema es procesado por el sistema para proporcionar ayudas adecuadas al alumno, y también pistas y realimentación oportunas.

La ventaja principal de la red POLA es que es dinámica, en el sentido de que se construye con cada nueva acción que realiza el alumno, y no es necesario guardar información de las inferencias realizadas en el pasado y de las que pudieran ocurrir en el futuro. Una consideración así en una red estática implicaría una representación explícita del tiempo en la red.

Estos dos sistemas, OLAE y POLA, además de aplicarse al área de física, pueden ser aplicados a otros dominios que se caracterizan por tener un espacio de soluciones muy grande para los problemas, como es el caso del álgebra, la geometría y la programación. La ventaja de estos sistemas es que no limitan el conjunto de soluciones a un problema, sino que se consideran todas las soluciones que el alumno da al problema.

Una evolución del sistema POLA viene con el sistema ANDES [14], que utiliza un sistema de colaboración entre el alumno y el tutor dando como resultado un aprendizaje colaborativo. Esta colaboración consiste en que por cada paso que da el alumno, el tutor le expresa su conformidad. Del mismo modo, cuando el alumno duda en un paso, el tutor le ayuda suministrándole las pistas necesarias.

La diferencia de ANDES con respecto a POLA está en que mientras que POLA proporciona diagnóstico en función de las acciones realizadas por el alumno, ANDES calcula las posibles inferencias generadas por el alumno pero que no ha expresado explícitamente en su solución y aquéllas que pueden contener errores.

El modelado del alumno en ANDES se basa también en una red bayesiana, con concreto, la red de POLA, pero con la adición de todos los que se consideran los “planes abstractos” para la resolución de un problema, que son los pasos que explican cómo se llega a las soluciones. La red resultante se denomina “red de solución de problema”.

Otro autor posterior que trata las redes bayesianas dinámicas es Reye [45], según el cual, en redes bayesianas ordinarias, se asume que las propiedades modeladas por la red no cambian, a pesar de que la red reciba información de entrada relacionada con ellas. Esto es útil, por ejemplo, en un entorno médico, porque la causa de una enfermedad no cambian durante un examen

médico.

Este enfoque estático es inadecuado para el modelado del estudiante en un sistema tutorial, que Reye [45] justifica con dos razones:

- La evolución dinámica del conocimiento del estudiante a lo largo de un período en el tiempo en el que se recolecta información del estudiante.
- Las características de incertidumbre asociadas con las acciones del estudiante.

Las redes bayesianas dinámicas contienen nodos con un estado que dura un período de tiempo. En los sistemas tutoriales, es más útil que los estados representen las interacciones del alumno con el sistema [47].

De forma general, se intuye que una interacción del alumno con el sistema producirá que cambie su estado de conocimiento. En la realidad, con cada interacción, son necesarias dos fases para actualizar el modelo del estudiante:

- Una fase de recolección de evidencia, que consiste en incorporar la evidencia (si la hay) al conocimiento previo a la interacción del estudiante.
- Una fase de actualización de los cambios esperados en el estado de conocimiento del estudiante como resultado de la interacción.

La red bayesiana de Reye [45] se muestra en la siguiente figura:

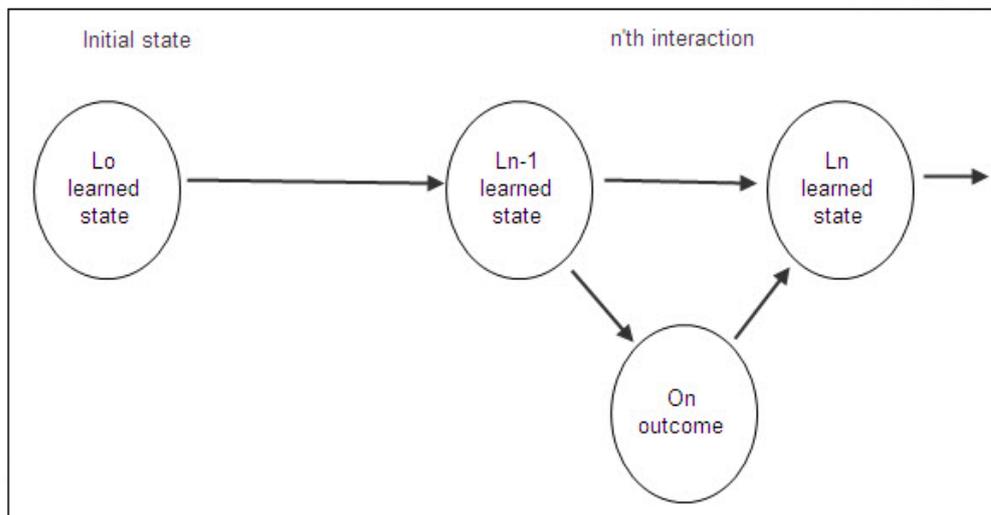


Figura 3.2: Red bayesiana dinámica en cada interacción del alumno con el sistema [45]

En la figura 3.2 tenemos:

- $O_n$  es un elemento del conjunto de *posibles salidas* de una interacción con el sistema, por ejemplo, el conjunto de todas las respuestas correctas/incorrectas del estudiante sobre un tópico del dominio.
- $p(L_{n-1})$  es la probabilidad de que el estudiante conoce un tópico del dominio dado antes de la interacción con el sistema, con  $n = 1, 2, \dots$
- $p(O_n|L_{n-1})$  es la probabilidad de que la salida  $O_n$  tendrá lugar cuando el estudiante ya conoce el tópico del dominio.
- $p(O_n|\neg L_{n-1})$  es la probabilidad de que la salida  $O_n$  tendrá lugar cuando el estudiante no conoce el tópico del dominio.

Las últimas dos probabilidades son las probabilidades condicionadas que definen el enlace entre  $L_{n-1}$  y  $O_n$ .

Para calcular el conocimiento en  $L_{n-1}$  dada la salida  $O_n$ , el sistema aplica el teorema de Bayes del siguiente modo:

$$p(L_{n-1}|O_n) = \frac{p(O_n|L_{n-1})p(L_{n-1})}{p(O_n|L_{n-1})p(L_{n-1}) + p(O_n|\neg L_{n-1})p(\neg L_{n-1})} \quad (3.5)$$

Con el cálculo de esta probabilidad estamos condicionando el resultado al valor  $p(L_{n-1})$  que se tenía antes de la evidencia. Es por ello, que es necesario realizar la fase 2 de actualización de las probabilidades, que consiste en calcular  $p(L_n|O_n)$ . Es decir, es preciso añadir dos enlaces desde el nodo  $L_{n-1}$  y  $O_n$  al nodo  $L_n$ . Estos dos enlaces requieren para su definición las siguientes dos probabilidades:

- $p(L_n|L_{n-1}, O_n)$ . Es la probabilidad de que el estudiante permanezca en el estado anterior después de la salida. Es una tasa de recuerdo. Tendrá el valor máximo 1 cuando la salida  $O_n$  sea correcta.
- $p(L_n|\neg L_{n-1}, O_n)$ . Es la probabilidad de que el estudiante haga la transición desde el estado en el que no sabe nada hasta el estado de conocimiento. Es una tasa de aprendizaje.

La probabilidad que nos interesa  $p(L_n|O_n)$  vendrá dada por la expresión:

$$p(L_n|O_n) = p(L_n|L_{n-1}, O_n)p(L_{n-1}|O_n) + p(L_n|\neg L_{n-1}, O_n)p(\neg L_{n-1}|O_n) \quad (3.6)$$

En esta fórmula, se simplifica la notación definiendo:

$$\rho(O_n) = p(L_n|L_{n-1}, O_n) \quad (3.7)$$

y

$$\lambda(O_n) = p(L_n|\neg L_{n-1}, O_n) \quad (3.8)$$

La ecuación 3.6 queda así simplificada:

$$p(L_n|O_n) = \lambda(O_n) + [\rho(O_n) - \lambda(O_n)]p(L_{n-1}|O_n) \quad (3.9)$$

Al combinar las ecuaciones 3.5 y 3.11 tenemos:

$$p(L_n|O_n) = \lambda(O_n) + \frac{[\rho(O_n) - \lambda(O_n)]\gamma(O_n)p(L_{n-1})}{1 + [\gamma(O_n) - 1]p(L_{n-1})} \quad (3.10)$$

que puede reescribirse como:

$$p(L_n|O_n) = \frac{\lambda(O_n) + [\rho(O_n)\gamma(O_n) - \lambda(O_n)]p(L_{n-1})}{1 + [\gamma(O_n) - 1]p(L_{n-1})} \quad (3.11)$$

En esta ecuación, cuando  $p(L_{n-1}) = 1$ , se cumple que  $p(L_n|O_n) = \rho(O_n)$ . En este caso, se considera que la probabilidad de conocer el concepto cuando ya lo conocía es igual a la tasa de recuerdo.

También sucede que cuando  $p(L_{n-1}) = 0$ , se cumple que  $p(L_n|O_n) = \lambda(O_n)$ . En este caso, se considera que la probabilidad de conocer el concepto cuando el alumno no lo conocía es igual a la tasa de aprendizaje.

Un ejemplo de red bayesiana adaptativa dinámica que es el tutor ACT (Programming Languages Tutor) [15].

El tutor ACT es un sistema tutor para el aprendizaje del lenguaje LISP que pretende presentar un nuevo método de diagnóstico del alumno. Está basado en reglas, derivando en acciones del sistema. El objetivo de ACT es diagnosticar el nivel de conocimiento, modelando el conocimiento del alumno de las reglas y modelar los cambios en el nivel de conocimiento del alumno con el tiempo.

El modelo de aprendizaje del sistema asigna a cada regla uno de dos posibles estados: aprendida y no aprendida. Una regla puede pasar del estado “no aprendida” al estado “aprendida” pero no a la inversa. Como se contempla la situación de que el estudiante ante una regla aprendida pueda fallarla y también adivinar una regla no aprendida, el sistema estima la probabilidad de que el alumno haya aprendido cada regla. Cada vez que el alumno aplica una regla del sistema, se actualiza la probabilidad de que el alumno conozca la regla, dependiendo de que la haya aplicado correctamente o no.

El modelo de aprendizaje utilizado es una red bayesiana que contiene dos parámetros de aprendizaje y dos parámetros de rendimiento. Las probabilidades empleadas son:

- $p(L_0)$ . Es la probabilidad de que una regla esté en el estado “aprendida” previamente a su aplicación. Es la probabilidad a priori de la regla.
- $p(T)$ . Es la probabilidad de que una regla pase del estado “no aprendida” al estado “aprendida” después de aplicarse.
- $p(G)$ . Es la probabilidad de que el estudiante adivine correctamente una regla si está en el estado “no aprendida”.
- $p(S)$ . Es la probabilidad de que el estudiante cometa un error si una regla está en el estado “aprendida”.

De esta manera, la probabilidad de que un estudiante realice una acción correcta con respecto a un objetivo  $g$ , es la suma de dos productos:

- La probabilidad de que una regla adecuada  $r$  esté en el estado “aprendida” por la probabilidad de que una respuesta sea correcta estando en el estado “aprendida”.
- La probabilidad de que una regla adecuada  $r$  no esté en el estado “aprendida” por la probabilidad de que una respuesta se adivine estando en el estado “no aprendida”.

Las probabilidades calculadas son la probabilidad de que cada regla esté en el estado “aprendida” después de que el alumno proporciona una respuesta correcta ( $C_n$ ) o incorrecta ( $E_n$ ), en la  $n$ -ésima oportunidad.

Según Reye [45], las ecuaciones que permiten el cálculo de estas probabilidades se pueden derivar de su red dinámica argumentando que aunque los autores Corbett y Anderson [15] no utilizan el concepto de red dinámica, sí emplean teoría de probabilidades que, por tanto, puede ser comparada con las probabilidades asociadas con una red bayesiana dinámica.

Reye [45] hace una serie de suposiciones sobre su propio modelo para poder deducir las del tutor ACT, que son más simples. Las restricciones supuestas son:

- El conjunto de posibles salidas tiene sólo dos valores:  $C_n$  (respuesta correcta) y  $E_n$  (respuesta incorrecta).
- $p(L_n | L_{n-1}, O_n) = \rho(O_n)$ , que es la tasa de recuerdo.

- $p(L_n | \neg L_{n-1}, O_n) = \lambda(O_n)$ , que es la tasa de aprendizaje. Esta tasa es una constante, lo que significa que la probabilidad de que el estudiante haga una transición desde el estado “no aprendido” hacia el estado “aprendido” es independiente de la salida o respuesta del usuario. En ACT, se usa el símbolo  $p(T)$  para esta tasa.
- No hay probabilidades condicionadas entre las distintas reglas.

Considerando estas restricciones, la ecuación 3.11 se puede reescribir del siguiente modo:

$$p(L_n | O_n) = p(T) + [1 - p(T)]p(L_{n-1} | O_n) \quad (3.12)$$

que resolviendo paréntesis, queda:

$$p(L_n | O_n) = p(L_{n-1} | O_n) + p(T)(1 - p(L_{n-1} | O_n)) \quad (3.13)$$

De esta ecuación, se deducen otras dos, sustituyendo  $O_n$  por sus dos posibles valores,  $C_n$  y  $E_n$  que son:

$$p(L_n | C_n) = p(L_{n-1} | C_n) + p(T)(1 - p(L_{n-1} | C_n)) \quad (3.14)$$

$$p(L_n | E_n) = p(L_{n-1} | E_n) + p(T)(1 - p(L_{n-1} | E_n)) \quad (3.15)$$

que son las ecuaciones que deducen los autores.

Del mismo modo, las probabilidades estáticas calculadas en la primera fase de Reye [45], con la nueva notación, tendrán la expresión:

$$p(L_{n-1} | C_n) = \frac{p(C_n | L_{n-1})p(L_{n-1})}{p(C_n | L_{n-1})p(L_{n-1}) + p(\neg L_{n-1})p(C_n | \neg L_{n-1})} \quad (3.16)$$

que es la misma expresión usada en ACT para calcular  $p(L_{n-1} | C_n)$  que es:

$$p(L_{n-1} | C_n) = \frac{p(L_{n-1}) * p(C|L)}{p(L_{n-1})p(C|L) + p(U_{n-1})p(C|U)} \quad (3.17)$$

Del mismo modo, el tutor ACT calcula la probabilidad en el caso de que el alumno haya respondido incorrectamente con la expresión:

$$p(L_{n-1} | E_n) = \frac{p(L_{n-1}) * p(E|L)}{p(L_{n-1})p(E|L) + p(U_{n-1})p(E|U)} \quad (3.18)$$

En estas dos ecuaciones 3.17 y 3.18, los autores definen:

- $p(L_{n-1})$ . Es la probabilidad de que una regla esté en el estado “aprendida” tras la  $n$ -ésima oportunidad de aplicar la regla.
- $p(L_{n-1})$ . Es la probabilidad de que una regla esté en el estado “aprendida” previamente a la  $n$ -ésima oportunidad de aplicar la regla.
- $p(U_{n-1})$ . Es la probabilidad de que una regla esté en el estado “no aprendida” previamente a la  $n$ -ésima oportunidad de aplicar la regla.
- $p(C|L)$ . Es la probabilidad de que el alumno responda correctamente si la regla está en el estado “aprendida”.
- $p(E|L)$ . Es la probabilidad de que el alumno se equivoque si la regla está en el estado “aprendida”.
- $p(C|U)$ . Es la probabilidad de que el alumno adivine la respuesta si la regla está en el estado “no aprendida”.
- $p(E|U)$ . Es la probabilidad de que el alumno se equivoque si la regla está en el estado “no aprendida”.
- $p(T)$ . Es la probabilidad de que una regla pase del estado “no aprendida” al estado “aprendida” dada la oportunidad de aplicar la regla.

La probabilidad dinámica de la segunda fase de Reye [45] viene dada por las siguientes dos ecuaciones:

$$p(L_n|C_n) = p(L_{n-1}|C_n) + [1 - p(L_{n-1}|C_n)] * p(T) \quad (3.19)$$

$$p(L_n|E_n) = p(L_{n-1}|E_n) + [1 - p(L_{n-1}|E_n)] * p(T) \quad (3.20)$$

Para comprobar la validez del modelo, el sistema calcula la tasa de error que comete cada alumno en cada objetivo de aprendizaje para después obtener la correlación de las tasas entre los distintos objetivos.

Otro ejemplo de red dinámica es la del sistema SMART [47], que utiliza un número de funciones para actualizar el modelo del estudiante.

En el contexto del modelo del estudiante con redes bayesianas y los tests adaptativos está la red bayesiana de preguntas y conceptos de Millán y Pérez de la Cruz [38]. Este modelo bayesiano lo denominamos en el presente trabajo como “modelo de red bayesiana de Millán” y lo describimos ampliamente en la sección 4.4.1.

### 3.4.5. Comparativa entre formas de modelado

Desde mi punto de vista, las restricciones son una forma estándar para reflejar el conocimiento de un sistema. Esta forma de modelado permite representar las relaciones entre los hechos de un problema de manera sencilla. Las reglas después pueden codificarse en forma de condiciones para la toma posterior de decisiones. El primer inconveniente que se ve a priori es lo que sucede cuando el número de reglas a manejar es ingente, lo que podemos tener en cualquier sistema donde el conocimiento es amplio independientemente del dominio de aplicación.

En cuanto a las inferencias de hipótesis y predicciones de las causas que dan como resultado una hipótesis, hemos visto en la sección 3.4.1 cómo un modelo basado en reglas lleva a cabo una serie de operaciones sobre los factores de certeza asignados a las reglas que se pueden considerar sencillas.

No obstante, a pesar de la facilidad de operación con los sistemas de reglas, Heckerman [23] ya demostró las incoherencias que podemos tener en un sistema de este tipo. Hay dos tipos de inferencia de información según Heckerman que son diagnóstica y predictiva. La inferencia diagnóstica está relacionada con la deducción de la hipótesis dada primeramente la causa o evidencia. Y la inferencia predictiva con la inferencia de la causa o evidencia dado el efecto o hipótesis. La situación anómala que puede darse se produce cuando tenemos reglas de diagnóstico y de predicción mezcladas en el mismo sistema porque no se combinan de manera adecuada teniendo factores de certeza que no son coherentes. Sucede también que cuando varias hipótesis comparten distintas causas, la creencia en la hipótesis final disminuye al tener evidencia procedente de más de una de las causas, lo cual no es lógico desde del punto de vista del razonamiento humano. Heckerman [23] lo enuncia diciendo que debido al principio de separación (*detachment*) en la combinación de los factores de creencia, la creencia final está sobrevalorada cuando las causas están positivamente correlacionadas y está subestimada cuando la correlación entre las causas es negativa.

Además de estas anomalías de los sistemas de reglas, debido a la forma de interpretar las reglas de izquierda a derecha, es decir, dada una evidencia, deducir posteriormente una hipótesis, en los dominios sobre todo médicos esta forma de razonar no es la más cercana al mundo real ya que en medicina se suele deducir a partir de una enfermedad cuáles son sus síntomas.

En cuanto al modelado basado en redes, esta representación es más rica porque es capaz de realizar operaciones más complejas sobre el dominio de forma más directa y sencilla a como se haría con el modelado basado en reglas. Las relaciones entre los nodos en un sentido y en otro llevan implícitas los dos tipos de operaciones de abducción y de predicción.

En este tipo de representaciones en forma de red, hemos visto cómo se puede tener un modelo de factores de certeza y lógica difusa en el tratamiento de la incertidumbre del alumno, con contraposición con los valores de probabilidad que utilizan las redes bayesianas.

Si comparamos un sistema de reglas y de lógica difusa con probabilidades, tenemos que el paso de unos sistemas a otros es más o menos directo y que aunque los factores de certeza y la lógica difusa se aproximan en sus valores a los de las probabilidades, no acaban de tener la misma precisión. Es cierto que el cálculo de los factores en un modelo de factores de certeza y en la lógica difusa son más sencillos, pero en muchas ocasiones interesa la complejidad del teorema de Bayes para tener resultados mejores.

Asimismo, en los sistemas de redes bayesianas, el cálculo de las probabilidades con el Teorema de Bayes, dan lugar a resultados más coherentes, a pesar de las suposiciones de independencia que se hacen sobre las variables del modelo.

Tras esta comparativa de modelos, en el presente trabajo optamos por el modelado con redes bayesianas debido a su mayor validez para la representación de la incertidumbre y la existencia de trabajos desarrollados en el modelado del estudiante con redes bayesianas que presentan resultados muy aceptables y que describiremos en la sección 4.4.1.

## Capítulo 4

# ESTADO DEL ARTE DE LOS TESTS ADAPTATIVOS

### 4.1. Introducción

En este capítulo, hacemos un estudio de las formas de modelado empleadas hasta el momento en el modelado de los tests adaptativos informatizados. Se trata de ver los modelos de estudiante presentes en los tests adaptativos informatizados que permiten que cada pregunta de un test sea adaptada al nivel de conocimiento del alumno.

### 4.2. Tests adaptativos modelados con la Teoría de Respuesta por Item

Una de las teorías más sólidas utilizadas como fundamento en los tests adaptativos informatizados es la Teoría de Respuesta por Item [22].

La Teoría de Respuesta por Item estima el nivel de rasgo del usuario del test mediante la definición de cada ítem haciéndolo depender de la variable  $\Theta$ , que representa el nivel de rasgo.

Esta teoría asume que usando como entrada un conjunto de respuestas de los alumnos a un conjunto de preguntas, el nivel de conocimiento del alumno es estimado con algún método estadístico. Estos pasos se repiten hasta que se cumple un criterio de parada. Dependiendo de los métodos estadísticos utilizados para estimar  $\Theta$  y para seleccionar la siguiente pregunta del test, tendremos diferentes modelos de TRI [22].

Además de los modelos TRI que hacen depender el rendimiento de cada ítem con el nivel de conocimiento  $\Theta$ , denominados “unidimensionales”, tam-

## 4.2. TESTS ADAPTATIVOS MODELADOS CON LA TEORIA DE RESPUESTA POR ITEM

---

bién existen modelos TRI que dependen de uno o más rasgos denominados “multidimensionales”. Por otra parte, si el modo de cuantificación de las preguntas es dicotómico, el modelo es dicotómico y en el caso de considerarse más de dos categorías de respuesta, el modelo es politómico [42].

Vamos a centrar nuestra revisión en los modelos unidimensionales dicotómicos, debido a que son los utilizados por la mayoría de los TAI existentes en la actualidad.

Todos los modelos TRI tienen algunas características comunes:

- Suponen la existencia de aptitudes en el alumno.
- La relación entre la aptitud y las respuestas de una persona a los ítems del test se puede describir con una función exponencial monótona creciente llamada Curva Característica del Ítem (CCI).

Hay tres tipos de modelos TRI dependiendo del tipo de parámetros que intervienen en la fórmula que define la probabilidad de respuesta correcta. Hambleton y Rogers [22] comentan los tres modelos:

- Modelo logístico de un parámetro. Este modelo se le conoce también como modelo Rasch. Los autores Hambleton y Rogers [22] recomiendan Rasch (1960) y Wright y Stone (1917) para ampliar información acerca del modelo Rasch.
- Modelo logístico de dos parámetros. Este modelo fue deducido según los autores por Lord (1952) a partir de una función que describía una distribución normal. Según los autores, esta evolución desde un modelo normal a un modelo logístico fue ventajosa porque el modelo logístico dependiente de dos parámetros no necesita operaciones como la integración y además presenta propiedades estadísticas.
- Modelo logístico de tres parámetros. Este modelo es el más completo porque cada ítem queda mejor caracterizado.

Gráficamente, el funcionamiento de cualquier ítem descrito por un modelo dicotómico unidimensional puede representarse mediante la *curva característica del ítem* (CCI), que pone en relación los niveles  $\Theta$  con sus correspondientes  $P(\Theta)$  [42].

Olea y Ponsoda [42] nos presentan el modelo logístico de un parámetro (1P) o modelo de Rasch como el modelo unidimensional dicotómico más sencillo que tiene la expresión:

$$p(\Theta) = \frac{\exp D(\Theta - b_j)}{1 + \exp D(\Theta - b_j)} \quad (4.1)$$

En este modelo  $P(\Theta)$  es la probabilidad de acertar el ítem  $j$  cuando la persona tiene un nivel de rasgo  $\Theta$ . El parámetro  $\Theta$  normalmente asume valores entre  $-4$  y  $+4$ ,  $b_j$  es el parámetro de dificultad del ítem (normalmente asume valores entre  $-4$  y  $+4$ , pues se mide en la misma escala que  $\Theta$ ),  $D$  es un valor constante (si  $D=1$  se habla de escala logística; si  $D=1.7$ , de escala normal), y  $e$  es la base de los logaritmos naturales ( $e=2.718$ ).

Con el modelo 1P se está asumiendo en realidad que el rendimiento de un ítem depende únicamente del nivel de rasgo del alumno y de la dificultad del ítem.

Olea y Ponsoda [42] listan una serie de inconvenientes del modelo 1P:

- Si el nivel de rasgo  $\Theta$  coincide con la dificultad del ítem, la probabilidad de acertarlo es igual a 0.5.
- Si el nivel de rasgo  $\Theta$  es muy pequeño, la probabilidad de acierto se aproxima a cierto. Es decir, este modelo considera que no se producen aciertos por azar.
- La pendiente que tiene la CCI es la misma independientemente del valor de  $b$  que hayamos asignado a cada ítem. Esto significa que todos los ítems tienen la misma discriminación o capacidad para diagnosticar el nivel de rasgo del usuario.
- La dependencia del rasgo  $\Theta$  únicamente de la dificultad del ítem se muestra gráficamente en que la curva CCI esté desplazada hacia la izquierda ( $b_j$  pequeño) o a la derecha ( $b_j$  alto).

Una ventaja destacada del modelo 1P es que a la hora de calibrar los ítems o lo que es lo mismo, decidir los valores de los parámetros, se necesita un banco pequeño de ítems a diferencia de los modelos 2P y 3P.

El modelo logístico de dos parámetros (2P) incorpora a la función logística un parámetro de discriminación del ítem ( $a_j$ ) que tiene la expresión:

$$p(\Theta) = \frac{\exp Da_j(\Theta - b_j)}{1 + \exp Da_j(\Theta - b_j)} \quad (4.2)$$

donde  $a_j$  es proporcional a la pendiente de la CCI en el valor  $\Theta = b_j$ . Este parámetro de discriminación, que suele oscilar entre 0 y 3, indica el grado en que el ítem discrimina entre los niveles  $\Theta$  superiores e inferiores a la dificultad del ítem.

Los inconvenientes del modelo 2P son:

- Si la pendiente de la curva CCI es baja ( $a_j$  es pequeño), el ítem resulta poco discriminativo. Por el contrario, si la pendiente es alta, el ítem

## 4.2. TESTS ADAPTATIVOS MODELADOS CON LA TEORIA DE RESPUESTA POR ITEM

---

sirve para diferenciar los niveles de rasgo por encima y por debajo de su dificultad.

- Este modelo tampoco considera la probabilidad de acertar por azar.

El modelo logístico de tres parámetros (3P) añade al modelo 2P un nuevo parámetro ( $c_j$ ) denominado factor de azar que puede oscilar aproximadamente entre 0 y 0.5 y que representa la probabilidad de acertar el ítem cuando el nivel  $\Theta$  es muy bajo. Su expresión matemática es:

$$p(\Theta) = c_j + (1 - c_j) \frac{\exp Da_j(\Theta - b_j)}{1 + \exp Da_j(\Theta - b_j)} \quad (4.3)$$

Hambleton y Rogers [22] definen las propiedades fundamentales que cumplen los tres modelos logísticos:

La primera propiedad es la que se conoce como independencia local, que se enuncia de la siguiente manera: una vez que se define el conjunto de habilidades de los alumnos que se someten a un test, las respuestas a los ítems de los alumnos son estadísticamente independientes. Además de la independencia local, está la *propiedad de invarianza*, que según los autores es la piedra angular que la distingue de la teoría clásica. La *invarianza* implica que los parámetros que caracterizan a un ítem no dependen de la distribución de habilidad de los examinados y el parámetro que caracteriza a un examinado no depende tampoco del conjunto de ítems del test. Gráficamente, esta propiedad supone que las curvas CCI correspondientes a diferentes distribuciones de habilidad de un conjunto de alumnos son equivalentes. Hay que destacar que valores distintos de los parámetros de la CCI da lugar a formas diferentes en las curvas. A continuación, se muestran las curvas CCI con 4 niveles de habilidad distintos en un modelo logístico 3paramétrico:

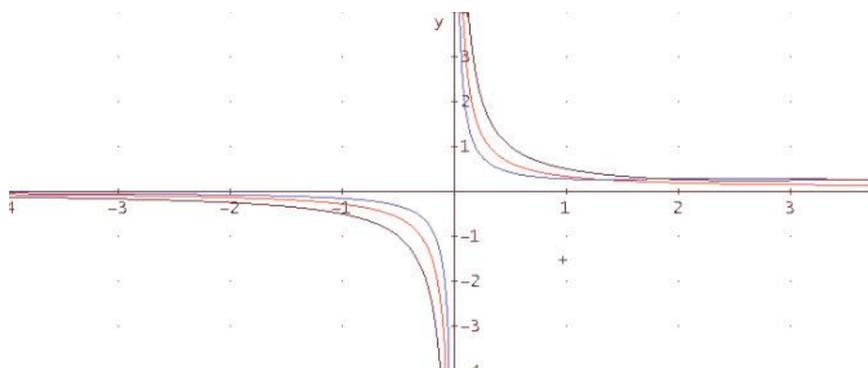


Figura 4.1: Curvas CCI de 3 parámetros para 3 ítems realizadas con ELVIRA [18]

Para poder emplear un modelo TRI, hay una serie de tareas que hay que resolver previamente [42]:

- Ajuste al modelo.
- Estimación del nivel de rasgo.
- Calibración de los ítems.

**1. Ajuste al modelo** El ajuste al modelo TRI elegido a los datos experimentales con los que contamos, es una ventaja de la teoría TRI frente a la clásica TCT. Como primera medida de ajuste, se comprueba el grado de unidimensionalidad que tiene un test para lo que según los autores en [42] se pueden aplicar métodos de análisis factorial. También se puede comprobar la varianza de los parámetros.

Olea y Ponsoda en [42] se introducen los modelos multidimensionales y politómicos, los cuales son una ampliación de los modelos unidimensionales.

**1. Estimación del nivel de rasgo** Para la estimación del nivel de rasgo, pueden aplicarse varios métodos. El más utilizado es la *función de máxima verosimilitud condicionada*, dado que consiste en buscar el valor  $\Theta$  asociado al valor máximo de la función de verosimilitud, tomados los parámetros de los ítems como conocidos. La función de verosimilitud tiene la siguiente expresión:

$$L(u|\Theta) = \prod_{(1,n) \in j} P_j^{u_j} Q_j^{1-u_j} \quad (4.4)$$

donde  $u$  es el vector de respuestas ( $u_i=1$  cuando se ha acertado la respuesta y  $u_i=0$  cuando no se ha acertado) a los ítems,  $P_j$  es la probabilidad de acertar el ítem  $j$  dado un nivel de rasgo  $\Theta$  y  $Q_j$  es la probabilidad de fallar el ítem.

La ecuación 4.5 demuestra que, dado un determinado patrón de respuestas a los ítems que forman el test, el valor  $L$  asociado a un nivel de rasgo  $\Theta$  se obtiene multiplicando las correspondientes probabilidades de acierto y error obtenidas mediante el modelo TRI que se haya escogido.

En el problema de la estimación del nivel de rasgo, hay que destacar que para tener buenos resultados (con mayor precisión), se necesita un gran número de ítems y un tamaño de población de usuarios para disponer de muchos valores del nivel de rasgo  $\Theta$ ). El tamaño de las muestras necesarias será mayor cuanto mayor es también el número de parámetros a estimar.

Para estimar el nivel de rasgo y los parámetros del modelo TRI, será necesario maximizar la función de verosimilitud. La primera forma de hacerlo

## 4.2. TESTS ADAPTATIVOS MODELADOS CON LA TEORIA DE RESPUESTA POR ITEM

---

es derivando la función  $L$  con respecto a  $\Theta$  y a los parámetros obteniendo un conjunto de ecuaciones e igualándolas a 0 para obtener las soluciones del sistema. Como se apunta en [41], el problema no tiene solución analítica y se recurre a procedimientos numéricos para encontrar sus soluciones. Otra alternativa es maximizar la función logaritmo neperiano de  $L$  puesto que es más sencilla y las soluciones en ambos casos son equivalentes.

Este método de máxima verosimilitud tiene el inconveniente de que no es capaz de calcular el valor máximo de  $L$  cuando los patrones en la respuesta del usuario son constantes, por ejemplo, acertar todas las preguntas o fallarlas todas.

Para solucionar este problema, están los *procedimientos de estimación bayesianos* que consisten en incorporar a la función de verosimilitud información sobre la distribución a priori del rasgo en la población. Así, por ejemplo, podemos tener procedimientos que establecen que el estimador de  $\Theta$  será la moda (estimación bayesiana MAP) o la media (estimación bayesiana EAP) de la distribución de probabilidad posterior de  $\Theta$ , cuya expresión es:

$$P(\Theta|u) = g(\Theta)L(u|\Theta) \quad (4.5)$$

donde  $g(\Theta)$  es la función de densidad de  $\Theta$  y  $L(u|\Theta)$  es la función de verosimilitud.

La estimación de los parámetros en TRI es asunto complejo. Los procedimientos de estimación vistos pueden combinarse con tres estrategias distintas y dan lugar a los métodos de estimación conjunta, marginal o condicionada. En el primer caso, se estiman simultáneamente los parámetros de los ítems y de los niveles de rasgo. En el segundo, a partir de la distribución conjunta de los ítems y de los niveles de rasgo, se obtiene la distribución marginal de los estimadores de los parámetros de los ítems, y a partir de esta información se obtienen los parámetros de las personas. En los métodos condicionados, los parámetros de los ítems se obtienen a partir de su distribución condicionada, en vez de la marginal.

En cuanto a la precisión en la estimación de los parámetros, se utilizan medidas sencillas como son el *error típico de medida* ( $S_e$ ) que se calcula bajo la suposición de que la distribución que sigue la variable  $\Theta$  es normal con media  $\mu$  y varianza igual a:

$$\sigma^2 = \frac{1}{\sum_{(1,n) \in i} \frac{P_i^2}{P_i Q_i}} \quad (4.6)$$

El denominador de la expresión se denomina *información del test* o información de Fisher y que puede simbolizarse por  $I(\Theta)$ .

$$S_e = \frac{1}{\sqrt{I(\Theta)}} \quad (4.7)$$

A partir de la expresión anterior, donde el valor de la función  $I$  se sustituirá por el valor adecuado para el modelo 1P, 2P o 3P, puede deducirse que la información de un test, y por tanto el error típico de medida, para un determinado nivel de rasgo  $\Theta$ , depende fundamentalmente de: los parámetros de discriminación de los ítems (cuanto mayores sean los valores de  $a$ , mayor será el valor de la información), del parámetro de azar  $c$  (cuanto más bajos sean los valores de  $c$ , mayor será el valor de la información), de la cercanía entre los valores del parámetro de dificultad  $b$  y el nivel de rasgo  $\Theta$  (cuanto más próximos sean los valores, mayor es el producto de  $P_i Q_i$  y por último, del número de ítems del test porque suponemos que a mayor número de ítems tenemos más información.

En el caso de la estimación bayesiana MAP, la varianza del estimador es el inverso de la función de información, que en este caso tiene la expresión:

$$S_e = \frac{1}{\sigma^2} + I(\Theta) \quad (4.8)$$

Vemos que cuantitativamente es mayor esta varianza que en el error típico de medida de la estimación de la máxima verosimilitud, por lo que el método bayesiano arroja estimaciones más precisas.

**1. Calibración de los ítems** La calibración de los ítems consiste en estimar sus parámetros la primera vez antes de aplicarse. También es necesario estimar los parámetros  $\Theta$  de los usuarios. Una vez que el test está calibrado, el problema estadístico de los modelos TRI se reduce a estimar los niveles de rasgo de los usuarios.

A la hora de elaborar un banco de ítems, puede resultar un proceso muy costoso en función del número de ítems del banco y también de las características que deben cumplir los ítems en función de su tipo. Aunque lo más usual es utilizar un formato de respuesta de elección múltiple [42], también es posible otro tipo de ítems más elaborados, lo cual dificulta y ralentiza su construcción.

Una posibilidad para facilitar la tarea, es la generación automática de ítems, que consiste en construir el banco de ítems mediante un algoritmo. Según Olea y Ponsoda en [42], es un método del cual no se ha comprobado aún su validez. No obstante, los autores nos recomiendan programas para tal efecto como son la versión para Windows del programa BILOG que emplea el método de máxima verosimilitud marginal y los métodos bayesianos EAP y MAP para estimar los parámetros de los tres modelos logísticos. Están

también los programas ASCAL (modelos 2P y 3P) y RASCAL (modelo 1P) que utilizan conjuntamente los métodos de máxima verosimilitud y el procedimiento MAP. Además, está el programa MULTILog que permite estimar los parámetros de los tres modelos logísticos y de la mayoría de los modelos politómicos. Por último, también hay programas para estimar el modelo 1P como WINSTEPS y FACETS.

#### 4.2.1. Modelo logístico TRI de un parámetro

A pesar de los inconvenientes que hemos comentado del modelo 1P, hay que destacar que el número de parámetros que son necesarios estimar en este modelo es mucho menor que en los modelos 2P ( $2 \times$  número de ítems) y modelo 3P ( $3 \times$  número de ítems).

Olea y Ponsoda [42] indican que bastantes de los TAIs operativos que más se usan como el CAT-GRE o pruebas adaptativas para la certificación de la profesión de especialistas médicos se basan en un modelo 1P.

#### 4.2.2. Modelo logístico TRI de dos parámetros

Hambleton y Rogers [22] informan que además de los modelos logísticos de tres parámetros se han desarrollado otros modelos TRI. Los autores nos citan como ejemplo el de Bock (1972), que desarrolló un modelo logístico de dos parámetros que se puede aplicar a todas las respuestas en un ítem de respuesta múltiple.

El propósito del *modelo de respuesta nominal* de Bock era maximizar la precisión de la estimación del nivel de habilidad utilizando para ello toda la información contenida en las respuestas de los examinados, no sólo si el ítem era o no respondido correctamente.

Bock define los parámetros  $a$  y  $b$  para cada opción de respuesta de un ítem de respuesta múltiple. De este modo, se calcula la probabilidad de responder correctamente cada una de las opciones, cumpliéndose que la suma de las probabilidades de las opciones es igual a uno. En mi opinión, esta solución es muy informativa sobre cada ítem, pero demasiado farragosa si tenemos que especificar los parámetros para cada una de las posibles respuestas de cada ítem.

A priori, Bock no establece un orden entre las opciones, lo que sí realiza Samejina (1969) según los autores estableciendo un orden entre las opciones con un escala numérica.

Un ejemplo de TAI basado en el modelo 2P que citan Olae y Ponsoda [42] es el sistema MATHCAT para la evaluación de conocimientos matemáticos de personas adultas. Este TAI se acomoda muy bien al modelo 2P porque

dispone de un procedimiento de respuesta (teclear una solución numérica) que hace mínima la probabilidad de acierto aleatorio.

A continuación, vamos a revisar el sistema SIETTE, que se basa en el modelo logístico de tres parámetros y que hemos considerado un sistema eficaz y completo que modela perfectamente los sistemas de tests adaptativos informatizados.

### 4.2.3. Modelo logístico TRI de 3 parámetros. El sistema SIETTE

SIETTE es un sistema de tests adaptativos basado en el modelo 3paramétrico de la teoría TRI. Es una herramienta web desarrollada para apoyar a alumnos y profesores en el proceso de evaluación [11]. Forma parte del proyecto TREE (Training of European Environment Trainers and Technicians), fundado por el programa europeo Leonardo cuyo propósito principal es la clasificación e identificación de diferentes especies vegetales con un sistema inteligente.

Los tests son generados en función de las especificaciones del profesor y son adaptativos, es decir, las preguntas se seleccionan de modo que se correspondan con el nivel de conocimiento del alumno en cada momento.

Los autores afirman que una de las ventajas de SIETTE es que se obtienen estimaciones más precisas del nivel de conocimiento del alumno con tests significativamente más cortos.

El sistema SIETTE cuenta con el uso de applets de Java que pueden incluir tanto preguntas de respuesta múltiple como preguntas con respuesta abierta. En la presente sección, nos centraremos en el uso de SIETTE aplicado al ámbito de las preguntas de respuesta múltiple puesto que, desde mi punto de vista, constituyen los tests adaptativos más fáciles de evaluar en el conjunto de un test y de calibrar los parámetros IRT. Por ejemplo, para el parámetro  $c$  se le puede asignar el valor  $1/n$  siendo  $n$  el número de respuestas posibles del ítem.

#### Arquitectura de SIETTE

Los módulos de los que se compone el proyecto TREE [11] son:

- Un sistema experto (ES), que se utiliza para clasificar e identificar especies.
- Un sistema tutorial inteligente (ITS), para ayudar a los alumnos en el proceso de aprendizaje del dominio.

## 4.2. TESTS ADAPTATIVOS MODELADOS CON LA TEORIA DE RESPUESTA POR ITEM

---

- Un sistema de generación de tests (TGS), para mostrar a los alumnos como apoyo si han alcanzado un adecuado nivel de conocimiento.
- Una base de conocimiento del dominio (KB) que utilizan los tres módulos anteriores y que contiene información sobre el dominio botánico. Esta base es actualizada continuamente por expertos vía web.

El sistema SIETTE se puede emplear de dos modos:

- El profesor o experto del dominio puede usar SIETTE para desarrollar los tests, es decir, definir los tópicos, las preguntas, los parámetros y especificaciones.
- Los alumnos pueden usar SIETTE para pasar los tests.

En el “modelo bayesiano propuesto” del presente trabajo, nos centraremos en la segunda funcionalidad de SIETTE y dedicaremos el estudio a los resultados de los alumnos en los tests.

Vamos a ver los módulos más importantes de SIETTE.

### **El modelo del estudiante en SIETTE**

El modelo del estudiante es un modelo temporal, que SIETTE crea y actualiza para cada estudiante que pasa el test [11]. El modelo de estudiante se compone de un vector de  $k$  probabilidades  $(p_0, p_1, \dots, p_{k-1})$ , donde  $p_i$  representa la probabilidad de que el alumno alcance el nivel de conocimiento  $i$  sobre el dominio siendo 0 el nivel más bajo y  $k - 1$  el nivel experto.

El módulo generador de tests es el módulo principal de SIETTE que selecciona la siguiente pregunta a ser presentada en el test y que dependerá del modelo temporal del estudiante y de las especificaciones del test.

### **Base de conocimiento en SIETTE**

En SIETTE, los tests tienen una estructura basada en el currículum. Cada test de una determinada materia se estructura en tópicos y preguntas. La estructura permite que una pregunta pueda pertenecer a varios tópicos y un tópico a diferentes tests.

SIETTE no hace uso intensivo de esta estructura en el sentido de que al terminar el test, el sistema devuelve una estimación del nivel de conocimiento del alumno, pero no una respuesta relacionada con los tópicos del test [11].

### Diseño de tests en SIETTE

En SIETTE, en el diseño de preguntas de un test, hay que especificar los siguientes parámetros:

- El título y número de posibles respuestas.
- El texto de la pregunta, que debería ser simple y nunca ambiguo.

Como hemos comentado en la sección 4.2.3, SIETTE se basa en el modelo TRI 3paramétrico, o más concretamente cada ítem de un test SIETTE, va a estar modelado por una función CCI de 3 parámetros  $a$ ,  $b$  y  $c$ . En esta función, el nivel de conocimiento del alumno se mide con una variable aleatoria discreta  $\Theta$  que toma valores enteros en  $[0, k - 1]$  en vez de la aleatoria continua que se usa en la teoría TRI.

Esta afirmación significa que una CCI en SIETTE es un conjunto  $p_i$  de  $k$  valores, que se corresponden con la probabilidad de responder correctamente a una pregunta dado que el nivel de habilidad del alumno es  $i$ . Por ejemplo, si  $U$  es la variable aleatoria que representa la respuesta del alumno al ítem, el conjunto CCI estaría formado por  $p_i = p(U_i = 1/\Theta = i); i = 1 \dots k - 1$ .

En esta función de distribución, hay que especificar los valores de los tres parámetros  $a$ ,  $b$  y  $c$ .

El factor  $c$  es determinado por SIETTE en función del número de diferentes respuestas que se muestran al alumno  $c = 1/n$ . Los valores de  $a$  deben ser un número entero entre 0,5 y 1,5. El parámetro  $b$  puede tomar valores entre 0 y  $k - 1$ .

### El módulo generador de tests.

En SIETTE, el algoritmo que genera los tests se basa en el algoritmo de cualquier sistema TAI y consiste de los siguientes procedimientos:

1. Selección de los ítems.
2. Estimación del nivel de conocimiento.
3. Criterio de terminación.

Vamos a ver cada uno de ellos.

**1. Selección de los ítems.** SIETTE suministra tres posibles procedimientos para la selección de los ítems:

- Un procedimiento bayesiano, que selecciona el ítem que minimiza la posterior desviación estándar.
- Un procedimiento basado en la dificultad, que selecciona el ítem que minimiza la distancia entre la media de la CCI y la media del nivel de habilidad actual del alumno.
- Un procedimiento aleatorio, que selecciona el ítem aleatoriamente.

Además de usar estos procedimientos, el sistema se amplía con las siguientes posibilidades:

- Selección aleatoria de ítems. Si hay un empate al elegir entre dos preguntas, el ítem se selecciona aleatoriamente.
- Equilibrio de contenido. SIETTE utiliza los pesos que se especifican para cada uno de los tópicos en el diseño de los tests. Los pesos determinan el porcentaje de preguntas de cada tópico. SIETTE compara los porcentajes de las preguntas que ya han sido presentadas con los porcentajes deseables, y selecciona el tópico que tenga una mayor diferencia entre ambos porcentajes.
- Tests longitudinales. SIETTE evita que las mismas preguntas de un test se le muestren al alumno si ya ha realizado el test con anterioridad. Para conseguirlo, se consulta la información del modelo del estudiante sobre los ítems ya pasados en tests previos. Una de las características del sistema es la capacidad de proporcionar realimentación inmediata. Mientras un estudiante está pasando el test, el sistema le puede mostrar la solución a la pregunta que acaba de contestar.

**2. Estimación del nivel de conocimiento** Una vez que el alumno ha respondido una pregunta, SIETTE calcula el nuevo nivel de habilidad del alumno actualizado en base a sus respuesta. Para calcular la probabilidad, SIETTE aplica el Teorema de Bayes, que calcula la probabilidad de que el alumno responda correctamente dado un nivel de habilidad  $\Theta = k$ .

El Teorema de Bayes necesita las probabilidades a priori, que en este caso son todos los valores de la función distribución de probabilidad  $p(\Theta = k)$ , con  $k = 0 \dots k - 1$ , que se asume que es uniforme porque  $p(\Theta = k) = 1/k$  y las probabilidades condicionadas calculadas por la CCI,  $p(U_i = 1/\Theta = k)$ , con  $i = 0 \dots k - 1$ .

Y con estas probabilidades Bayes calcula la probabilidad a posteriori de que el alumno tenga un nivel de conocimiento  $\Theta$  dada la respuesta al ítem:

$$p(\Theta = k|U_i) = \frac{p(U_i = k)p(= k)}{\sum_{(0,k-1) \in i} p(U_i = i)p(= i)} \quad (4.9)$$

que se puede interpretar como que la probabilidad de que el alumno tenga un nivel de habilidad  $k$  en función de la respuesta  $U_i$  es el cociente entre la probabilidad de CCI dado que el alumno tiene un nivel de habilidad  $k$  dividido entre la suma de las probabilidades de CCI para los casos en que el alumno tiene todos los niveles de habilidad anteriores a  $k$  (desde 0 hasta  $k-1$ ). Dicho en otras palabras, es la probabilidad de que el alumno responda bien el ítem de dificultad  $k$  del conjunto total de ítems.

El nivel de habilidad del alumno  $\Theta$  se puede obtener con la función de máxima verosimilitud [8] que tiene la siguiente expresión:

$$L(u|\Theta) = L(u_1, \dots, u_n|\Theta) = \prod_{(1,n) \in i} p_i(\Theta)^{u_i} (1 - p_i(\Theta))^{(1-u_i)} \quad (4.10)$$

donde  $u = (u_1, \dots, u_n)$  es el vector de respuestas del alumno y cada  $u_i$  puede tomar valor 1 si el alumno ha respondido correctamente el ítem  $i$  y 0 si no ha respondido correctamente.

Por ejemplo, para un test de tres ítems, un estudiante responde bien el primer ítem ( $u_1 = 1$ ), incorrectamente el segundo, ( $u_2 = 0$ ) y correctamente el tercero ( $u_3 = 1$ ). Su habilidad se obtendría maximizando:

$$P_1^1 Q_1^0 \times P_2^0 Q_2^1 \times P_3^1 Q_3^0 = P_1 \times Q_2 \times P_3 \quad (4.11)$$

Basta con encontrar los valores de  $\Theta$  para los cuales la ecuación es máxima.

**3. Criterio de terminación** SIETTE también da posibilidades para configurar este criterio que son:

- La desviación estándar de la distribución del nivel de conocimiento del alumno es menor que un valor fijado (es decir, la estimación es lo bastante precisa).
- La probabilidad de que el conocimiento del alumno sea mayor (menor) o igual a un nivel de conocimiento sea a su vez mayor que un valor umbral fijado.
- El sistema ha presentado todas las preguntas del test.

- El sistema ha mostrado al menos el mínimo número de preguntas de cada tópico especificado en el test.

### 4.3. Modelos de regla de inferencia para tests adaptativos

Hay sistemas hipermedia de tests adaptativos que utilizan este tipo de reglas, como es el caso de AHA [1]. En AHA, el modelado del estudiante está formado por conceptos y atributos.

Este modelo está influido por la estructura de los conceptos y las reglas de adaptación. La estructura de los conceptos en AHA contiene los conceptos y los atributos que contienen los conceptos.

La adaptación en AHA es a través de los contenidos. En concreto, se muestran páginas con ítems de test en función de los cambios que se producen en el modelo del estudiante. El sistema proporciona una herramienta de autoría llamada *graph author tool* para la definición de los conceptos y las reglas. Un ejemplo de regla adaptativa es la siguiente:

```
tutorial.enduser.knowledge >50 and tutorial.concepts.knowledge <40
```

En la regla anterior, se deduce que hay un objeto llamado *tutorial* que representa a un test, una variable *enduser* que representa al usuario que está realizando el test y una variable *knowledge* que significa el nivel de conocimiento inicial del usuario. El objeto *concepts* representa el conjunto total de conceptos del sistema conocidos por el usuario. Por tanto, la regla anterior codifica la condición de que el conocimiento inicial del alumno en el sistema sea mayor del 50 % y el total de conceptos aprendidos sea menor del 40 %.

Una peculiaridad de AHA es que distingue entre tipos de conceptos, pudiendo definirse conceptos asociados a páginas y también conceptos abstractos que no tienen página asociada. Los tipos de sistemas de aprendizaje web que permite diseñar las arquitectura AHA son los sistemas de test de respuesta múltiple.

Según el tutorial de AHA, hay que seguir los siguientes pasos para el diseño de un test con opciones de respuesta múltiple:

1. Definición de los conceptos.
2. Definición de un conjunto de ítems para un test. La creación del ítem va asociada con la elección de un concepto que evaluará ese ítem.
3. Definición de un conjunto de respuestas del ítem.

4. Añadir respuestas al ítem.
5. Definir parámetros de la Item Response Theory como son la discriminación y dificultad del ítem.

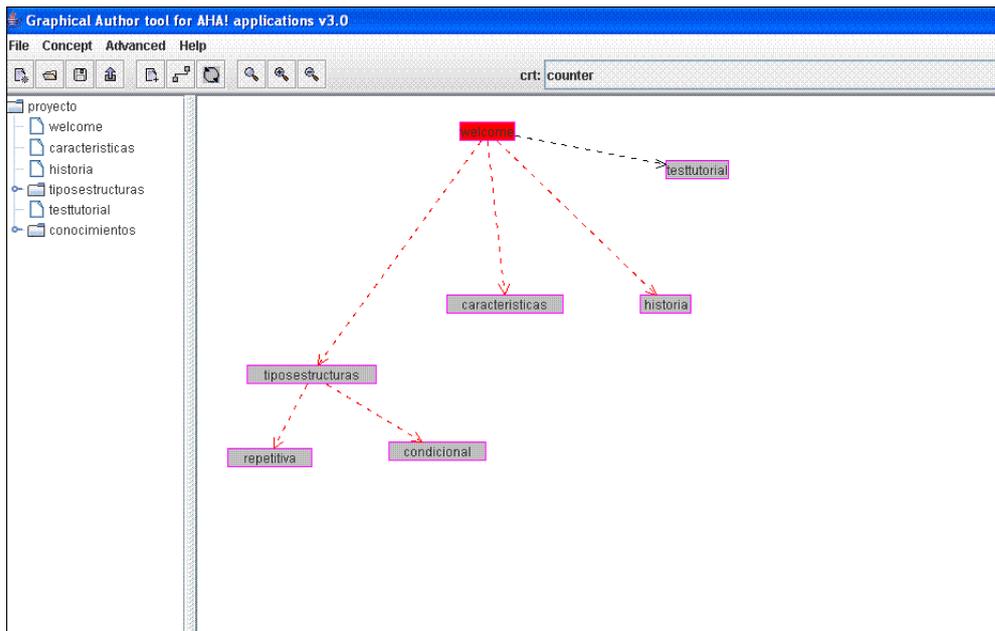


Figura 4.2: Graph Author Tool de un curso adaptativo AHA [1]

En mi opinión, las herramientas AHA son un método fácil e intuitivo para diseñar un test por parte de un profesor aunque obliga a definir una red de conceptos y sus relaciones. Las reglas adaptativas las calcula el sistema AHA automáticamente. Deduzco que para un test de pocas preguntas, la red de conceptos puede ser más o menos sencilla de diseñar pero para el caso tener muchas preguntas y tener muchos conceptos, el diseño de la red se complica.

## 4.4. Modelos de red bayesiana para tests adaptativos

### 4.4.1. Redes bayesianas con modelo respuesta del ítem

Millán en [37] define una red bayesiana para modelar los conceptos de una materia y las preguntas de un test adaptativo sobre la materia.

El modelo estructural de la red de Millán [37] se compone de dos tipos de nodos:

#### 4.4. MODELOS DE RED BAYESIANA PARA TESTS ADAPTATIVOS

---

- Nodos concepto
- Nodos evidencia

Los nodos concepto representan cada concepto indivisible en que dividimos la materia de estudio. Estos nodos son binarios, es decir, pueden tomar valores 0 y 1 solamente. Cada nodo concepto hace referencia, por tanto, a una variable concepto, que denotamos por  $C$  y que tiene una distribución de Bernouilli, según la cual  $C$  tomará el valor 1 cuando el alumno conoce ese concepto y valor 0, cuando el alumno no lo conoce.

La distribución de probabilidad de la variable  $C$  se define como:

$$p(C = x) = p^x(1 - p)^{1-x} \quad (4.12)$$

Los nodos evidencia recolectan información sobre el estado de conocimiento del alumno. Millán [37] los clasifica en dos tipos:

- Preguntas tipo test multirespuesta. Son preguntas tipo test en las que el alumno tiene que elegir entre varias posibles respuestas, y en las que sólo una de ellas es correcta. Los nodos de tipo pregunta también son binarias y las variables  $P$  a las que representan siguen una distribución de Bernouilli ya que tomarán valor 1, cuando el alumno seleccione la respuesta correcta y valor 0, en caso contrario.

La distribución de probabilidad de la variable  $P$  es:

$$p(P = x) = p^x(1 - p)^{1-x} \quad (4.13)$$

- Tareas, problemas o ejercicios que se le asignan al alumno para que los resuelva. Se utilizará una variable aleatoria  $E$  que tomará el valor 0 si el ejercicio se resuelve incorrectamente y el valor 1 si se resuelve correctamente.

La distribución de probabilidad de la variable  $E$  es:

$$p(E = x) = p^x(1 - p)^{1-x} \quad (4.14)$$

donde  $p$  es la probabilidad de que el alumno resuelva correctamente el ejercicio  $E$ , y  $x$  toma los valores 0 y 1.

Hay dos tipos de relaciones entre los nodos de la red de Millán que son: relaciones de agregación y relaciones causales.

- Relaciones de agregación. Son las relaciones que se dan entre los nodos concepto y otros nodos que modelan también el conocimiento del alumno, que se corresponden con los temas y asignaturas. Las relaciones unen los nodos concepto con los nodos tema y los nodo tema con los nodos asignatura. Son relaciones de agregación porque el conocimiento de un nodo tema supone el conocimiento del alumno en todos los conceptos de los que se compone y lo mismo sucede con los nodos tema y asignatura. La distribución de probabilidad de una variable es la suma de las variables de las que consta.
- Relaciones causales. Son las relaciones que se dan entre los nodos concepto y los nodos evidencia, que acabamos de comentar que pueden ser generalmente de dos tipos: nodos ejercicio y nodos de pregunta multi-respuesta. Las relaciones causales son distintas dependiendo de si el nodo evidencia es un ejercicio o una pregunta multirespuesta.

En el “modelo bayesiano propuesto” del capítulo 6 no consideramos las relaciones de agregación al obviar el modelado de los nodos tema y asignatura por una sencilla razón: nuestro interés es determinar el grado de conocimiento que el alumno adquiere en los distintos conceptos, y en base a estos conceptos asignarle un nivel de conocimiento, que puede ser un simple valor numérico. No obstante, me parece una solución muy eficiente y aplicable en los casos en que el profesor necesita evaluar al alumno por temas.

El modelado de relaciones entre nodos de conocimiento y nodos de evidencia se puede hacer con un modelo estático y un modelo dinámico. La red bayesiana es la misma en ambos casos y la diferencia fundamental está en que como afirma Reye [45], la red dinámica permite que los nodos evidencia puedan modificar dinámicamente la información del resto de la red.

El modelo estructural de la red bayesiana de Millán es el siguiente:

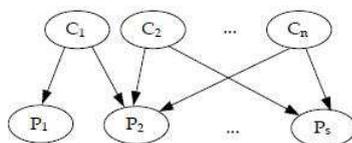


Figura 4.3: Relaciones entre conceptos y preguntas en Millán [37]

Dadas las relaciones de dependencia causal del modelo estructural, los parámetros a especificar en esta red son las “probabilidades a priori” de los conceptos y las “probabilidades condicionadas” de las preguntas dados los conceptos.

#### 4.4. MODELOS DE RED BAYESIANA PARA TESTS ADAPTATIVOS

Las probabilidades condicionadas de las preguntas dados los conceptos se modelan con el modelo matemático IRT 3-parámetro, que es una función exponencial monotónica creciente también denominada Curva Característica del Ítem (CCI).

La CCI del ítem  $i$  indica la probabilidad de que el alumno responda correctamente ( $u_i = 1$ ) supuesto que  $x = \Theta$  y tiene la expresión:

$$p_i(x) = p(u_i|\theta = x) = c_i + \frac{1 - c_i}{1 + \exp(-1,7a_i(x - b_i))} \quad (4.15)$$

donde  $a_i$  es el índice de discriminación del ítem,  $b_i$  es el grado de dificultad del ítem y  $c_i$  es el factor de adivinanza o azar.

En el modelo de Millán, que considera los conceptos como unidades del conocimiento  $\theta$ , redefine la ecuación 4.15 en base a los conceptos del sistema.

Por cada pregunta del sistema, si está relacionada con  $p$  conceptos, hay un total de  $2^p$  probabilidades condicionadas. En su cálculo intervienen 4 parámetros:

- $c$ : es el factor de azar, que representa la probabilidad de adivina la respuesta correcta. Su valor es de  $1/n$ , donde  $n$  es el número de respuestas posibles.
- $a$ : es el factor de discriminación, que según SIETTE, se puede fijar en un valor comprendido entre 0.5 y 1.5.
- $b$ : nivel de dificultad del ítem.
- $s$ : factor de descuido, que representa la probabilidad de fallar el ítem aún conocidos todos los conceptos que en ella intervienen.

Millán aplica una transformación lineal a la ecuación 4.15 para trabajar finalmente con la función  $G(x)$ :

$$G(x) = 1 - \frac{(1 - c)(1 + \exp(-1,7ab))}{1 + \exp(-1,7a_i(x - b_i))} \quad (4.16)$$

En esta función  $G(x)$  la variable  $x$  queda redefinida por el número de conceptos que el alumno conoce.  $G(x)$  tiene como valor mínimo para  $x = 0$  el valor  $c$ . De este modo, se puede contemplar que la probabilidad de responder correctamente la pregunta cuando el número de conceptos conocidos es 0 es el factor de azar  $c$ . Además el valor máximo de esta función es  $1 - s$  para así considerar la situación de que aunque el alumno conozca todos los conceptos, puede fallar la pregunta con un factor de descuido  $s$ .

Las probabilidades condicionadas tienen por el momento los siguientes valores:

- $G(0) = c$ , que indica que en caso de que no se conozca ningún concepto, el alumno puede responder a la pregunta y adivinarla con un factor de azar  $c$ , que es  $1/n$ , donde  $n$  es el número de opciones de respuesta posible.
- $G(x^*) = 1 - s$ , que indica que en caso de que se conozcan todos los conceptos de la pregunta, es posible fallarla con un factor de descuido  $s$ .
- El resto de probabilidades Millán las calcula con la función  $G(x)$  del siguiente modo: Se calcula un factor  $k$  por cada probabilidad, que permite ordenar las probabilidades en función de la importancia del concepto o conceptos implicados en la pregunta.

Este factor  $k$  se multiplica por el valor  $x^*$  asociado con el valor máximo de  $G(x)$  y después se divide el resultado por el número de probabilidades totales  $2^p - 1$  de la pregunta, para así tener un total de valores de probabilidad (hasta  $2^p - 1$ ) y que cada valor de probabilidad esté interpolado entre  $c$  y  $G(x^*)$ .

Cada probabilidad condicionada contiene una combinación de posibles estados de los conceptos de una pregunta de modo que además del estado del concepto (“dominado” o “no dominado”) se tiene en cuenta la importancia de cada concepto en la pregunta. Para asignar este orden, Millán simplemente ordena los conceptos por orden de importancia formando una palabra binaria con los estados 1 (dominado) y 0 (no dominado) con los conceptos de menor importancia en la parte más baja de la palabra binaria. Cada palabra binaria obtenida se corresponde entonces con un valor de probabilidad condicionada. Sin considerar la palabra binaria formada por todo ceros (caso de no conocer el alumno ningún concepto), para el resto de palabras binarias, el valor de  $k$  se asigna desde 1 hasta  $2^p - 1$ , ambos valores incluidos.

Para ilustrar bien el cálculo de las palabras binarias y la asignación posterior de un valor de probabilidad condicionada  $G(x)$  a cada palabra binaria, vamos a verlo con un ejemplo.

**Ejemplo 1.**

Sean  $C1, C2, C3$  y  $C4$ , los conceptos necesarios para responder la pregunta  $P$ . Supongamos que los conceptos están ordenados de acuerdo a su importancia en la pregunta, de manera que  $C1$  es el concepto más importante y  $C4$  el menos importante. En este ejemplo, los valores de los parámetros IRT son:  $c = 0,25$ ,  $s = 0,01$ ,  $b = 5$  y  $a = 0,3$ . Lo primero que se calcula es  $x^*$  tal que  $G(x^*) = 1 - s = 0,99$ . Se obtiene para  $x^* = 13,716$ . Para 4 conceptos, hay un total de  $2^4 = 16$  combinaciones binarias. Las probabilidades condicionadas

#### 4.4. MODELOS DE RED BAYESIANA PARA TESTS ADAPTATIVOS

asignadas a cada una de las combinaciones se calculan evaluando  $x$  en  $G(x)$  con la expresión:  $kx/15$ , para  $k = 0, \dots, 15$ .

C1	C2	C3	C4	x	$G(x) = p(P = 1)$
0	0	0	0	0	0.25
0	0	0	1	0.914	0.233
0	0	1	0	1.829	0.280
0	1	0	0	2.743	0.345
1	0	0	0	3.658	0.427
0	0	1	1	4.572	0.522
0	1	0	1	5.487	0.622
0	1	1	0	6.401	0.717
1	0	0	1	7.316	0.797
1	0	1	0	8.230	0.861
1	1	0	0	9.145	0.907
0	1	1	1	10.059	0.939
1	0	1	1	10.973	0.961
1	1	0	1	11.888	0.975
1	1	1	0	12.802	0.984
1	1	1	1	13.717	0.99

Tabla 4.1: Tabla de probabilidades condicionadas de la red bayesiana de Millán

Las probabilidades se asignan de forma creciente como se muestra en la siguiente figura:

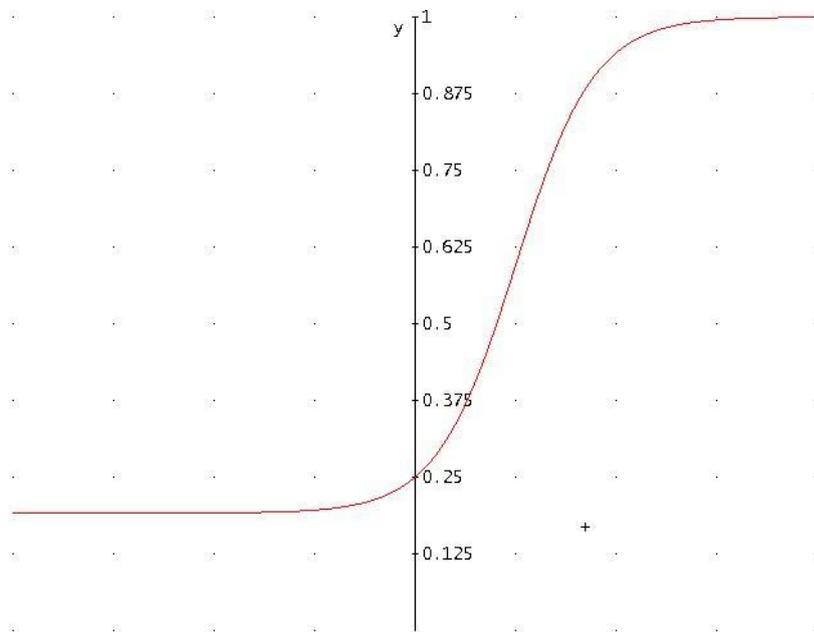


Figura 4.4: Valores de la función  $G(x)$

En la figura 4.4 puede apreciarse cómo el valor mínimo de la función es el valor de  $c = 0,25$  y el valor máximo es de  $0,99$  que coincide con el valor  $1 - s$  para  $s = 0,01$  que habíamos fijado inicialmente. De este modo, quedan representadas en la función los aciertos de un alumno por el factor de azar  $c$  y la posibilidad de fallo con el factor de descuido  $s$ .

## 4.5. Conclusiones

Tras hacer un estudio del estado del arte de los tests adaptativos existentes hasta el momento, se ha decidido que el modelo del estudiante del “modelo de red bayesiana con pesos propuesto” del capítulo 6 que se presenta en este trabajo, sea una red bayesiana de preguntas y conceptos. Asimismo, dada la efectividad comentada y sabida probada del sistema SIETTE, nos ha parecido que cada nodo pregunta de la red bayesiana puede modelarse con una CCI de 3 parámetros, siendo solamente necesario especificar para cada ítem, el valor de los parámetros  $a$ ,  $b$  y  $c$ .

De esta manera, el “modelo bayesiano propuesto” aprovecha las características de una red bayesiana en cuanto a su capacidad de representar las relaciones entre sus nodos pregunta y concepto y además la teoría TRI nos permite calibrar y seleccionar un ítem del test que sea conforme al nivel de habilidad actual del alumno, característica fundamental de los tests adapta-

tivos.

Antes de comentar las consideraciones que nos han llevado a utilizar las redes bayesianas para el modelado del estudiante que pueden encontrarse en la sección 5.4, presentamos las características básicas de las redes bayesianas en el siguiente capítulo.

# Capítulo 5

## LAS REDES BAYESIANAS

### 5.1. Introducción

En este capítulo, presentamos en primer lugar en la sección, unas nociones básicas que son necesarias conocer para la comprensión del funcionamiento de las redes bayesianas, y en la sección, los conceptos de las redes que hemos aplicado en el modelo bayesiano centro de nuestro trabajo, que hemos renombrado al principio del trabajo como “modelo de red bayesiana con pesos propuesto”.

### 5.2. Nociones básicas

Tal y como se introduce en [19], dados un conjunto de datos observados, las redes bayesianas capturarán las relaciones de dependencia existentes entre los atributos de los datos observados. Las redes bayesianas describen la distribución de probabilidad que gobierna un conjunto de variables.

Existen distintas aplicaciones de interés para las redes bayesianas: clasificación, predicción, diagnóstico, etc. Entre las características que poseen las redes bayesianas se puede destacar que permiten aprender relaciones de dependencia y causalidad, combinar conocimiento con datos y manejar bases de datos incompletas.

Vamos a ver una definición formal matemática de red bayesiana.

#### 5.2.1. Teoría de grafos

Las redes bayesianas son un tipo especial de grafo, por lo que antes de definir formalmente una red bayesiana, vamos a ver unos conceptos sobre grafos.

### 5.2.2. Conceptos básicos

Supóngase un conjunto de objetos  $X = X_1, X_2, \dots, X_n$  que pueden relacionarse entre sí. El conjunto  $X$  puede ser representado gráficamente por una colección de nodos o vértices, asociando un nodo a cada elemento de  $X$ . Estos nodos pueden conectarse por aristas, indicando las relaciones existentes entre los mismos. Una arista entre los nodos  $X_i$  y  $X_j$  se denotará mediante  $L_{ij}$ . Se tiene también que el conjunto de todas las aristas es  $L = \{L_{ij} | X_i y X_j \text{ estn conectados}\}$ .

Por tanto, un grafo queda determinado por el conjunto de nodos  $X$ , y las relaciones entre ellos,  $L$ .

El concepto de grafo puede definirse de una forma más general. Por ejemplo, puede permitirse que dos nodos estén conectados por más de una arista o incluso que un nodo esté conectado consigo mismo. Sin embargo, como se afirma en [5], en el campo de los sistemas expertos, los grafos se utilizan para representar un conjunto de variables proposicionales (nodos), y unas relaciones de dependencia entre ellos (aristas).

Algunos conceptos adicionales sobre grafos son:

- Arista dirigida. Dado un grafo  $G = (X, L)$ , si  $L_{ij} \in L$  y  $L_{ij} \ni L$ , la arista  $L_{ij}$  entre los nodos  $X_i$  y  $X_j$  se denomina dirigida y se denota mediante  $X_i \longrightarrow X_j$ .
- Arista no dirigida. Dado un grafo  $G = (X, L)$ , si  $L_{ij} \in L$ , la arista  $L_{ij}$  se denomina no dirigida y denota mediante  $X_i \setminus X_j$  o  $X_j \setminus X_i$ .
- Grafo dirigido. Un grafo en el cual todas las aristas son dirigidas se denomina grafo dirigido.
- Grafo no dirigido. Un grafo en el cual todas las aristas son no dirigidas se denomina no dirigido.
- Conjunto adyacente. Dado un grafo  $G = (X, L)$ , y un nodo  $X_i$ , el conjunto adyacente del nodo  $X_i$  es el conjunto de nodos que son directamente alcanzables desde  $X_i$ , es decir,  $Ady(X_i) = \{X_j \in X | L_{ij} \in L\}$ . Este concepto nos permite definir un grafo como un conjunto de nodos y de los conjuntos adyacentes de cada uno de los nodos.
- Camino entre dos nodos. Un camino del nodo  $X_i$  al nodo  $X_j$  es una sucesión de nodos  $(X_{i_1}, \dots, X_{i_r})$ , comenzando en  $X_{i_1} = X_i$  y finalizando en  $X_{i_r} = X_j$ , de forma que existe una arista del nodo  $X_{i_k}$  al nodo  $X_{i_{k+1}}$ ,  $k = 1 \dots r - 1$ , es decir,  $X_{i_{k+1}} \in Ady(X_{i_k}, k = 1 \dots r - 1$

- Longitud de camino,  $r - 1$ , se define como el número de aristas que contiene.
- Camino cerrado. Un camino  $(X_{i_1}, \dots, X_{i_r})$  se dice que es cerrado si el nodo inicial coincide con el final, es decir,  $X_{i_1} = X_{i_r}$ .

### 5.2.3. Características de los grafos dirigidos

Debido al tipo de red bayesiana que vamos a utilizar para modelar el conocimiento del “modelo de red bayesiana con pesos propuesto”, donde hay una estructura clara de dependencia entre los elementos del conjunto  $X$ , deducimos que el tipo de grafo en el que se basa es dirigido. Es por ello, que vamos a citar algunas características de los grafos dirigidos, dejando los grafos no dirigidos para investigación del lector, que puede profundizar en ellos en [5].

- Padre e hijo. Cuando exista una arista dirigida,  $X_i \longrightarrow X_j$ , del nodo  $X_i$  al nodo  $X_j$ , entonces se dice que el nodo  $X_i$  es un padre del nodo  $X_j$ , y que el nodo  $X_j$  es un hijo de  $X_i$ .
- Familia de un nodo. El conjunto formado por un nodo y sus padres se denomina familia del nodo.
- Ascendientes de un nodo. Un nodo  $X_j$  se denomina ascendiente del nodo  $X_i$  si existe un camino de  $X_j$  a  $X_i$ .
- Conjunto ancestral. Un conjunto de nodos  $S$  se denomina un conjunto ancestral si contiene los ascendientes de todos sus nodos.
- Descendientes de un nodo. Un nodo  $X_j$  se denomina descendiente del nodo  $X_i$  si existe un camino de  $X_i$  a  $X_j$ .
- Ordenación. Dado un conjunto  $X = X_1, \dots, X_n$  de nodos, una ordenación,  $\alpha$ , es una biyección que asigna un número del conjunto  $1, \dots, n$  a cada nodo:  $\alpha : 1, \dots, n \longrightarrow X_1, \dots, X_n$ . Por tanto,  $\alpha_i$  denota el  $i$ -ésimo nodo de la numeración.
- Ciclo. Un ciclo es un camino cerrado en un grafo dirigido.
- Grafos cíclicos y acíclicos. Un grafo dirigido se denomina cíclico si contiene al menos un ciclo; en caso contrario, se denomina grafo dirigido acíclico. Los grafos dirigidos acíclicos son precisamente la base para construir las redes bayesianas.

- Grafos simples y poliárboles. Un árbol dirigido se denomina un árbol simple si cada nodo tiene como máximo un padre; en caso contrario, se denomina un poliárbol.

Las siguientes definiciones sirven para los grafos de tipo multinivel como se explica en el siguiente apartado. Se trata de definiciones que afectan a un tipo de representación de los grafos.

- Profundidad ascendente. La profundidad ascendente de un nodo  $X_i$  en un grafo dirigido acíclico,  $PA(X_i)$ , es la longitud máxima de los caminos del grafo que terminan en el nodo  $X_i$ .
- Profundidad descendente. La profundidad descendente de un nodo  $X_i$  en un grafo dirigido acíclico,  $PD(X_i)$ , es la longitud máxima de los caminos del grafo que comienzan en el nodo  $X_i$ .
- Niveles de profundidad de un grafo. Dado un grafo dirigido, el nivel  $k$  de profundidad ascendente,  $NA_k$ , es el subconjunto de nodos  $X_i \in X | PA(X_i) = k$ . De forma similar, el nivel  $k$  de profundidad descendente,  $ND_k$ , es el subconjunto  $X_i \in X | PD(X_i) = k$ .

#### 5.2.4. Formas de representación de un grafo

Como se explica en [5], un grafo puede ser representado de varias formas equivalentes que pongan de manifiesto en mayor o menor medida determinadas características.

Las formas de representación más comunes son:

- De forma simbólica. Un grafo se puede representar como un par  $(X, L)$ , donde  $X$  es un conjunto de variables y  $L$  es un conjunto de aristas entre pares de variables.
- De forma gráfica. Un grafo puede pintarse en un diagrama formado por un conjunto de nodos (uno por cada variable) y un conjunto de líneas o flechas (una por cada arista del conjunto  $L$ ).
- De forma numérica, utilizando matrices.

De todas estas representaciones, la que mejor explica la topología de un grafo es la gráfica, porque permite observar las distintas relaciones que existen entre las variables, pero tiene la desventaja de que para un número de nodos y de aristas muy elevado, se vuelve demasiado compleja. Por su parte, la representación numérica permite obtener características del grafo mediante la aplicación de operaciones algebraicas sobre una matriz.

Entre los dos tipos de representación de un grafo, destacan dos formas muy utilizadas como son:

- Representación circular.
- Representación multinivel.

En mi opinión, la que mejor modela una jerarquía entre padres e hijos a distintos niveles es, como su propio nombre indica, la representación multinivel. Según [5], la idea de representación multinivel es organizar los nodos en distintos niveles, o capas, de tal forma que no existan aristas entre nodos del mismo nivel y que todo nodo en un nivel esté conectado con algún nodo del nivel previo.

### 5.2.5. Teorema de Bayes

Un sistema experto basado en probabilidades es un tipo de sistema experto cuya base de conocimiento está formada por elementos de un espacio probabilístico y el motor de inferencia se basa en el marco de inferencia bayesiano a través de la estrategia conocida como razonamiento probabilístico o inferencia probabilística [5].

En un sistema experto basado en probabilidades, se asume la probabilidad como una medida intuitiva de incertidumbre, en la que la distribución de probabilidad conjunta de las variables se usa para describir las relaciones de dependencia entre ellas y se sacan conclusiones con fórmulas muy conocidas de la teoría de la probabilidad.

Vamos a definir el teorema de bayes que se aplica en los modelos probabilísticos para el cálculo de la distribución de probabilidad conjunta.

Se denota por  $p(A)$  a la probabilidad de que ocurra el suceso  $A$ . Asimismo, se denota por  $p(AB)$  a la probabilidad de que ocurran los sucesos  $A$  y  $B$ . Por último, se denota por  $p(A|B)$  a la probabilidad de que ocurra  $A$  cuando sabemos que ha ocurrido  $B$  y se la denomina probabilidad condicionada del suceso  $A$  dado el suceso  $B$ .

Se ha determinado por teoría de la probabilidad[4], que la probabilidad de ocurrencia de  $A$  y  $B$  es igual al producto de multiplicar la probabilidad de  $B$  por la de  $A$  una vez ocurrido  $B$ :

$$p(AB) = p(B)p(A) \tag{5.1}$$

Supongamos que el conjunto  $A$  lo dividimos en 2 subconjuntos  $(A_1, A_2)$ . Estos sucesos son mutuamente excluyentes y constituyen todas las posibles situaciones. Se cumple:

$$p(A_1) + p(A_2) = 1 \quad (5.2)$$

El suceso  $B$  se puede producir en cualquiera de las dos situaciones que se pueden dar en el conjunto  $A$ . Se cumple:

$$p(A_1B) = p(B)p(A_1|B) \quad (5.3)$$

Si en esta ecuación, despejamos  $p(A_1|B)$  tenemos:

$$p(A_1|B) = \frac{p(B|A_1)}{p(B)} \quad (5.4)$$

De esta ecuación,  $p(B)$  es igual a la suma de todas las ocurrencias de  $A$ :

$$p(B) = p(B|A_1) + p(B|A_2) + p(B|A_3) = \sum_{(1,3) \in i} p(B|A_i) = \sum_{(1,3) \in i} p(B|A_i)p(A_i) \quad (5.5)$$

Por tanto:

$$p(A_1|B) = \frac{p(B|A_1)}{\sum_{(1,3) \in i} p(B|A_i)} \quad (5.6)$$

Esta expresión es el Teorema de Bayes que se ha deducido a partir de un ejemplo para hacerlo más comprensible.

### 5.2.6. Definición formal de red bayesiana

Una red bayesiana [4] es un par  $(D, P)$ , donde  $D$  es un grafo dirigido acíclico,  $P = p(X_i|pa(X_i))$ , con  $i = 1, \dots, n$ , es un conjunto de  $n$  funciones de probabilidad condicionada, una para cada variable, y  $pa(X_i)$  es el conjunto de padres del nodo  $X_i$  en  $D$ . El conjunto  $P$  define una función de probabilidad asociada mediante la factorización:

$$p(x) = \prod_{(1,n) \in i} p(x_i|pa(X_i)) \quad (5.7)$$

Como se explica en [19], el grafo dirigido de las redes bayesianas está formado por nodos donde cada nodo se corresponde con una variable aleatoria y los arcos que unen los nodos son las probabilidades condicionadas que expresan las relaciones entre las variables. De este modo, un arco que vaya del nodo  $A$  al nodo  $B$  significa que  $A$  influye a  $B$ , o lo que es lo mismo  $B$  depende condicionalmente de  $A$ .

Por cada nodo de la red se escribe su tabla de probabilidad condicionada que contiene los valores de las probabilidades condicionadas de la variable dados todos los valores de sus padres en el grafo.

El conjunto de la red bayesiana será la probabilidad condicionada conjunta de todas las variables aleatorias del grafo. La probabilidad conjunta de  $n$  variables se puede expresar utilizando una tabla de tamaño  $O(2^n)$ . Debido a los factores de la forma factorizada, es suficiente con almacenar  $n$  tablas de tamaño  $O(2^k)$  donde  $k$  es el número máximo de arcos que entran en un nodo (sus padres).

Existen diversos tipos de redes bayesianas dependiendo del tipo de variables que contenga el grafo (discretas, continuas o ambas) y del tipo de distribución que se considere para cada variable. Dos tipos importantes de redes bayesianas son las redes multinomiales y las redes normales o gaussianas.

- Redes bayesianas multinomiales. En una red bayesiana multinomial, se supone que todas las variables son discretas, es decir, que cada variable puede tomar valores dentro de un conjunto finito. La función de probabilidad condicionada asociada a cada variable también es de tipo multinomial. En el apartado 5.2.7 vemos un ejemplo de red bayesiana multinomial. Estas funciones de probabilidad se describen mediante tablas con valores.
- Redes bayesianas gaussianas. En una red bayesiana gaussiana (también denominada normal) se supone que las variables del conjunto  $X$  están distribuidas por medio de una distribución normal  $N(\mu, \sigma)$  dada por la función de densidad de una distribución normal.

### 5.2.7. Construcción de redes bayesianas

En [5] se explica que la base de conocimiento de un sistema experto probabilístico está formada por un conjunto de variables y un modelo probabilístico (una función de probabilidad conjunta) que describe las relaciones entre ellas. Castillo et al., afirma en [5] que el funcionamiento de un sistema experto depende de la correcta definición de la función de probabilidad conjunta.

Para la definición del modelo, es conveniente seguir los siguientes pasos:

1. Planteamiento del problema. El primer paso en el desarrollo de un sistema experto es la definición del problema a resolver. Por ejemplo, dado que un alumno es capaz de resolver una serie de ejercicios planteados sobre una materia de estudio, el eje del problema será determinar cuáles son los conceptos de la materia que el alumno domina.

2. Selección de variables. El segundo paso consiste en la definición de variables que sean relevantes para la definición del problema. Por ejemplo, las variables relevantes en el aprendizaje de una materia de estudio, serán los conceptos de los que consta. Las variables relevantes deben seleccionarse de forma que se eviten las redundancias y son definidas por el experto.
3. Adquisición de datos. Una vez definidas las variables en el paso anterior, hay que analizar los valores que van a tomar en el modelo. Los valores pueden ser cuantitativos o cualitativos.
4. Construcción del modelo probabilístico. Una vez que están perfectamente definidas las variables y los valores que pueden tomar, el siguiente paso consiste en expresar las relaciones que se hay entre las variables mediante la definición de una función de probabilidad conjunta. Las funciones de probabilidad conjunta también están supeditadas a las relaciones de dependencia y de independencia que se den entre las variables. La estructura de la función de probabilidad depende de un número de variables o parámetros.

La parte cuantitativa de la red está formada por las siguientes probabilidades:

- a) Una tabla de probabilidad por cada una de las variables que no tienen hijos.
- b) Una tabla de probabilidad condicionada por cada variable dados los valores de sus padres.

Para la construcción del modelo, se puede utilizar un grafo para representar las relaciones entre las variables. Según Cooper and Herskovits en [16], la propiedad principal de las redes bayesianas se expresa diciendo que un nodo es condicionalmente independiente de sus nodos no descendientes, dados los valores de sus padres inmediatos.

Las condiciones de independencia entre las variables de la red bayesiana se pueden determinar con el criterio de separación gráfica separaci6nd [19].

Gyftodimos and Flach [19] enuncia el criterio de separaci6n del siguiente modo: Sean  $X, Y, Z$  tres conjuntos de v6rtices en un grafo ac6clico dirigido. Se dice que  $Z$  separa a  $X$  de  $Y$  si para cada camino indirecto entre cualquier  $x \in X$  e  $y \in Y$  existe un nodo  $w \in Z$  en ese camino. Se cumple que  $p(X|Y, Z) = p(X|Z)$ , es decir,  $X$  es independiente de  $Y$  dado  $Z$ .

Como resultado de este criterio, se podría eliminar la variable  $Y$  del grafo. Veremos en el apartado que el criterio de separación permite simplificar la estructura de una red bayesiana, y como consecuencia, el algoritmo de propagación que implementa la red.

### 5.2.8. Inferencia en redes bayesianas

#### Propagación de evidencia

Como afirma Castillo et al., en [5], la propagación de evidencia es una de las tareas más importantes de un sistema experto porque permite obtener conclusiones cuando se dispone de nueva información. Vamos a explicar el proceso de propagación de evidencia en cualquier modelo probabilístico formado por un conjunto de variables discretas  $X = X_1, X_2, \dots, X_n$ , y una función de probabilidad  $p(x)$  en  $X$ .

Cuando no se dispone de ninguna información, es decir, cuando no existe evidencia, el proceso de propagación consiste en calcular las probabilidades marginales  $p(X_i = x_i)$ , también denotadas por  $p(x_i)$ , para cada  $X_i \in X$ . Estas probabilidades se denominan “a priori” porque dan una primera información sobre los valores que pueden tomar las variables. Por contra, cuando se dispone de cierta evidencia, es decir, cuando se conoce un conjunto de variables  $E \subset X$  que tienen asociados los valores  $X_i = e_i$ , para  $X_i \in E$ , el proceso de propagación debe tener en cuenta esta nueva información.

Se define la *evidencia* como un conjunto de variables  $E \subset X$  cuyos valores son conocidos,  $E = e$ .

La propagación de evidencia consiste en calcular las funciones de probabilidad condicionada  $p(x_i|e)$  para cada variable  $X_i \ni E$ , dada la evidencia  $e$ . Las funciones de probabilidad condicionada miden el efecto producido por la evidencia en cada variable. Cuando no se dispone de evidencia ( $E = \emptyset$ ), las funciones de probabilidad condicionadas se corresponden con las probabilidades a priori  $p(x_i)$ .

Para calcular las probabilidades condicionadas, para cuando hay evidencia, se puede utilizar el teorema de Bayes del siguiente modo:

$$p(x_i, e) = \frac{p(x_i, e)}{p(e)} \quad (5.8)$$

donde  $1/p(e)$  es una constante de proporcionalidad. Por tanto, se puede obtener  $p(x_i|e)$  calculando y normalizando las probabilidades marginales  $p(x_i, e)$ . De esta forma se tiene:

$$p(x_i, e) = \sum_{(x_1, \dots, x_n) \ni e} p_e(x_1, \dots, x_n) \quad (5.9)$$

donde  $p_e(x_1, \dots, x_n)$  es la función de probabilidad obtenida sustituyendo en  $p(x_1, \dots, x_n)$  las variables con evidencia  $E$ , por sus valores  $e$ . Por consiguiente, para calcular  $p(x_i, e)$  ha de sumarse  $p_e(x_1, \dots, x_n)$  para todas las posibles combinaciones de valores de las variables que no estén contenidas en  $E$ , excepto la variable  $X_i$ . Cuando no se dispone de evidencia, esta ecuación se convierte en:

$$p(x_i) = \sum_{(x_1, \dots, x_n) \ni x} p(x_1, \dots, x_n) \quad (5.10)$$

Este método involucra un número elevado de combinaciones de valores por lo que resulta muy ineficiente, incluso en redes con un reducido número de variables. Por ejemplo, en el caso de variables binarias, hay un total de  $2^{n-1}$  probabilidades que se suman en la ecuación.

El problema de estas dos ecuaciones es que no tienen en cuenta ningún tipo de independencia entre las variables implicadas.

En el caso de las redes bayesianas que ocupan el presente trabajo, la factorización de la función de probabilidad se obtiene de forma sencilla a partir del grafo dirigido considerando un conjunto de funciones de probabilidad condicionada que involucran a cada nodo con sus padres.

Hay varios algoritmos de propagación que tienen en cuenta esta estructura de independencia entre las variables que se aplican a redes con una estructura específica.

Conviene estudiar estos algoritmos, para estudiar sus ventajas e inconvenientes y poder decidir el más adecuado para “el modelo de red bayesiana con pesos propuesto”.

El primer algoritmo que revisamos es el algoritmo de propagación de paso de mensajes, que se utiliza en redes con una estructura gráfica arbitraria.

### Algoritmo de paso de mensajes

El algoritmo de paso de mensajes se considera un mecanismo muy eficiente en el cálculo de las probabilidades de la red [5].

En el paso de mensajes, a cada nodo de la red se le asocia un “estado de certeza” que es un vector cuyos elementos suman 1 y que se corresponden con la proporción de cada valor en el dominio de valores de la variable de ese nodo.

La probabilidad de un nodo puede calcularse a partir de las probabilidades de sus padres e hijos. Siempre y cuando se produce un cambio en el valor de

la probabilidad de un nodo, directamente en él o indirectamente debido al cambio de un nodo vecino, se recalcula su valor de probabilidad y se propaga a los nodos padres e hijos. Este proceso es repetido hasta que la red alcanza un nivel de equilibrio.

La actualización de probabilidades se realiza siguiendo los pasos:

1. Actualización de la probabilidad. Consiste en el cálculo del vector de probabilidades de un nodo usando la información de los nodos vecinos.
2. Propagación bottom-up (de abajo a arriba). Consiste en calcular los mensajes  $\lambda$  que los nodos hijos envían a sus padres.
3. Propagación top-down (de arriba a abajo). Consiste en calcular los mensajes  $\pi$  que los nodos padres envían a sus hijos.

En este procedimiento de propagación de evidencia [33], se distingue entre probabilidades fijas que etiquetan los enlaces de la red, por ejemplo,  $p(y|x)$ , y los valores dinámicos de las probabilidades de los nodos actualizados, por ejemplo,  $p(x|e)$  donde  $e$  es una evidencia o valor que se asigna a una variable de la red. Esta última probabilidad, Pearl [43], la denomina  $BEL(X)$  o certeza de la variable  $X$  dada la evidencia recibida  $e$ . Por tanto:

$$BEL(x) = p(x|e) \quad (5.11)$$

Cuando trabajamos con variables discretas las funciones  $\lambda(x)$ ,  $\pi(x)$  y  $BEL(x)$  pueden considerarse listas, o vectores, donde cada componente se corresponde con un valor diferente de la variable  $X$ . Pearl [43] muestra el vector  $BEL(x)$  para una variable  $X$  cuyo dominio de valores es *Hot, Medium, Cold*:

$$BEL(x) = (BEL(X = Hot), BEL(X = Medium), BEL(X = Cold)) = (0,1, 0,2, 0,7) \quad (5.12)$$

Se define  $BEL(x)$  de un nodo como:

$$BEL(x) = \alpha\pi(x)\lambda(x) \quad (5.13)$$

donde  $\pi(x)$  y  $\lambda(x)$  son los vectores que un nodo recibe de sus nodos vecinos. Vamos a definir las variables  $\lambda$  y  $\pi$  como sigue:

$$\lambda(x) = p(e_x|x) \quad (5.14)$$

$$\pi(x) = p(x|e_{x+}) \quad (5.15)$$

$\lambda(x)$  representa el diagnóstico que el nodo  $X = x$  recibe de sus descendientes y  $\lambda(x)$  representa la predicción que el nodo  $X = x$  aporta a sus nodos descendientes.

Para calcular estas probabilidades, se aplica la generalización de la regla del Bayes así:

$$p(x|e) = \alpha * p(e|x) * p(x) \tag{5.16}$$

Para ver la información procedente de los descendientes de X que se fusiona en el nodo X, se particiona el conjunto  $e_{x-}$  en subconjuntos distintos, uno por cada hijo de X.

El proceso de actualización de un nodo X de la red puede verse en la siguiente figura:

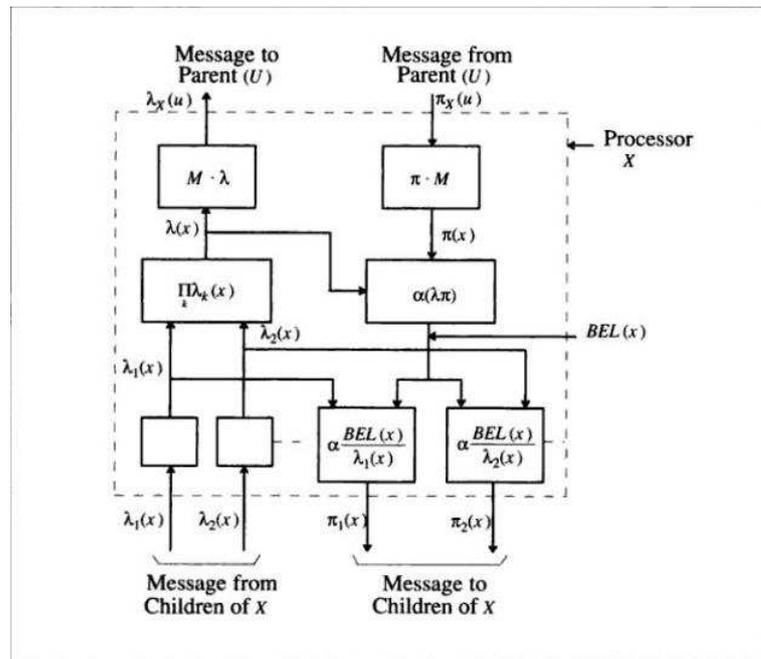


Figura 5.1: Paso de mensajes entre un nodo X y sus vecinos [43]

### Algoritmos de propagación en poliárboles

En [5], se define el *poliárbol* como uno de los modelos gráficos más simples para construir redes bayesianas.

Los algoritmos basados en poliárboles tienen la característica fundamental de que son lineales respecto al tamaño de la red (número de nodos y aristas)

que la componen) a diferencia del método de fuerza bruta que requiere un número exponencial de operaciones para realizar la propagación.

En un poliárbol dos nodos cualesquiera están unidos por un único camino, lo cual implica que cada nodo divide el poliárbol en dos poliárboles inconexos: uno que contiene a sus padres y a los nodos a los que está conectado pasando por sus padres, y otro que se compone de sus hijos y de los nodos a los que está conectado pasando por sus hijos.

El proceso de propagación puede realizarse en este tipo de grafos de un modo eficiente combinando la información procedente de los distintos subgrafos mediante el envío de mensajes de un subgrafo a otro.

Vamos a indicar las probabilidades que son necesarias calcular durante la propagación de evidencia en un poliárbol.

Suponiendo que se conoce una cierta evidencia  $E = e$  y que se quieren calcular las probabilidades  $p(x_i|e)$  para todos los valores  $x_i$  de un nodo cualquiera  $X_i$  que no esté contenido en  $E$ . Para facilitar el cálculo de estas probabilidades, gracias a la división en subgrafos, el conjunto de evidencia  $E$  se puede descomponer en dos subconjuntos distintos, cada uno de los cuales está contenido en uno de los subgrafos separados el nodo  $X_i$  en el poliárbol original. Por tanto  $E$  puede descomponerse como:

- $E_i^+$ , que es el subconjunto de  $E$  accesible desde  $X_i$  a través de sus padres.
- $E_i^-$ , que es el subconjunto de  $E$  accesible desde  $X_i$  a través de sus hijos.

Para calcular la probabilidad condicionada  $p(x_i|e)$  se aplica la regla de Bayes del modo:

$$p(x_i|e) = \frac{p(x_i, e)}{p(e)} \quad (5.17)$$

donde  $1/p(e)$  es una constante de proporcionalidad. Para el cálculo del numerador  $p(x_i, e)$ , se podría calcular y normalizar las probabilidades marginales  $p(x_i, e)$ , pero este cálculo en una red con demasiadas variables, implica un elevado número de combinaciones de variables, resultando ineficiente.

Como sucede en el algoritmo de paso de mensajes (ver 5.2.8), la eficiencia puede mejorarse aplicando el teorema de Bayes para el cálculo de la probabilidad condicionada así:

$$p(x_i|e) = \frac{1}{p(e_i^-, e_i^+)} p(e_i^-|x_i) p(e_i^+|x_i) p(x_i) = k \lambda_i(x_i) \rho_i(x_i) \quad (5.18)$$

donde  $k = \frac{1}{p(e_i^-, e_i^+)}$  es una constante de normalización.

Es decir, en la ecuación 5.18,  $\lambda_i(x_i)$  es la probabilidad  $p(e_i^-|x_i)$ , es decir, la información que un nodo recibe de sus nodos hijos y  $\rho_i(x_i)$  es la probabilidad  $p(x_i, e_i^+)$ , que es la información que un nodo envía a sus padres.

Según el algoritmo de propagación en poliárboles que consultarse en [4], las funciones  $\lambda$  y  $\rho$  de cada nodo se calculan en distintas etapas. Por tanto, si sólo interesa una variable objetivo  $X_i$ , el algoritmo puede detenerse una vez que se conozcan las funciones  $\rho_i(x_i)$  y  $\lambda_i(x_i)$  de ese nodo.

Para la implementación del algoritmo, puede distribuirse de forma paralela el cálculo en cada uno de los nodos de la red, encargados cada uno del cálculo de los mensajes  $\rho_i(x_i)$  y  $\lambda_i(x_i)$ .

La complejidad del algoritmo es lineal en el número de nodos y aristas (tamaño de la red). Por tanto, cuando una red bayesiana tiene estructura de poliárbol, el proceso de propagación puede ser llevado a cabo de forma eficiente mediante este algoritmo.

A diferencia de las redes bayesianas con una estructura arbitraria, el problema de la propagación en redes bayesianas es un problema NP-complejo [5], lo que significa que no puede resolverse en un tiempo lineal.

Vamos a analizar en la siguiente sección, los tipos de algoritmos que pueden aplicarse en estos casos de redes con estructura arbitraria.

### Algoritmos de propagación para redes múltiples

Las redes simples [2], consideradas aquéllas que sólo contienen un camino entre cada par de nodos, no suponen gran cantidad de cómputo si nos limitamos a calcular la probabilidad de cada nodo de la red aplicando el teorema de Bayes y propiedades básicas de la teoría de probabilidad. No sucede lo mismo en las redes múltiples que serán aquéllas que contienen más de un camino entre cada par de nodos. Para poder tratar con estas redes, se desarrollan algoritmos de propagación que factorizan el espacio de estados para permitir cálculos más eficientes. Básicamente, se pueden clasificar en dos tipos: algoritmos de condicionamiento y algoritmos de agrupamiento, que pueden consultarse en [5] y pasamos a describir a continuación.

1. Algoritmo de condicionamiento. Un árbol es múltiplemente conexo cuando contiene algún ciclo.

La idea básica del algoritmo de condicionamiento es cortar las vías alternativas de comunicación contenidas en los bucles asignando un valor arbitrario a un conjunto de nodos.

El problema de encontrar un conjunto de corte minimal es también un problema NP-duro, aunque hay heurísticos que permiten obtener conjuntos de corte aunque no con todos los nodos necesarios.

2. Algoritmo de agrupamiento. El método de agrupamiento se basa en la construcción de subconjuntos de nodos (aglomerados) de tal modo que el proceso de propagación de evidencia se realiza calculando probabilidades locales (que dependen de un número reducido de variables) evitando así calcular probabilidades globales (que dependen de todas las variables).

El algoritmo de agrupamiento obtiene, en primer lugar, el conjunto de conglomerados del grafo; a continuación obtiene la función de probabilidad condicionada de cada nodo marginalizando la función de probabilidad de cualquier conglomerado en el que esté contenido. Este método de agrupamiento se aplica tanto a las redes de Markov como a las redes bayesianas.

En [38], se explica un algoritmo basado en técnicas de agrupamiento que consiste en crear grupos, con cada grupo con un conjunto de nodos, de forma que la propagación de evidencia se limita a cada grupo consiguiendo con ello una disminución del tiempo computacional del algoritmo.

El algoritmo empleado en [38], se basa en el algoritmo HUGIN, que da nombre a la herramienta comercial HUGIN [25]. Dicho algoritmo consiste en hacer una serie de transformaciones sobre el grafo de partida y con los grupos obtenidos, definir las fórmulas de propagación en sentido ascendente y descendente, que serán similares a las del algoritmo de propagación de paso de mensajes.

En [32], se explica el esquema de inferencia seguido por HUGIN para la propagación de evidencia en redes con conexiones múltiples acíclicas.

En el algoritmo de HUGIN, el proceso de inferencia no se hace directamente sobre la estructura de la red, sino que el dominio se particiona en un conjunto de subdominios llamados “belief universes”, más comúnmente conocidos como “cliques”.

La tabla de probabilidad de cada nodo de la red es un conjunto de probabilidades conjuntas “no” normalizadas.

Según [32], la construcción de un sistema de “cliques”, que sea equivalente al dominio del sistema, se hace en los siguientes pasos:

- a) Construir un grafo moral. Para cada nodo de la red, se añaden enlaces entre todos sus padres y se eliminan, posteriormente, las direcciones de los arcos.

- b) Triangular el grafo moral. Un grafo es triangulado si todos los ciclos de longitud mayor de 3 están relacionados. Se añaden enlaces al grafo moral hasta que esté triangulado.
- c) Construir el sistema de cliques. Los conjuntos de nodos son los cliques de un grafo triangulado. Las tablas de probabilidad iniciales se calculan con el producto de las probabilidades condicionadas.
- d) Organizar el sistema en una estructura de árbol. El árbol se construye teniendo en cuenta que para cada par de cliques  $U$  y  $V$ , cada clique en el camino entre  $U$  y  $V$  contiene  $|C| \cap |V|$ , es decir, siempre hay un elemento común entre un clique y el siguiente.

Todos los pasos excepto el segundo dedicado a triangular el grafo son siempre iguales. A partir de un grafo moral, se determina un conjunto de cliques asociado y después se organizan en forma de árbol.

En cuanto a la propagación de evidencia, Jensen et al., en [32], describe el procedimiento seguido de la siguiente manera:

HUGIN considera que una red casual probabilística se construye sobre un universo, que es el conjunto de nodos donde cada nodo tiene un conjunto finito de estados. Los nodos se llaman variables y el universo está organizado en un grafo acíclico dirigido. El conjunto de padres de un nodo se denota por  $pa(A)$ . Cada variable tiene asociada una tabla de probabilidad condicional denotada por  $p(A|pa(A))$ .

Sea  $V$  un conjunto de variables. El espacio de  $V$  es el producto cartesiano de los conjuntos de estados de  $V$  denotados por  $Sp(V)$ . Las tablas de probabilidad condicional se denotan por las letras griegas  $\Phi$  y  $\Psi$ . Si  $A$  es una variable, entonces  $\Phi_A = p(A|pa(A))$  que mapea  $Sp(pa(A) \cup A)$  en el intervalo  $[0,1]$ . Es conveniente usar funciones que no estén normalizadas y valores arbitrarios no negativos. Estas funciones serán  $\Phi$  y  $\Psi$ .

Una tabla de probabilidad es un conjunto no normalizado de probabilidades conjuntas de un nodo. Si  $S \subset V$ , entonces el conjunto no normalizado de probabilidades conjuntas para el espacio de estados  $S_p(S)$  o todos los valores de  $S$ , puede obtenerse a partir de  $\Phi_v$  mediante una operación llamada “marginalización” que se define así:

$$\Phi_v = \sum_{(s) \ni v} \Phi_v \quad (5.19)$$

La evidencia se transmite por toda la red mediante la operación de absorción  $(U, \Phi_u)$  desde un nodo  $U$  a sus vecinos  $V$  y  $V$  a partir de las funciones  $\sum_{(s)\ni v} \Phi_v, \dots, \sum_{(u)\ni w} \Phi_w$ .

A partir de esta operación local de absorción, se definen dos operaciones de propagación denominadas `CollectEvidence` y `DistributeEvidence` que son dos métodos que existen en cada nodo de la red.

- Cuando el método `CollectEvidence` de  $V$  es invocado desde un nodo vecino  $W$ , entonces  $V$  invoca el método `CollectEvidence` en todos sus nodos vecinos (excepto el del propio  $W$ ) y  $V$  absorbe después la evidencia de los nodos vecinos.
- Cuando la función `DistributeEvidence` del nodo  $V$  es invocada por un vecino  $W$ , entonces  $V$  absorbe desde  $W$  e invoca al método `DistributeEvidence` de todos sus vecinos.

Para calcular la probabilidad a posteriori de un nodo  $U$  se aplica la siguiente fórmula:

$$\Phi'_u = \Phi'_u * \frac{\sum_{(u)\ni v} \Phi_v}{\sum_{(v)\ni u} \Phi_u} * \dots * \frac{\sum_{(u)\ni w} \Phi_w}{\sum_{(w)\ni u} \Phi_u} \quad (5.20)$$

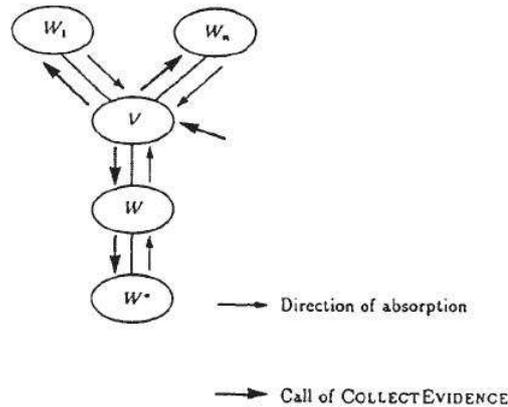


Figura 5.2: Propagación de evidencia en HUGIN [32]

En [5], se describe el algoritmo propagación de [32], de una forma más general, denominándolo algoritmo de agrupamiento para redes de Markov descomponibles. A cada clique se le llama “conglomerado” y se

define para cada clique dos conjuntos, conjunto separador y conjunto residual.

En el caso de tener una red bayesiana, no es necesario definir estos dos conjuntos aunque sí definir el conjunto de conglomerados de la red y asociando a cada nodo de la red la función de probabilidad que define la familia del nodo<sup>1</sup>.

### 5.3. Algoritmos del modelo de red bayesiana con pesos propuesto

En la sección 5.2, se han expuesto los algoritmos de propagación utilizados en las redes bayesianas con una estructura de poliárbol y también con una estructura más general.

En la presente sección, vamos a revisar los algoritmos que seleccionamos como más adecuados para la propagación de evidencia en la estructura de red bayesiana de la que se caracteriza el “modelo de red bayesiana con pesos propuesto”.

#### 5.3.1. Algoritmo de agrupamiento para redes bayesianas

Vamos a detallar los pasos del algoritmo de propagación de redes bayesianas de [4] y a demostrar con un ejemplo cómo funciona con un subconjunto de la red bayesiana del “modelo de red bayesiana con pesos propuesto” del capítulo 6.

---

<sup>1</sup>Se considera familia de un nodo al propio nodo y sus padres

---

**Algoritmo 1** Algoritmo de agrupamiento para redes bayesianas

---

**entrada:** red bayesiana  $(D, P)$  definida en un conjunto de variables  $X_i$

- 1: *Moralizargrafo* $D$
  - 2: *Triangulargrafo* $D$
  - 3: *ObtenerelconjuntoCdelosconglomeradosde* $G$
  - 4: **para**  $i \leftarrow 1, n$  **hacer**
  - 5:      $C_i \leftarrow familia(X_i)$
  - 6: **fin para**
  - 7:  $A_j$  el conjunto de nodos asociados al conglomerado  $C_i$
  - 8: **para**  $j \leftarrow 1, n$  **hacer**
  - 9:      $\phi_j(c_i) \leftarrow \prod_{(x_j) \in A_j} p(x_i | \pi_j)$
  - 10: **fin para devolver:**  $P(x_i | e)$
- 

Este algoritmo de agrupamiento para redes bayesianas es muy importante en el presente trabajo, puesto que demuestra que dada una red bayesiana, es posible expresar la probabilidad de cada nodo factorizándola con los nodos de su familia y que no hace falta ningún nodo más de la red en este cálculo.

Sea la siguiente red de ejemplo, un subconjunto de la red bayesiana del “modelo de red bayesiana con pesos propuesto”:

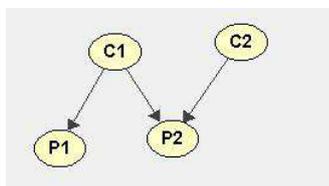


Figura 5.3: Red bayesiana de conceptos y preguntas realizada con ELVIRA [18]

Paso 1. Moralizar el grafo. Para moralizar el grafo, es necesario añadir la arista  $C1 \rightarrow C2$ . El grafo resultante es triangulado, por lo que no necesita ninguna transformación adicional.

Paso 2. Obtener los conglomerados del grafo. Los conglomerados del grafo son  $C_1 = C1, P1$  y  $C_2 = C1, C2, P2$ .

Paso 3. Asignar cada nodo a un conglomerado. Cada uno de los nodos ha de ser asignado a un conglomerado que contenga a su familia. Por ejemplo, los nodos  $C1, C2$  y  $P2$  se asignan al conglomerado  $C_2$ , y el nodo  $P1$  al conglomerado  $P1$ .

Paso 4. Definir las funciones potenciales. Tenemos las siguientes funciones potenciales:

$$\phi_1(C1, P1) = p(P1|C1) \quad (5.21)$$

$$\phi_2(C1, C2, P2) = p(P2|C1, C2) \quad (5.22)$$

Paso 5. Devolver las  $p(x_i|e)$  Una vez calculadas las funciones potenciales, ya tenemos los valores de las probabilidades condicionadas deseadas.

Por consiguiente, vemos que es posible calcular cada probabilidad condicionada confinada al subconjunto de nodos con el que está relacionada, que será por cada pregunta del sistema  $P_i$ , el propio nodo  $P_i$  y sus nodos padre.

### 5.3.2. Algoritmo de HUGIN

Vamos a aplicar ahora el algoritmo de agrupamiento, pero considerando ahora que nuestra red bayesiana es un modelo descomponible de Markov. Este algoritmo aunque también puede consultarse en [5] lo mostramos a continuación porque es muy importante la información que aporta sobre el modo de propagar la evidencia.

---

**Algoritmo 2** Algoritmo de agrupamiento para modelos descomponibles de Markov.

---

**entrada:** red bayesiana  $(D, P)$  definida en un conjunto de variables  $X$  y una evidencia  $E = e$

Moralizar y triangular el grafo  $D$ , obteniendo un grafo no dirigido  $G$ .

2:  $C \leftarrow$  elconjuntodelosconglomeradosde $G$

**para**  $i \leftarrow 1, n$  **hacer**

4:  $C_i \leftarrow$  familia( $X_i$ )

**fin para**

6:  $A_i \leftarrow C_i$

**para**  $C_i \leftarrow 1, n$  **hacer**

8:  $\phi_i(c_i) \leftarrow \prod_{(x_i) \in A_i} p(x_i|\pi_i)$

**fin para** devolver:  $P(x_i|e)$

---

Tras examinar los algoritmos 5.3.1 y algoritmo 2, concluimos que el algoritmo que va a utilizar el “modelo de red bayesiana con pesos propuesto” para la propagación de la evidencia por la red es una mezcla de los dos del siguiente modo: el algoritmo del “modelo de red bayesiana con pesos propuesto” propaga la evidencia solamente en el conglomerado del nodo del que se tiene evidencia como hace el algoritmo 2 pero tomando como conglomerado de un nodo su familia como hace el algoritmo 5.3.1.

Sea la siguiente red bayesiana, un subconjunto de la red bayesiana del “modelo de red bayesiana con pesos propuesto”:



Figura 5.4: Red bayesiana de conceptos y preguntas

Si aplicamos el algoritmo ?? a esta red de ejemplo tenemos:

Paso 1. Obtener los conglomerados de la red. En este caso, la red es sencilla con un único conglomerado  $C_1$  formado por los nodos  $C$  y  $P_1$ . Sus conjunto separador y residual son los siguientes:

$$S_1 = \emptyset \quad (5.23)$$

$$R_1 = C_1, P_1 \quad (5.24)$$

Paso 2. Calcular la probabilidad conjunta. Como sólo hay un conglomerado, tenemos directamente que la expresión de la probabilidad conjunta es:

$$p(P_1, C_1) = \phi(P_1, C_1) \quad (5.25)$$

Por tratarse de una red bayesiana, sabemos el valor de la función potencial:

$$\phi(P_1, C_1) = p(P_1|C_1)p(C_1) \quad (5.26)$$

El valor de  $m_1()$  es:

$$m_1() = \sum_{(P_1, C_1) \in A_i} p(P_1|C_1)p(C_1) \quad (5.27)$$

La expresión final de la probabilidad conjunta es:

$$\phi_1^*(P_1, C_1) = \frac{\phi_1(P_1, C_1)}{m_1()} \quad (5.28)$$

Esta expresión se corresponde con las probabilidades condicionadas de la siguiente tabla:

Una vez que se tienen las probabilidades conjuntas, para calcular la probabilidad de una variable, hay que marginalizar la probabilidad conjunta del siguiente modo:

5.3. ALGORITMOS DEL MODELO DE RED BAYESIANA CON PESOS  
PROPUESTO

P1	C1	$p(P1, C1)$
0	0	$\frac{(p(P1=0 C1=0)p(C1=0))}{\sum_{(P1, C1) \in A_i} p(P1 C1)p(C1)}$
0	0	$\frac{(p(P1=0 C1=1)p(C1=1))}{\sum_{(P1, C1) \in A_i} p(P1 C1)p(C1)}$
0	1	$\frac{(p(P1=1 C1=0)p(C1=0))}{\sum_{(P1, C1) \in A_i} p(P1 C1)p(C1)}$
0	1	$\frac{(p(P1=1 C1=1)p(C1=1))}{\sum_{(P1, C1) \in A_i} p(P1 C1)p(C1)}$

Tabla 5.1: Tabla probabilidades conjuntas calculadas por el algoritmo de HUGIN

Ante una nueva evidencia, por ejemplo,  $P1 = 1$ , que representa que el alumno ha respondido correctamente la pregunta P1, tenemos que particularizar esta expresión a los casos de  $P1 = 1$ , o lo que es lo mismo, absorber la evidencia dando como resultado:

$$p(C1|P1 = 1) = \frac{p(P1 = 1|C1 = 1)p(C1 = 1)}{p(P1 = 1|C1 = 0)p(C1 = 0) + p(P1 = 1|C1 = 1)p(C1 = 1)} \quad (5.29)$$

Si seguimos simplificando la ecuación 5.31:

$$p(C1|P1 = 1) = \frac{p(P1 = 1|C1 = 1)p(C1 = 1)}{p(P1 = 1)} \quad (5.30)$$

que será el valor de la probabilidad del nodo C1.

Cabe destacar que el denominador  $P1 = 1$  actúa como constante de normalización, que será el valor de  $\alpha$  de la ecuación 5.13 que normaliza la certeza o BEL(x) de un nodo X en el algoritmo de propagación de paso de mensajes (ver 5.2.8). Con esto deducimos que el algoritmo 2 se corresponde con una propagación de mensajes en cada nodo de la red. La diferencia es que hemos reducido la propagación de mensajes a cada conjunto conglomerado.

Si ahora para esta misma red de ejemplo con los nodos C1 y P1, utilizamos la herramienta software HUGIN LITE [25], asignando como valores de probabilidades  $p(C1 = 1) = 0,5$ ,  $p(C1 = 0) = 0,5$ ,  $p(P1 = 1|C1 = 1) = 0,7$  y propagando la evidencia  $P1 = 1$ , tenemos que devuelve como resultado de  $p(C1)^* = 0,1892$ , que se obtiene también con la ecuación 5.31 así:

$$p(C1|P1 = 1) = \frac{p(P1 = 1|C1 = 1)p(C1 = 1)}{p(P1 = 1)} = \frac{0,7 * 0,5}{0,1850} \quad (5.31)$$

Por tanto, podemos concluir que la herramienta HUGIN Lite o el algoritmo que la implementa conocido por algoritmo de HUGIN se corresponde con el algoritmo 2 para modelos de redes de Markov descomponibles.

En el caso de utilizar la herramienta HUGIN LITE, esta herramienta dispone de un API en varios lenguajes de programación (C++, Visual Basic, Java) con funciones para lo que llama “propagar la evidencia” y después “adaptar” la red, que consiste en asegurarse de que la red alcance un estado de equilibrio.

Un ejemplo de código de la API de Java de HUGIN LITE que se puede consultar en el manual de su API [26] es el siguiente:

```

h_domain_t d = h_kb_load_domain(".^sia.hkb",NULL);
h_node_t n = h_domain_get_node_by_name(d, "S");
            h_node_select_state(n,0);
h_domain_propagate(d,h_equilibrium_sum,h_mode_normal);
            h_domain_adapt;
    
```

Tabla 5.2: Ejemplo de código para propagación de evidencia con API JAVA de Hugin Lite [26]

Estas mismas funciones para propagar la evidencia las tiene HUGIN LITE en una versión en coma flotante, que en lugar de utilizar el valor de la constante de normalización utilizan una versión logarítmica de la constante para casos en los que las probabilidades obtenidas sean muy pequeñas.

Este algoritmo de HUGIN será el que se aplica también en el “modelo de red bayesiana con pesos propuesto” del presente trabajo, pero con un matiz más, que será que sólo se calculen las probabilidades conjuntas para los nodos de los que interesa calcular su probabilidad, que convierte al algoritmo de HUGIN es un “algoritmo orientado a objetivo”.

### 5.3.3. Algoritmo orientado a un objetivo

Hasta ahora, hemos visto cómo se propaga la evidencia en todos los nodos no evidenciales de la red  $X_i$  dada una evidencia  $E = e$ . Este proceso es correcto, pero supongamos que sólo estamos interesados en conocer la probabilidad de un único nodo. Según el algoritmo orientado a un objetivo, este nodo será el *nodo objetivo*.

Lo que hace este algoritmo, es reducir el grafo de partida, considerando únicamente el conjunto de nodos *relevantes*, y a continuación, realizar la propagación en el grafo reducido. El proceso para la obtención de los nodos relevantes puede consultarse en [5]. De este algoritmo, sacamos como conclusión fundamental que el conjunto de nodos relevantes de un nodo lo

conforman él mismo y sus nodos ascendientes, que en el caso de cualquier subred del “modelo de red bayesiana con pesos propuesto” 6.3.2 será el conjunto formado por el propio nodo y sus nodos padre.

En el Anexo I, puede consultarse la implementación final del algoritmo de HUGIN orientado a un objetivo que utiliza el “modelo de red bayesiana con pesos propuesto” del capítulo 6.

### 5.4. Justificación del uso de Redes Bayesianas

Las redes bayesianas proporcionan un buen método para representar y razonar acerca de la incertidumbre presente en el modelado del estudiante [47]. Las redes bayesianas son una manera de estructurar estos modelos y permiten definir un tratamiento sistemático de la información que se va recopilando en un sistema de aprendizaje.

En un sistema de este tipo, según Reye [45], es necesario el conocimiento del alumno en cualquier momento en el tiempo. Los sistemas que son capaces de medir cada nuevo ítem de evidencia en conjunción con el resto de evidencias previas que se tengan del estudiante, permiten tomar decisiones pedagógicas, y por consiguiente, son más adecuados para este contexto educativo. Otro enfoque que puede tratar con la información sobre el estudiante son los sistemas de reglas, pero los sistemas de reglas que manejan incertidumbre, hemos razonado en la sección 3.4.1 que, a menudo, producen un conjunto de reglas inconsistentes.

Parece más adecuado utilizar un modelo probabilístico para manejar la incertidumbre que contenga en su representación la relación entre las variables del modelo. Las opciones disponibles son las redes de Markov y las redes Bayesianas.

Castillo et. al., [5] nos dan la clave para distinguir entre las redes de Markov y las redes bayesianas asegurando que en el caso de las redes bayesianas, la factorización de la función de probabilidad se obtiene de forma sencilla a partir del grafo dirigido considerando un conjunto de funciones de probabilidad condicionada que involucran a cada nodo con sus padres. Y que además, la factorización de una red de Markov requiere varios pasos previos como: construir los conglomerados del grafo, ordenar este conjunto de modo que satisfaga la propiedad de intersección dinámica, encontrar los conjuntos separadores y residuales de cada conglomerado, etc. Por ello, en este trabajo hemos considerado que el uso de las redes bayesianas resulta más sencillo y como hemos visto en los algoritmos de agrupamiento explicados, las probabilidades condicionadas de la red bayesiana ya definen los conglomerados para poder aplicar estos algoritmos de agrupamiento de una forma muy directa.

Las redes bayesianas tienen el inconveniente de que es necesario especificar las probabilidades condicionadas de la red inicialmente antes de propagar la evidencia por la red, que serán los parámetros de la red. Es un problema trabajar con redes bayesianas cuando el número de parámetros a especificar es muy elevado. Pero nuestro trabajo se centra en los trabajos anteriormente desarrollados hasta el momento que han tratado de simplificar la forma de calcular los parámetros de la red, utilizando métodos sistemáticos para calcularlos de forma sencilla. En el presente trabajo, se propone otro método sencillo de cálculo de los parámetros de la red que era el segundo objetivo principal del trabajo puesto que el primario es que el valor de las probabilidades calculadas represente con mayor precisión las relaciones reales entre las variables de la red.

Otra ventaja de utilizar las redes bayesianas es que existe software comercial para diseñar y operar redes bayesianas como es HUGIN LITE [25] y NETICA [41]. Además existen librerías desarrolladas en diferentes lenguajes de programación que pueden ser integradas en un sistema nuevo desarrollado, como por ejemplo, las API de C++, Visual Basic y Java de HUGIN [26], JAVABAYES [33] y SMILE [49].

#### 5.4. JUSTIFICACIÓN DEL USO DE REDES BAYESIANAS

---

## Capítulo 6

# MODELO BAYESIANO DEL ALUMNO CON PESOS PROPUESTO

### 6.1. Introducción

En este capítulo describiremos un modelo del alumno basado en redes bayesianas. Este modelo del alumno presentado es el que hemos renombrado al principio del trabajo como “modelo de red bayesiana con pesos propuesto”. Este modelo es una modificación del que hemos también renombrado “modelo de red bayesiana de Millán” que se define en [38]. A lo largo del capítulo, iremos citando los dos modelos para destacar las diferencias existentes entre ambos.

### 6.2. Modelado del dominio

Antes de definir el modelo del estudiante, es importante definir el modelado del dominio del sistema del test adaptativo en el que está embebido, para ver los puntos coincidentes, ya que como hemos comentado en el capítulo 2, hay sistemas en los que el modelo del dominio y del estudiante son el mismo y otros, en los que hay diferencias.

El modelo del dominio, como en cualquier sistema de enseñanza\_ aprendizaje, es con el que trata el sistema, es decir, es su base de conocimiento.

El conocimiento es el una materia de estudio de la rama de Informática, Sistemas Operativos. Esta asignatura pertenece al primer curso del ciclo formativo de formación profesional de grado superior denominado “Administración de Sistemas Informáticos en Red”.

El conocimiento de la materia “Sistemas Operativos” se divide en una serie de “tópicos” o áreas de conocimiento:

1. Introducción a los Sistemas Informáticos.
2. Introducción a los Sistemas Operativos.
3. Gestión de procesos.
4. Gestión de memoria.
5. Gestión de entrada y salida.

Al definir estos cinco tópicos, se pretende que el conocimiento del sistema sea variado y equilibrado, porque el banco de preguntas del test va a contener preguntas de todos los tópicos y en el mismo número (o casi el mismo) en cada uno. El test adaptativo que contiene el “modelo de red bayesiana con pesos propuesto” no se va a preocupar, no obstante, de tener un equilibrio de contenido, cuestión que plantearemos como una futura línea de investigación en el capítulo 8.

El conocimiento del sistema se va a dividir en “conceptos” que son las unidades mínimas en las que lo dividimos para tratar con él de una forma más precisa y manejable.

El profesor, al igual que hace el sistema SIETTE, va a definir los conceptos que él considera en que se descompone la materia de “Sistemas Operativos” en cada uno de los tópicos considerados.

### 6.3. Modelo del estudiante

El módulo del estudiante debe contener toda la información del estudiante que recaba el sistema para poder evaluarle y guiarle en sus proceso de aprendizaje.

La información del modelo del estudiante son un subconjunto de “conceptos”, que se corresponden con las unidades de aprendizaje “conceptos” de las que se compone el modelo del dominio. El modelo del estudiante definido es, por tanto, un modelo de *superposición* [38], en el que el conocimiento del estudiante es un subconjunto del conocimiento del sistema.

#### 6.3.1. Nodos de la red bayesiana

El conocimiento del sistema se va a modelar con una red bayesiana que está formada por dos tipos de nodos:

## CAPÍTULO 6. MODELO BAYESIANO DEL ALUMNO CON PESOS PROPUESTO

---

- Nodos que representan el conocimiento del alumno. Son nodos “concepto”.
- Nodos que permiten recolectar evidencia, con los que se pretende recoger todo aquello que proporcione información sobre el conocimiento del alumno. Son nodos “pregunta”.

En cuanto a los “nodos concepto”, hay un nodo por cada “concepto” del sistema, que representa el conocimiento del alumno sobre el concepto. Estos nodos son binarios, es decir, toman valor 1 cuando el alumno “conoce” el concepto, y valor 0 cuando lo desconoce.

En cuanto a los nodos pregunta, hay un nodo pregunta por cada “pregunta del sistema, que representa el total de pregunta que “conoce” el alumno dados los conceptos relacionados con ella que el alumno domina.

Al ser la red bayesiana un modelo probabilístico, cada nodo de la red tendrá asociado un valor de probabilidad, que será el valor de conocimiento del alumno en cada nodo.

A diferencia del “modelo de red bayesiana de Millán” descrito en el apartado ?? y cuya descripción más completa puede encontrarse en [38], sólo vamos a considerar estos nodos de la red porque el objetivo del presente trabajo es la estimación del conocimiento del alumno en cada uno de los conceptos del sistema, que se hará a partir de las respuestas “correctas” o “incorrectas” a las preguntas del test que se le presenten.

### 6.3.2. Relaciones entre los nodos

Las relaciones de la red bayesiana son: relaciones concepto-pregunta. Son relaciones causales que representan que el conocimiento de partes del sistema (conceptos) tiene influencia causal en que las preguntas del sistema se respondan correcta o incorrectamente.

Las relaciones entre nodos concepto y nodos pregunta se representan en la siguiente red bayesiana:

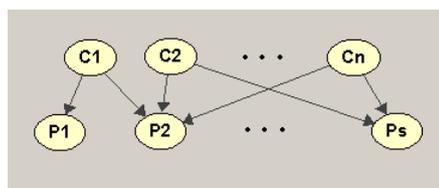


Figura 6.1: Relaciones entre conceptos y preguntas realizada con ELVIRA [18]

Hay que señalar que este modelo bayesiano es un modelo dinámico, en el sentido de que el conocimiento del alumno sobre los nodos pregunta puede cambiar a lo largo del tiempo, pudiendo variar también el conocimiento sobre el resto de nodos de la red.

## 6.4. Proceso de diagnóstico

La parte de diagnóstico de la red, es la que determina, a partir de las respuestas del alumno, el conjunto de conceptos que conoce y desconoce.

El algoritmo de diagnóstico funciona del siguiente modo:

- Paso 1. Determinar el nivel de conocimiento inicial del alumno. El alumno tiene un conjunto de conceptos que inicialmente conoce.
- Paso 2. Se selecciona una pregunta del sistema para mostrar al alumno cuyo nivel de dificultad se corresponde con su nivel de habilidad. Después comentaremos el algoritmo de selección de pregunta utilizado.
- Paso 3. El alumno responde a la pregunta. Independientemente de tener evidencia positiva (la respuesta a la pregunta es correcta) o evidencia negativa (la respuesta a la pregunta es incorrecta), se actualiza el grado de conocimiento del alumno en los conceptos que están exclusivamente relacionados con la pregunta. Más tarde, comentaremos también el algoritmo de propagación de evidencia utilizado en esta red bayesiana.
- Paso 4. Se examina el grado de conocimiento del alumno en todos y cada uno de los conceptos del sistema. Si el alumno conoce todos los conceptos o bien se ha mostrado hasta un número fijado de preguntas del sistema, el algoritmo termina calculando el nivel de habilidad final del alumno en base al número de conceptos conocidos. Si el alumno no conoce todos los conceptos del sistema o todavía no se ha llegado al máximo número de preguntas a mostrar, se selecciona una pregunta del sistema volviendo al Paso 2.

Este algoritmo realiza dos pasos fundamentales, que es esencial examinar con mayor detalle: determinación del grado de conocimiento del alumno y selección de la pregunta.

1. Determinación del grado de conocimiento. Este proceso consiste en calcular para cada concepto del sistema, el tanto por ciento de conocimiento que tiene el alumno. Esto es equivalente a calcular las probabilidades a posteriori de cada concepto del sistema, que podemos representar por

$p^*(C1), p^*(C2), p^*(C3), \dots, p^*(Cp)$ . Estas probabilidades son diferentes a las probabilidades a priori que se fijan inicialmente para el alumno y se denotan por  $p(C1), p(C2), p(C3), \dots, p(Cp)$ .

El cálculo de las probabilidades  $p^*(Ci)$  se realiza tras la respuesta del alumno a la pregunta que se le presenta en cada momento, de tal modo sólo se actualizan las  $p^*(Ci)$  que influyan directamente en la pregunta recién respondida. Así, si denotamos por  $P$  la pregunta contestada, se calcula alternativamente  $p(Ci|P_{correcta})$  o  $p(Ci|P_{incorrecta})$  dependiendo de que la respuesta a la pregunta haya sido correcta o incorrecta. Todo este proceso de determinación del grado de conocimiento del alumno se describe en el algoritmo.

2. Selección de la pregunta. Todo test adaptativo consta de un elemento básico que lo define que es el método de selección de preguntas. Para este test adaptativo bayesiano, se sigue el algoritmo que se describe en el apartado.

## 6.5. Diagnóstico de cada ítem

El diagnóstico de la red bayesiana consiste en decidir si el alumno en base a un conjunto de conceptos que se han evaluado como que “conoce”, es capaz de responder correctamente a cada ítem del test. Este valor es, por tanto, la probabilidad de responder correctamente una pregunta dados los conceptos conocidos.

La probabilidad de cada nodo pregunta, o lo que es lo mismo, el modelo de respuesta asociado a cada pregunta del test adaptativo bayesiano, viene dado por la fórmula de ICC de la TRI 3-paramétrica que modificamos al igual que en el “modelo de red bayesiana de Millán” explicado en el apartado de la siguiente manera:

Sean  $C1, \dots, Cq$  el conjunto de conceptos del sistema que están relacionados (por relaciones de causalidad) con un ítem recién respondido que denotaremos por  $P1$ , se modifica la ICC con la siguiente expresión:

$p(P1_{correcta}|C1, C2, \dots, Cq) = G(x|C1, C2, \dots, Cq)$  que es la probabilidad de responder correctamente la pregunta dados todos los conceptos padre implicados con sus posibles estados “conocido” y “desconocido”.

Por tanto, para calcular la probabilidad de responder correctamente a una pregunta dada una combinación de estados de los conceptos que influyen directamente en la pregunta, se calcula una probabilidad condicionada evaluando la función  $G(x)$  en un valor. A continuación, explicamos la procedencia de la función  $G(x)$  y los valores que puede tomar.

El modelo IRT de 3-parámetros calibra cada pregunta del test con 3 parámetros con el fin de adaptar cada ítem al nivel de conocimiento que tiene el alumno en cada momento. Los tres parámetros son:

- a: factor de discriminación. Este factor que permite medir lo que influye la pregunta en el total del test, en este modelo bayesiano se le asignado un valor igual para todos los ítems y de esta manera estaríamos diciendo que todas las preguntas de un test tienen la misma importancia, no sobresaliendo ninguna sobre las demás. Esto se ha decidido así porque uno de los parámetros que más ha interesado medir en este estudio es la precisión del modelo de red bayesiana en la estimación del nivel de conocimiento del alumno y el parámetro IRT que está directamente relacionado con el grado de conocimiento es el factor b o el nivel de dificultad de cada pregunta.
- b: nivel de dificultad. Este factor mide el grado de dificultad de la pregunta. Se considera, igual que sucede en SIETTE, que toma la misma escala de valores que el parámetro nivel de conocimiento  $\Theta$ . Estos valores son discretos y varían en el rango  $[0 \dots \Theta]$ .
- c: factor de azar. Este factor mide la posibilidad de que el alumno escoja cualquier respuesta al azar al responder la pregunta. Su valor, al igual que sucede en SIETTE, depende del número de posibles respuestas y es igual a  $1/n$ , siendo n el total de posibles respuestas.

La expresión que sigue cada ítem es la curva característica de ítems ICC. Esta curva contiene los valores de probabilidad de respuesta correcta del ítem para todos los posibles valores de habilidad del alumno. La curva ICC del modelo 3-paramétrico de la IRT tiene la siguiente expresión:

$$F(x) = c + \frac{1 - c}{1 + \exp(-1,7a(x - b))} \quad (6.1)$$

Esta función  $F(x)$  se convierte en la función  $G(x)$  mediante una transformación lineal, para que el valor mínimo de la probabilidad no sea 0, sino el factor de azar  $c$ , y asimismo, el valor máximo de la probabilidad sea  $1 - s$  (donde  $s$  es el factor de descuido) y no el valor máximo 1.

$$G(x) = 1 - \frac{(1 - c)(1 + \exp(-1,7ab))}{1 + \exp(-1,7a(x - b))} \quad (6.2)$$

Esta misma función  $G(x)$  se había considerado en el “modelo de red bayesiana de Millán” descrito en el apartado.

Los valores de probabilidad que toma la función  $G(x)$  son:

## CAPÍTULO 6. MODELO BAYESIANO DEL ALUMNO CON PESOS PROPUESTO

---

- $G(0) = c$ .  $G(0)$  es la probabilidad de responder correctamente la pregunta cuando el alumno no tiene ningún nivel de conocimiento  $\Theta$  sobre la materia de estudio. Como el nivel de habilidad del alumno en el modelo bayesiano se mide en función de los “conceptos” del sistema, en esta función  $G(x)$ , consideraremos que  $x$  es la variable que contiene el número de conceptos dominados por el alumno.  $G(0)$ , será por tanto, la probabilidad de responder correctamente la pregunta cuando el alumno no sabe ningún concepto.
- $G(x) = 1 - s$ .  $G(x)$  es la probabilidad de responder correctamente la pregunta cuando el alumno conoce todos los conceptos. El alumno hace todo lo contrario a “adivinar” la pregunta, que es lo que indica el factor de adivinanza  $s$ .

El resto de probabilidades se calculan teniendo en cuenta los conceptos que va conociendo el alumno durante la ejecución del test bayesiano, con lo que se han de considerar todas las posibles combinaciones de estados de los conceptos (“conocido” o “desconocido”) que estén relacionados con la pregunta. Hay un total de  $2^p$  combinaciones de estados de los conceptos para  $p$  conceptos. Al final, por cada pregunta del sistema, habrá un total de  $2^p$  probabilidades condicionadas a los conceptos de la misma.

En el “modelo de red bayesiana de Millán” del apartado, se mostraba un método de cálculo de las  $2^p$  probabilidades. Este modelo de Millán asigna a cada probabilidad, el valor de  $G(x)$  evaluada en un “factor” que se calcula, para cada probabilidad, como:

$$factor = \frac{(k * x^*)}{2^p - 1} \quad (6.3)$$

donde  $x^*$  es el valor de  $x^*$  para el que  $G(x^*)$  tiene el valor máximo de probabilidad para el valor  $s$  de descuido del alumno. Los valores de las probabilidades del “modelo de red bayesiana de Millán” va desde  $c$  hasta  $1 - s$ .

En el “modelo de red bayesiana con pesos propuesto”, este factor en lugar de calcularse de este modo, se calcula con el parámetro “peso” de cada concepto en la pregunta. El “peso” se asigna a cada concepto, no teniendo por qué coincidir el peso de un concepto con el de otro, y se mide con una proporción del concepto en la pregunta, por lo que es un valor entre 0 y 1. El factor contiene en su fórmula los “pesos” sumados de todos los conceptos de la pregunta con una doble consideración:

1. Si en la combinación de conceptos, un concepto está en el estado “conocido”, interviene en la suma de los pesos con el valor del peso del concepto.

- Si en la combinación de conceptos, un concepto está en el estado “desconocido”, interviene en la suma de los pesos con el valor 1 menos el peso del concepto, es decir, con el complementario del peso.

El factor en el “modelo de red bayesiana con pesos propuesto” tendrá la expresión:

$$factor = \frac{(pesotodosconceptos * x^* * (2^p - 1))}{2^p - 1} \quad (6.4)$$

Comparando las ecuaciones 3.3 y 3.4, deducimos que:

$$K_{propuesta} = (pesotodosconceptos * (2^p - 1)) \quad (6.5)$$

Para obtener la probabilidad de responder correctamente la pregunta se aplicará la fórmula final:

$$G(x|C1...Cp) = G\left(\frac{pesotodosconceptos * x^* * 2^p - 1}{2^p - 1}\right) \quad (6.6)$$

donde “pesotodosconceptos” es la suma de los  $peso_i$  (para los conceptos “conocidos”) y  $1 - peso_i$  (para los conceptos “desconocidos”).

Vamos a explicar los componentes de esta fórmula:

- $pesotodosconceptos * 2^p$ . Esta multiplicación se hace para obtener un número que será siempre menor o igual a  $2^p$ , que es justo el valor máximo del factor k que se utiliza en el “modelo de red bayesiana de Millán”. Este producto, será, por tanto, el factor k correspondiente al “modelo de red bayesiana con pesos propuesto”.
- $pesotodosconceptos * x^* * 2^p$ . Esta multiplicación del factor por el valor  $x^*$  es el mismo producto que se realiza en el “factor” del “modelo de red bayesiana de Millán”. Se obtiene un número proporcional al valor  $x^*$ .
- $\frac{pesotodosconceptos * x^* * 2^p}{2^p - 1}$ . Este cociente se hace para que el valor del factor esté distribuido en el rango de valores 1 y  $2^p - 1$ , que son los límites máximo y mínimo del número de combinaciones posibles de los conceptos de la pregunta, o el número de probabilidades condicionadas de la pregunta a calcular.

Para ilustrar mejor el método propuesto de cálculo de las probabilidades condicionadas de la red, vamos a verlo con un ejemplo:

Para ilustrar mejor el método propuesto de cálculo de las probabilidades condicionadas de la red, vamos a verlo con un ejemplo:

Sean C1 y C2 dos conceptos asociados a la pregunta P del sistema que se presenta al alumno en el test. Para saber el conocimiento del alumno en la

CAPÍTULO 6. MODELO BAYESIANO DEL ALUMNO CON PESOS  
PROPUESTO

---

pregunta, pueden darse las siguientes situaciones: desconoce C1 y desconoce C2, desconoce C1 y conoce C2, conoce C1 y desconoce C2, conoce C1 y C2. El conjunto de probabilidades condicionadas que tenemos son:

$$p(P_{correcta}|C1_{desconocido}C2_{desconocido}) = G\left(\frac{((1 - peso_1) + (1 - peso_2)) * x^* * 2^p}{2^p - 1}\right) \quad (6.7)$$

$$p(P_{correcta}|C1_{desconocido}C2_{conocido}) = G\left(\frac{((1 - peso_1) + peso_2) * x^* * 2^p}{2^p - 1}\right) \quad (6.8)$$

$$p(P_{correcta}|C1_{conocido}C2_{desconocido}) = G\left(\frac{(peso_1 + (1 - peso_2)) * x^* * 2^p}{2^p - 1}\right) \quad (6.9)$$

$$p(P_{correcta}|C1_{conocido}C2_{conocido}) = G\left(\frac{(peso_1 + peso_2) * x^* * 2^p}{2^p - 1}\right) \quad (6.10)$$

donde *peso1* es el peso del concepto C1 y *peso2* es el peso del concepto C2.

La última probabilidad  $p(P_{correcta}|C1_{conocido}C2_{conocido})$  en realidad no haría falta calcularla, porque sabemos que es igual a  $1 - s$ .

En este punto, falta calcular las probabilidades a posteriori para los conceptos,  $p^*(C1)$  y  $p^*(C2)$ . Para ello, supongamos que tenemos la evidencia de que la pregunta P se ha respondido correctamente, entonces tenemos:

$$p^*(C1) = \left(\frac{p(C1_{conocido}|P_{correcta})}{p(P_{correcta})}\right) \quad (6.11)$$

En esta ecuación,  $p(C1_{conocido}|P_{correcta})$  puede calcularse aplicando el Teorema de Bayes así:

$$p^*(C1_{conocido}|P_{correcta}) = p(P_{correcta}|C1_{conocido}) * p(C1_{conocido}) \quad (6.12)$$

Con lo que sustituyendo en la ecuación 3.9 tenemos:

$$p^*(C1) = \left(\frac{p(P_{correcta}|C1_{conocido}) * p(C1_{conocido})}{p(P_{correcta})}\right) \quad (6.13)$$

donde  $1/P_{correcta}$  se calcula como una constante de normalización dividiendo  $p(C1_{conocido}|P_{correcta})$  por la suma de  $P_{correcta}$  y de  $P_{incorrecta}$ .

En el caso de tener la evidencia de que la pregunta se ha respondido incorrectamente, la ecuación 3.9 quedaría del modo:

$$p^*(C1) = \left( \frac{p(P_{incorrecta}|C1_{conocido}) * p(C1_{conocido})}{p(P_{incorrecta})} \right) \quad (6.14)$$

Las probabilidades  $p(P_{correcta}|C1_{conocido})$  y  $p(P_{incorrecta}|C1_{conocido})$  son complementarias y suman 1.

Análogamente, el proceso seguido para calcular  $p^*(C1)$  se sigue para calcular  $p^*(C2)$ . Veremos en el algoritmo, que si las probabilidades  $p^*(C1)$  y  $p^*(C2)$  superan un valor fijado por el sistema de 0.7, se concluye que el alumno “conoce” los conceptos C1 y C2.

---

**Algoritmo 3** Propagación de evidencia en red bayesiana propuesta.

---

- 1:  $p(C_i) = 0,7$ ; para todos aquellos conceptos que en la simulación se hayan deducido como dominados inicialmente por el alumno.
  - 2: Se selecciona una primera pregunta  $P$  con el algoritmo 2.
  - 3: **para**  $C_i \leftarrow 1, p$  **hacer**
  - 4:      $p(P|C1C2...Cp) = G(x|C1...Cp) \leftarrow G\left(\frac{pesotodosconceptos*x*2^p-1}{2^p-1}\right)$
  - 5: **fin para**
  - 6: **si**  $p_i$  es evidencia positiva **then**
  - 7:      $p(p_i|C) \leftarrow p(+p_i|C)$
  - 8: **else si** **then**  $p(p_i|C) \leftarrow p(-p_i|C)$
  - 9: **fin si** ▷ Calcular el factor de propagación  $\lambda$
  - 10:  $\lambda \leftarrow \prod_{(1,k) \in i} p(p_i|C)$
  - 11:  $P^*(C) = \alpha * \lambda * P(C)$  **devolver:**  $P^*(C)$
- 

Este algoritmo se muestra con todo detalle en el algoritmo 0 del Anexo I.

## 6.6. Criterio de selección de preguntas

Existen varios criterios a considerar par seleccionar el siguiente ítem al alumno, tal como sucedía en SIETTE.

En el “modelo de red bayesiana con pesos propuesto” del presente trabajo, al igual que se hacía en el “modelo de red bayesiana de Millán”, utilizaremos un *criterio adaptativo*.

Tomando como base el “modelo de red bayesiana de Millán” descrito en [37], la selección de la siguiente pregunta la haremos en base al rendimiento que haya mostrado el alumno en las preguntas anteriores, y más concretamente, en la estimación del nivel de conocimiento que se tenga del alumno en base a las preguntas anteriores. La autora explica que podemos adoptar dos tipos

de criterios adaptativos diferentes: los criterios basados en la cantidad de información que aporta cada pregunta y los criterios condicionados, que se basan en potenciar el comportamiento que ha demostrado el alumno hasta el momento.

Parece más lógico utilizar el segundo criterio condicionado dado el carácter dinámico de la red.

En [38], se presentan dos definiciones para la “cantidad de información” que aporta una pregunta, también denominada “utilidad de la pregunta”. En la primera de las definiciones, se define la utilidad de la pregunta en función del cambio que se produce en la probabilidad de un concepto dada la respuesta a la pregunta. Y en la segunda definición, se define la utilidad de la pregunta en función del cambio que se produce en la probabilidad de la propia pregunta al responderla.

La expresión de la “utilidad de la pregunta dado el cambio en la pregunta es el siguiente:

$$U(P, C) = (p(C = 1|P = 1) * p(P = 1) + (p(C = 0|P = 0) * p(P = 0) \quad (6.15)$$

La expresión de la “utilidad de la pregunta dado el cambio en un concepto de la pregunta es el siguiente:

$$U(P, C) = (p(P = 1|C = 1) * p(C = 1) + (p(P = 0|C = 0) * p(C = 0) \quad (6.16)$$

Elegimos finalmente el criterio de utilidad de la pregunta dado la pregunta porque según los experimentos descritos en [31], es el criterio que obtiene mejores resultados en la estimación del nivel de habilidad del alumno.

Una vez definida la utilidad de una pregunta para cada uno de los conceptos implicados en ella, falta por definir la utilidad global de una pregunta. Hay también varias definiciones, que son:

1. Criterio de la suma: la utilidad global de una pregunta se define como la suma de las utilidades de la pregunta para cada uno de los conceptos con ella relacionados.
2. Criterio del máximo: la utilidad global de una pregunta se define como el valor máximo de todas las utilidades de la pregunta de entre todos los conceptos de la pregunta con ella relacionados.

En el presente trabajo, optamos por el criterio del máximo de la pregunta, tanto en el algoritmo de diagnóstico del “modelo de red bayesiana con pesos propuesto” 0 como en el algoritmo de diagnóstico del “modelo de red bayesiana de Millán” 0. Este algoritmo de selección de la siguiente pregunta aparece implementado con todo detalle en el Anexo I en A.

---

**Algoritmo 4** Selección de la siguiente pregunta.

---

```

para  $P_i \leftarrow 1, s$  hacer
2:   para  $C_i \leftarrow 1, p$  hacer
       $\max(p(P|C)) \leftarrow \text{MAX}p(P|C)$ 
4:   fin para
fin paradevolver:  $\max p(P|C)$ 

```

---

## 6.7. Comparación de dos modelos bayesianos

En esta sección, vamos a comparar los valores de las probabilidades condicionadas de la red bayesiana en los dos modelos bayesianos, “modelo de red bayesiana de Millán” y “modelo de red bayesiana con pesos propuesto” definido en este capítulo, porque como habíamos comentado al principio del capítulo, el “modelo de red bayesiana con pesos propuesto” está basado en el “modelo de red bayesiana de Millán” y pretende ser una mejora con respecto al anterior.

Vamos a ilustrar el cálculo de las probabilidades condicionadas en ambos modelos bayesianos a través de un ejemplo.

Sean C1, C2 y C3 con pesos  $w_1 = 0,7$ ,  $w_2 = 0,2$  y  $w_3 = 0,1$ . Sea una pregunta del sistema P cuyos parámetros de IRT son  $a = 0,3$ ,  $c = 0,25$ ,  $b = 5$  y  $s = 0,25$ . Con estos parámetros a, b, c y s, tenemos que  $G(x^*) = 1 - s = 1 - 0,25 = 0,99$  y  $x^* = 13,716$ .

En la siguiente tabla, mostramos los valores de los factores  $k$  de ambas redes y de las probabilidades condicionadas  $G(x)$ .

C1	C2	C3	k(Millán)	k(propuesta)	G(Millán)	G(propuesta)
0	0	0	0	0	G(0)	G(0)
0	0	1	1	1.47	G(1.95)	G(2.88)
0	1	0	2	2.24	G(3.91)	G(4.38)
0	1	1	3	6.09	G(5.87)	G(4.52)
1	0	0	4	6.09	G(7.83)	G(11.93)
1	0	1	5	6.16	G(9.79)	G(12.07)
1	1	0	6	6.93	G(11.75)	G(13.57)
1	1	1	7	7	G(13.716)	G(13.716)

Tabla 6.1: Valores de las probabilidades condicionadas de  $G(x)$  en ambos modelos bayesianos

## CAPÍTULO 6. MODELO BAYESIANO DEL ALUMNO CON PESOS PROPUESTO

---

Siguiendo la ecuación 3.5 y teniendo en cuenta que el parámetro  $k$  en el “modelo de red bayesiana de Millán” va desde  $0..,2^{p-1}$  para cada probabilidad condicionada, vamos a detallar el cálculo de los valores del factor  $k$  en ambas redes:

Para la combinación de estados de conceptos 001:

$$k_{millan} = 1 \quad (6.17)$$

$$k_{propuesto} = ((0,1) + \frac{(1-0,2)}{10} + \frac{(1-0,7)}{10}) * 7 = (0,1 + 0,08 + 0,03) * 7 = 1,47 \quad (6.18)$$

Para la combinación de estados de conceptos 010:

$$k_{millan} = 2 \quad (6.19)$$

$$k_{propuesto} = ((0,2) + \frac{(1-0,7)}{10} + \frac{(1-0,1)}{10}) * 7 = (0,2 + 0,03 + 0,09) * 7 = 2,24 \quad (6.20)$$

Para la combinación de estados de conceptos 011:

$$k_{millan} = 3 \quad (6.21)$$

$$k_{propuesto} = ((0,2) + (0,1) + \frac{(1-0,7)}{10}) * 7 = (0,2 + 0,1 + 0,03) * 7 = 2,31 \quad (6.22)$$

Para la combinación de estados de conceptos 100:

$$k_{millan} = 4 \quad (6.23)$$

$$k_{propuesto} = ((0,7) + \frac{(1-0,2)}{10} + \frac{(1-0,1)}{10}) * 7 = (0,7 + 0,08 + 0,09) * 7 = 6,09 \quad (6.24)$$

Para la combinación de estados de conceptos 101:

$$k_{millan} = 5 \quad (6.25)$$

$$k_{propuesto} = ((0,7) + \frac{(1-0,2)}{10} + 0,1) * 7 = (0,7 + 0,08 + 0,1) * 7 = 6,16 \quad (6.26)$$

## 6.7. COMPARACIÓN DE DOS MODELOS BAYESIANOS

---

Para la combinación de estados de conceptos 110:

$$k_{millan} = 6 \quad (6.27)$$

$$k_{propuesto} = ((0,7) + (0,2) + \frac{(1 - 0,1)}{10}) * 7 = (0,7 + 0,2 + 0,09) * 7 = 6,93 \quad (6.28)$$

Para las combinaciones 000 y 111, sabemos que las probabilidades será  $G(0) = 0.25$  y  $G(13.716) = 0.99$  en ambas redes.

Si hacemos una comparativa de los valores de  $x$  en  $G(x)$  en cada una de las dos redes, podemos ver que en la mayoría de los casos, no en todos, el valor de  $x$  en el “modelo de red bayesiana con pesos propuesto” es mayor que el valor de  $x$  en el “modelo de red bayesiana de Millán”. Es por este motivo que el valor de  $G(x)$  conducirá a mayores valores de las probabilidades a posteriori de los conceptos. Los resultados de las probabilidades en ambas redes se detallan en el capítulo 7.

Puede deducirse también que tanto el “modelo de red bayesiana de Millán” como “el modelo de red bayesiana con pesos propuesto” realizan un cálculo de un total de  $2^p$  probabilidades condicionadas, donde  $p$  es el número de conceptos del sistema. De este modo, cumplimos con el segundo objetivo planteado al principio del trabajo que consistía en definir un modelo bayesiano con un método sencillo y rápido de los parámetros de la red. Este orden exponencial coincide en los dos modelos bayesianos y no se puede disminuir ya que en un principio, antes de poner en funcionamiento la red para inferir conocimiento a través de las evidencia(s) disponibles, se necesitan todos sus parámetros.

La probabilidad de respuesta correcta se puede obtener igualmente utilizando cualquier software especializado en redes bayesianas, como puede ser ELVIRA [18] o HUGIN LITE [25].

# Capítulo 7

## EXPERIMENTACIÓN DEL MODELO Y RESULTADOS

### 7.1. Objetivos de los experimentos

El objetivo de los experimentos es comparar la eficacia en la estimación del nivel de conocimiento del alumno en las dos redes bayesianas correspondiente a los modelos, “modelo de red bayesiana de Millán” del que parte nuestro trabajo y “modelo de red bayesiana con pesos propuesto”, presentado en este trabajo.

Para poder hacer esta comparación, es preciso considerar una o varias medidas de evaluación que determinen la bondad de la estimación de cada sistema. Como medidas de evaluación, vamos a considerar los mismos parámetros que usa el “modelo de Millán” en [38], que son:

- Número de conceptos dominados: es el número de conceptos que la red bayesiana diagnostica como que el alumno domina. Para determinar si un concepto es "dominado", el algoritmo compara si la probabilidad a posteriori del concepto supera un valor umbral, que hemos fijado en 0.7 en la realización de los experimentos.
- Número conceptos no dominados: es el número de conceptos que la red bayesiana diagnostica como que el alumno no domina. Para determinar si un concepto es "no dominado", el algoritmo compara si la probabilidad a posteriori del concepto es menor que un valor umbral, que hemos fijado en 0.3 para la realización de los experimentos.
- Número conceptos no diagnosticados: es el número de conceptos que la red bayesiana diagnostica como no diagnosticados. Para determi-

nar si un concepto es "no diagnosticado", el algoritmo compara si la probabilidad a posteriori del concepto es mayor de 0.3 y menor de 0.7.

El número de conceptos dominados nos permitirá saber con certeza que el alumno conoce perfectamente los conceptos y así concluir con que no es necesario reevaluar al alumno constantemente en ellos.

El número de conceptos no dominados nos permitirá saber con certeza qué conceptos el alumno no domina, y poder así, insistir en su aprendizaje, por ejemplo, con preguntas relacionadas con esos conceptos.

El número de conceptos no diagnosticados nos permite dudar de que los conceptos sean con cierta certeza "dominados", o por el contrario, "no dominados", aunque después de examinar los resultados de los experimentos, veremos que los conceptos no diagnosticados se acercan más a los dominados.

La realización de los experimentos se ha llevado a cabo en dos escenarios:

1. Experimentación con alumnos simulados.
2. Experimentación con alumnos reales.

Pasamos a describir cada uno de los escenarios.

### 7.2. Experimentación con alumnos simulados

Para la evaluación del algoritmo de diagnóstico de cada red, se han utilizado alumnos simulados. La simulación [38] permite evaluar el algoritmo sin necesidad de definir un test para una asignatura concreta y de disponer de un grupo de alumnos a los que aplicar dicho test.

En el presente trabajo, el funcionamiento de un alumno simulado es el siguiente: sean  $C_1, \dots, C_n$  los conceptos de la red de diagnóstico. Dado un valor  $s \in (0, 1)$ , se define el alumno simulado tipo  $s$  como un alumno que conoce el 100 %  $s$  de los conceptos. Para este alumno, dado el valor de  $s$  obtenido de forma aleatoria, se genera también aleatoriamente el conjunto de conceptos que inicialmente va a conocer.

Una vez generado el alumno simulado, se utiliza cada modelo para calcular las probabilidades de responder correctamente a cada pregunta que se selecciona para simular su respuesta. Dicha probabilidad se utilizará para simular el comportamiento del alumno en el test de la siguiente forma: supongamos que la probabilidad de responder correctamente a una pregunta  $P$  es  $p$ . Fijado un valor umbral próximo a 1, comparamos esta probabilidad  $p$  con el valor umbral. Si  $p \geq \text{umbral}$  se considera que el alumno ha respondido correctamente la pregunta, y si  $p < \text{umbral}$ , que la ha respondido incorrectamente.

## CAPÍTULO 7. EXPERIMENTACIÓN DEL MODELO Y RESULTADOS

Tras la respuesta simulada del alumno, el algoritmo de diagnóstico la utiliza para actualizar las probabilidades de los conceptos y elige la siguiente pregunta para presentarle al alumno.

En esta simulación, se ha utilizado una red bayesiana de 100 preguntas, que denotamos por  $P_1, P_2, \dots, P_{100}$  y 33 conceptos,  $C_1, C_2, \dots, C_{33}$ .

A continuación, mostramos las relaciones entre conceptos y preguntas. Como puede observarse, cada pregunta está relacionada con uno, dos o tres conceptos. Para una mayor legibilidad, las preguntas se han agrupado de 20 en 20.

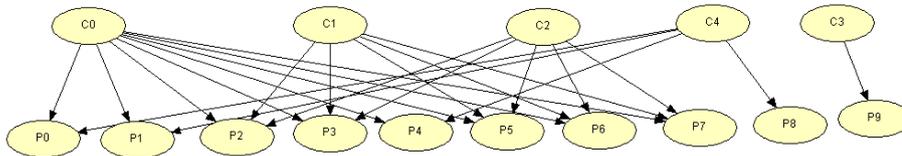


Figura 7.1: Preguntas de la 0 a la 9

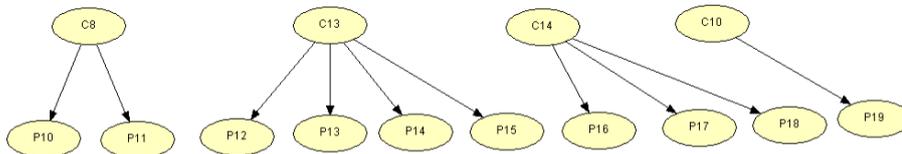


Figura 7.2: Preguntas de la 10 a la 19

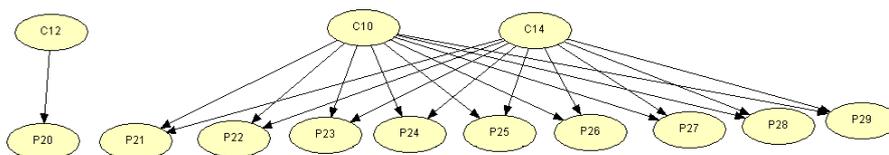


Figura 7.3: Preguntas de la 20 a la 29

En las figuras 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8, 7.9 y 7.10, puede observarse cómo la red de pruebas utilizada no sigue un criterio específico a la hora de asociar las distintas preguntas y conceptos. Se trata de una red del mundo real en la que puede haber preguntas que comparten conceptos y también conceptos que están representados por una única pregunta.

Cada una de las preguntas tiene 4 respuestas posibles, por tanto, un factor de azar de  $c = 1/4$ . Para todas las preguntas hemos escogido como valores

## 7.2. EXPERIMENTACIÓN CON ALUMNOS SIMULADOS

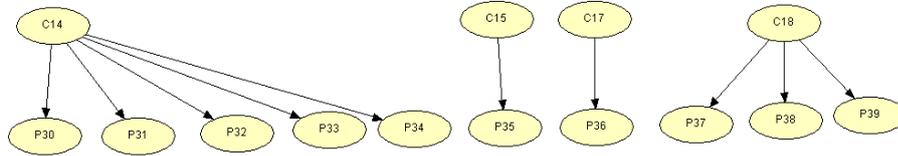


Figura 7.4: Preguntas de la 30 a la 39

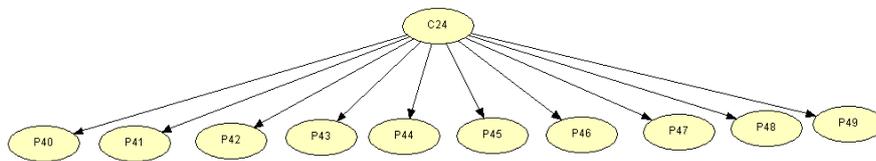


Figura 7.5: Preguntas de la 40 a la 49

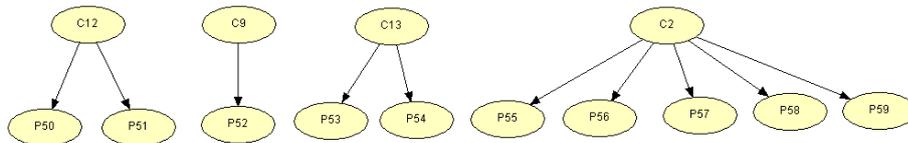


Figura 7.6: Preguntas de la 50 a la 59

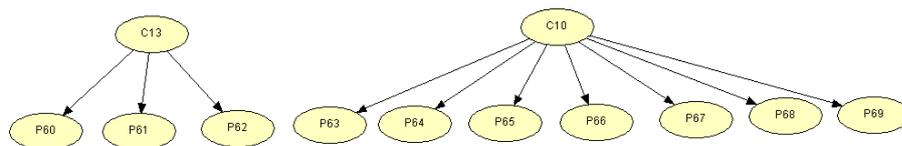


Figura 7.7: Preguntas de la 60 a la 69

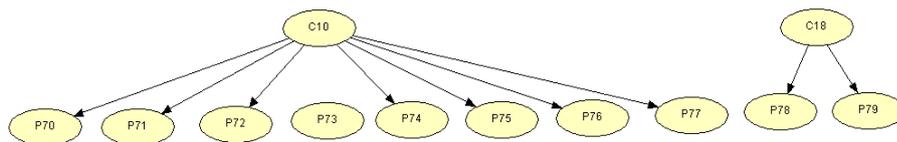


Figura 7.8: Preguntas de la 70 a la 79

## CAPÍTULO 7. EXPERIMENTACIÓN DEL MODELO Y RESULTADOS

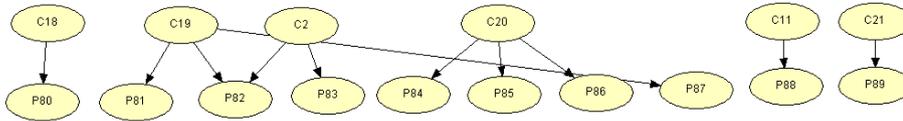


Figura 7.9: Preguntas de la 80 a la 89

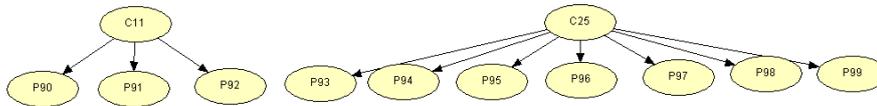


Figura 7.10: Preguntas de la 90 a la 99

de factor de discriminación  $a = 0,3$  y los factores  $b$  se corresponden con el nivel de dificultad de las preguntas. El algoritmo de diagnóstico selecciona las preguntas siguiendo el criterio de utilidad máxima de la pregunta.

Para comparar los resultados de simulación de ambas redes, vamos a definir el parámetro *Tipo de alumno* que se define como el número de conceptos que inicialmente conoce el alumno siendo el valor de este parámetro un porcentaje sobre el total de conceptos. De este modo, tenemos los siguientes valores para *Tipo de alumno*: 0.2 (el alumno conoce el 20 % de los conceptos), 0.4 (el alumno conoce el 40 % de los conceptos), 0.6 (el alumno conoce el 60 % de los conceptos), 0.8 (el alumno conoce el 80 % de los conceptos), 0.9 (el alumno conoce el 90 % de los conceptos) y 1 (el alumno conoce el 100 % de los conceptos).

En los experimentos se generaron 50 alumnos de los 6 tipos anteriores con un valor *Tipo de alumno* asignado a cada uno de forma aleatoria. Es decir, no se ha tenido el mismo número de alumnos de *Tipo de alumno* 0.2 que de cualquier otro tipo.

Cada alumno se sometió a la simulación de las dos redes bayesianas, obteniéndose los resultados mostrados en la siguiente figura:

## 7.2. EXPERIMENTACIÓN CON ALUMNOS SIMULADOS

Tipo de alumno	Nº conceptos dominados Millán	Nº conceptos dominados propuesto	Nº conceptos no dominados Millán	Nº conceptos no dominados	Nº no diagnosticados Millán	Nº no diagnosticados propuesto
0.2	7	7	26	26	0	0
0.2	11	11	22	22	0	0
0.2	13	13	20	20	0	0
0.4	15	16	18	17	0	0
0.8	12	12	21	21	0	0
0.8	15	15	18	18	0	0
0.2	15	15	18	18	0	0
0.2	16	16	17	17	0	0
0.4	16	16	17	17	0	0
0.4	21	22	12	11	0	0
0.8	13	13	20	20	0	0
0.2	15	15	18	18	0	0
0.8	15	15	18	18	0	0
0.8	18	19	15	14	0	0
0.2	17	17	16	16	0	0
0.4	19	19	14	14	0	0
0.2	23	23	10	10	0	0
0.2	25	25	8	8	0	0
0.6	17	17	16	16	0	0
0.2	19	19	14	14	0	0
0.4	19	19	14	14	0	0
0.2	22	22	11	11	0	0
0.2	23	23	10	10	0	0
0.6	17	17	16	16	0	0
0.4	19	19	14	14	0	0
0.4	21	21	12	12	0	0
0.8	16	17	17	16	0	0
0.4	19	19	14	14	0	0
0.2	18	19	15	14	0	0
0.2	18	18	15	15	0	0
0.2	21	21	12	12	0	0
0.2	22	22	11	11	0	0
0.4	22	22	11	11	0	0
0.4	23	23	10	10	0	0
0.2	22	22	11	11	0	0
0.6	19	19	14	14	0	0
0.8	19	19	14	14	0	0
0.2	17	17	16	16	0	0
0.6	13	13	20	20	0	0
0.2	16	16	17	17	0	0
0.8	17	17	16	16	0	0
0.6	15	15	18	18	0	0
0.4	18	18	15	15	0	0
0.2	18	18	15	15	0	0
0.8	14	14	19	19	0	0
0.6	15	15	18	18	0	0
0.2	19	19	14	14	0	0
0.2	21	21	12	12	0	0
0.4	17	17	16	16	0	0
0.6	16	16	17	17	0	0

Figura 7.11: Número de conceptos dominados, no dominados y no diagnosticados por ambas redes

Las medidas que nos interesan de estos resultados son: el número de conceptos dominados por cada alumno, el número de conceptos no dominados y finalmente el número de conceptos no diagnosticados.

Vamos a analizar cada medida una a una.

En cuanto al número de conceptos no dominados, puede observarse cómo es menor en el “modelo de red bayesiana con pesos propuesto” que en el “modelo de red bayesiana de Millán”, y en cuanto al número de conceptos dominados por el alumno, es mayor en el “modelo de red bayesiana con pesos propuesto” que en el “modelo de red bayesiana de Millán”. A priori, dado que a nosotros nos debe preocupar más los conceptos clasificados como no dominados, que era uno de los objetivos de nuestro trabajo, parece que el “modelo de red bayesiana de Millán” nos proporciona más datos acerca de los conceptos no

## CAPÍTULO 7. EXPERIMENTACIÓN DEL MODELO Y RESULTADOS

---

dominados, deducción que no tiene por qué ser cierta. Por tanto, para poder evaluar la efectividad de cada una de las dos redes, habrá que analizar qué red se acerca más al número de conceptos dominados y no dominados *reales* del alumno. Para poder cuantificar la proximidad de cada red a la realidad, trabajamos con el error estándar que ya se empleaba en [38] para medir la diferencia entre el número de conceptos no dominados diagnosticados por cada red y el número de conceptos no dominados realmente por el alumno.

Para calcular el número de conceptos no dominados en el mundo real por cada alumno, se parte del número de preguntas que ha respondido correctamente el alumno en la simulación, de tal modo que si responde correctamente todas las preguntas que influyen en un mismo concepto, se deducirá que este concepto es dominado por el alumno. Entonces, en el momento que el alumno no responde correctamente una pregunta relacionada con el concepto, ya se deduce que no domina el concepto. Para la realización de los experimentos con alumnos simulados, se ha pasado un test de 10 preguntas a cada uno de los 50 alumnos para así poder comparar los resultados de los conceptos dominados/no dominados por cada alumno simulado con los que dominaría/no dominaría un alumno con el perfil de alumno que indica su atributo *Tipo de alumno*.

Siguiendo los resultados de la figura 7.11, tenemos que en el “modelo de red bayesiana de Millán” para por ejemplo, el primer alumno de tipo 0.4, el número de conceptos diagnosticados como “no dominados” es de 18 y de 17 en el “modelo de red bayesiana con pesos propuesto”. En la simulación se tiene que el alumno ha respondido correctamente a 4 preguntas sobre 10. Teniendo en cuenta que las preguntas respondidas correctamente tienen asociados conceptos que no están en el conjunto de los conceptos inicialmente conocidos por el alumno y que estas 4 preguntas (0,1,2,3) se refieren a los mismos 4 conceptos (0,4,1,2), podemos restar al conjunto de conceptos inicialmente no conocidos por el alumno 4 conceptos para tener el número de conceptos reales no dominados por el alumno. El número de conceptos iniciales no dominados para un alumno de tipo 0.4 es el 60 % de 33 conceptos, que son 19.8, que restándoles 4 conceptos, es un total de 15.8 conceptos reales no dominados. El error estándar será:

- Error estándar en modelo de red bayesiana de Millán:  $|15.8 \text{ reales} - 18 \text{ simulados}| = 2.2$
- Error estándar en modelo de red bayesiana con pesos propuesto:  $|15.8 \text{ reales} - 17 \text{ simulados}| = 1.2$

Asimismo, para el mismo alumno de tipo 0.4 de la figura 7.11, tenemos que en el “modelo de red bayesiana de Millán”, el número de conceptos dominados

es de 15 y de 16 en el “modelo de red bayesiana con pesos propuesto”. En la simulación se tiene que el alumno ha respondido correctamente a 4 preguntas sobre 10. Teniendo en cuenta que las preguntas respondidas correctamente tienen asociados conceptos que no están en el conjunto de los conceptos inicialmente conocidos por el alumno y que estas 4 preguntas (0,1,2,3) se refieren a los mismos 4 conceptos (0,4,1,2), podemos sumar a este conjunto de conceptos inicialmente conocidos por el alumno 4 conceptos para tener el número de conceptos reales dominados por el alumno. El número de conceptos iniciales para un alumno de tipo 0.4 es el 40 % de 33 conceptos, que son 13.2, que sumándoles 4 conceptos, es un total de 17.2 conceptos reales dominados. El error estándar será:

- Error estándar en el modelo de red bayesiana de Millán:  $|17.2 \text{ reales} - 15 \text{ simulados}| = 2.2$
- Error estándar en modelo de red bayesiana con pesos propuesto:  $|17.2 \text{ reales} - 16 \text{ simulados}| = 1.2$

Según el valor del error estándar, el “modelo de red bayesiana de Millán” diagnostica con el mismo error el número de conceptos dominados y el número de conceptos no dominados. Por otro lado, el “modelo de red bayesiana con pesos propuesto” diagnostica con el mismo error estándar que el “modelo de red bayesiana de Millán” el número de conceptos dominados y el número de conceptos no dominados.

Para tener una mayor seguridad de que estas conclusiones sean las correctas, será fundamental realizar experimentos con alumnos reales, cuyos errores cometidos en el test se correspondan con su perfil de alumno. Estos experimentos se desarrollan en el siguiente apartado.

### 7.3. Experimentación con alumnos reales

Además de realizar los experimentos con alumnos simulados, ha sido posible utilizar alumnos reales, por disponer de una aplicación web, que presenta visualmente las preguntas al usuario y cuya lógica interna es la misma que en el caso simulado de las dos redes, dándole al usuario la opción de ejecutar el test siguiendo el mecanismo del “modelo de red bayesiana de Millán” o siguiendo el algoritmo del “modelo de red bayesiana con pesos propuesta”. Remitimos al lector al detalle de los algoritmos de diagnóstico utilizados por cada una de las dos redes y que se han utilizado tanto en el caso simulado (aplicación batch) como en el real (aplicación interactiva), que están en el Anexo I.

## CAPÍTULO 7. EXPERIMENTACIÓN DEL MODELO Y RESULTADOS

---

En mi opinión, es muy interesante realizar experimentos con alumnos reales porque en un entorno real, es como se puede examinar el funcionamiento de un sistema en base a los aciertos o errores de los alumnos con el test.

Los experimentos se han realizado con una clase de 6 alumnos reales, que son estudiantes del primer ciclo de grado superior de Formación Profesional titulado Administración de Sistemas Informáticos en Red. Cada alumno tiene asociado un perfil según el número de conceptos dominados inicialmente por el alumno. Este perfil se corresponde con el atributo *Tipo de alumno* de un alumno en el caso simulado y se ha conseguido, proporcionando a cada alumno la lectura de material del módulo de Sistemas Operativos adaptado a cada nivel. Como el material debe ser entendible y coherente con los conceptos tratados, en cada perfil de un nivel superior al inmediato anterior, se va ampliando y enriqueciendo. Lo que quiero decir es que en la confección del material no se ha buscado que los conceptos iniciales que un alumno conoce sean más sencillos que los de otro de un perfil superior, aunque sí se supone que un material de estudio didáctico avanza hacia conceptos cada vez de mayor dificultad. Ninguno de los alumnos tenía conceptos previos de Sistemas Operativos antes de la lectura del material de estudio adaptado a cada uno de ellos.

En el experimento, a cada alumno se le suministran primero 10 ítems, después 20 ítems, 30 ítems y por último 40 ítems, y se miden el número de conceptos dominados, número de conceptos no dominados y número de conceptos no diagnosticados, cuyos valores se muestran en las siguientes tablas:

Tipo de alumno	D.prop	D.Mill	ND.prop	ND.Mill	ND.prop	ND.Mill
0.1	3	2	30	31	0	0
0.9	2	1	31	32	0	0
0.8	2	1	31	31	0	0
0.6	2	2	31	32	0	0
0.4	2	1	31	32	0	0
0.2	1	1	31	32	1	0

Tabla 7.1: Resultados de 6 alumnos reales tras administrarles 10 ítems

### 7.3. EXPERIMENTACIÓN CON ALUMNOS REALES

Tipo de alumno	D.prop	D.Mill	ND.prop	ND.Mill	ND.prop	ND.Mill
0.1	5	3	28	30	0	0
0.9	4	5	29	28	0	0
0.8	6	5	26	28	1	0
0.6	5	5	28	28	0	0
0.4	3	4	30	29	0	0
0.2	3	3	30	30	0	0

Tabla 7.2: Resultados de 6 alumnos reales tras administrarles 20 ítems

Tipo de alumno	D.prop	D.Mill	ND.prop	ND.Mill	ND.prop	ND.Mill
0.1	8	9	25	24	0	0
0.9	6	6	27	27	0	0
0.8	7	7	26	26	0	0
0.6	6	6	27	27	0	0
0.4	5	5	28	28	0	0
0.2	4	3	29	30	0	0

Tabla 7.3: Resultados de 6 alumnos reales tras administrarles 30 ítems

Tipo de alumno	D.prop	D.Mill	ND.prop	ND.Mill	ND.prop	ND.Mill
0.1	12	12	21	21	0	0
0.9	10	10	23	23	0	0
0.8	11	9	22	24	0	0
0.6	10	12	23	21	0	0
0.4	8	6	25	27	0	0
0.2	5	6	28	27	0	0

Tabla 7.4: Resultados de 6 alumnos reales tras administrarles 40 ítems

Para poder decidir qué red (“modelo de red bayesiana de Millán” o “modelo de red bayesiana con pesos propuesto”) diagnostica mejor el número de conceptos que el alumno domina, los que no domina y de los que aún no se tiene suficiente información, es preciso medir la precisión de estimación de cada modelo de red bayesiana. Como medida de evaluación, utilizamos el error estándar, que se define como la diferencia entre el valor real y el valor estimado por cada red, en cada uno de los dos parámetros: número de conceptos dominados y número de conceptos no dominados. Del número de conceptos no diagnosticados no calculamos el error estándar, puesto que en el mundo real, un concepto estará catalogado siempre como no dominado o como dominado, aunque este dato del número de conceptos no diagnosticados por las redes, puede ser útil para sacar conclusiones, como haremos posteriormente.

## CAPÍTULO 7. EXPERIMENTACIÓN DEL MODELO Y RESULTADOS

---

De este modo, calcularemos el error estándar en el número de conceptos dominados en la realidad por el alumno y el número de conceptos que cada red estima como dominados. En concreto, el error estándar será la diferencia del porcentaje del número de alumnos dominados realmente por el alumno y el porcentaje de conceptos dominados por el mismo. La razón de restar los porcentajes en lugar del número de conceptos, es porque en una serie de preguntas de test que se presentan al alumno, no intervienen todos los conceptos del sistema y por su parte, cada red hace una estimación del número de conceptos dominados sobre el total de conceptos del sistema. El error estándar será, por tanto, una cantidad comprendida entre 0 y 1, y se calculará para cada red, el error estándar en el porcentaje de conceptos dominados, el error estándar en el porcentaje de conceptos no dominados, y finalmente, el error estándar en el número de conceptos no diagnosticados.

Vamos a mostrar los resultados del error estándar en cada modelo para la muestra completa de 40 ítems. Hay que destacar, que cada red elige como ítem siguiente a mostrar en el test, aquél cuya probabilidad del ítem sea máxima dados todos y cada uno de los conceptos del sistema. Esta probabilidad de pregunta, ya hemos visto que tiene un valor distinto en cada red, por lo que la secuencia de preguntas que administra cada red es distinta. A continuación mostramos los resultados clasificados por *Tipo de alumno* y después los explicamos.

Para un alumno de Tipo 1.0, la secuencia de ítems que calcula el “modelo de red bayesiana con pesos propuesto” comparte un total de 22 conceptos distintos. En un alumno con *Tipo de alumno* igual a 1.0, 22/22 es el porcentaje de alumnos realmente conocidos por el alumno, es decir, el 100% de los conceptos de los ítems que se le han presentado. El “modelo de red bayesiana con pesos propuesto” estima 12 conceptos como dominados, con lo que 12/33 es el porcentaje de conceptos dominados que estima el modelo. Tenemos, por tanto: El error estándar en el número de conceptos dominados es:

$$\begin{aligned} \text{Error estándar conceptos dominados} &= |21/21 \text{ real} - 12/33 \text{ estimado}| \\ &= |1 \text{ real} - 0.36 \text{ estimado}| = 0.64. \end{aligned}$$

En cuanto al número de conceptos no dominados, el “modelo de red bayesiana con pesos propuesto” estima 21/33 como el porcentaje de conceptos no dominados. El error estándar en el número de conceptos no dominados es:

$$\begin{aligned} \text{Error estándar conceptos no dominados} &= |0 \text{ real} - 21/33 \text{ estimado}| \\ &= |0 \text{ real} - 0.63 \text{ estimado}| = 0.63. \end{aligned}$$

Si a un alumno con *Tipo de alumno* igual a 1.0, le presentamos 40 ítems calculados con el “modelo de red bayesiana de Millán”, obtenemos que la secuencia de ítems comparte un total de 22 conceptos distintos. El “modelo

### 7.3. EXPERIMENTACIÓN CON ALUMNOS REALES

Nº ítems	Tipo de alumno		
40	1.0	conceptos dominados	conceptos no dominados
Preguntas modelo red bayesiana propuesto	0-1-3-5-7-21-23-25-26-82-97-9-11-13-15-17-18-30-32-34-40-42-44-46-48-50-52-67-69-71-73-75-77-79-81-84-86-88-90-91	0.64	0.63
Preguntas modelo red bayesiana de Millán	0-2-4-6-20-22-23-25-27-29-95-97-8-10-12-14-16-18-30-32-34-38-40-42-44-46-48-50-52-66-67-69-71-75-78-80-83-85-87-89	0.64	0.63

Tabla 7.5: Error estándar para un alumno con Tipo de alumno 1.0 tras 40 ítems

de red bayesiana de Millán” estima 12/33 como el porcentaje de conceptos dominados. El error estándar en el número de conceptos dominados es:

$$\begin{aligned} \text{Error estándar conceptos dominados} &= |1 \text{ real} - 12/33 \text{ estimado}| \\ &= |1 \text{ real} - 0.36 \text{ estimado}| = 0.64. \end{aligned}$$

En cuanto al número de conceptos no dominados, el “modelo de bayesiana de Millán” estima 21/33 como el porcentaje de conceptos no dominados por el alumno. El error estándar en el número de conceptos dominados es:

$$\begin{aligned} \text{Error estándar conceptos no dominados} &= |0 \text{ real} - 21/33 \text{ estimado}| \\ &= |0 \text{ real} - 0.64 \text{ estimado}| = 0.64. \end{aligned}$$

Para un alumno de Tipo 0.9, la secuencia de ítems que calcula el “modelo de red bayesiana con pesos propuesto” comparte un total de 23 conceptos distintos. En un alumno con *Tipo de alumno* igual a 0.9, el 90%23, que es 20.7, son los conceptos realmente conocidos por un alumno de este tipo. El “modelo de red bayesiana con pesos propuesto” estima 10 conceptos como

CAPÍTULO 7. EXPERIMENTACIÓN DEL MODELO Y RESULTADOS

Nº ítems	Tipo de alumno		
40	0.9	conceptos dominados	conceptos no dominados
Preguntas modelo red bayesiana propuesto	0-2-4-6-8-21-23-25-27-95-97-12-14-16-18-30-32-34-38-41-43-45-47-49-53-65-67-69-73-75-77-79-81-84-86-88-90-92-94-90	0.60	0.59
Preguntas modelo red bayesiana de Millán	0-2-3-5-7-21-23-25-27-29-95-10-12-14-16-18-30-32-34-36-38-40-41-43-45-47-49-51-65-67-71-74-76-78-80-83-85-86-88-90	0.60	0.59

Tabla 7.6: Error estándar para un alumno con Tipo de alumno 0.9 tras 40 ítems

dominados, con lo que  $10/33$  es el porcentaje de conceptos dominados que estima el modelo. Tenemos, por tanto: El error estándar en el número de conceptos dominados es:

$$\begin{aligned} \text{Error estándar conceptos dominados} &= |20.7/23 \text{ real} - 10/33 \text{ estimado}| \\ &= |0.9 \text{ real} - 0.3 \text{ estimado}| = 0.60. \end{aligned}$$

En cuanto al número de conceptos no dominados, son  $2.3/23$  los conceptos que el alumno realmente no domina. El “modelo de red bayesiana con pesos propuesto” estima  $23/33$  como el porcentaje de conceptos no dominados. El error estándar en el número de conceptos no dominados es:

$$\begin{aligned} \text{Error estándar conceptos no dominados} &= |2.3/23 \text{ real} - 23/33 \text{ estimado}| \\ &= |0.1 \text{ real} - 0.69 \text{ estimado}| = 0.59. \end{aligned}$$

Si a un alumno con *Tipo de alumno* igual a 0.9, le presentamos 40 ítems calculados con el “modelo de red bayesiana de Millán”, obtenemos que la secuencia de ítems comparte un total de 27 conceptos distintos, de los cuales conoce el 90 %, que son 24.3 conceptos. El “modelo de red bayesiana de

### 7.3. EXPERIMENTACIÓN CON ALUMNOS REALES

---

Millán” estima 10/33 como el porcentaje de conceptos dominados. El error estándar en el número de conceptos dominados es:

$$\begin{aligned} \text{Error estándar conceptos dominados} &= |24.3/27 \text{ real} - 10/33 \text{ estimado}| \\ &= |0.9 \text{ real} - 0.3 \text{ estimado}| = 0.60. \end{aligned}$$

En cuanto al número de conceptos no dominados, que son 2.7/27, el “modelo de bayesiana de Millán” estima 23/33 como el porcentaje de conceptos no dominados por el alumno. El error estándar en el número de conceptos dominados es:

$$\begin{aligned} \text{Error estándar conceptos no dominados} &= |2.7/27 \text{ real} - 23/33 \text{ estimado}| \\ &= |0.1 \text{ real} - 0.69 \text{ estimado}| = 0.59. \end{aligned}$$

Hasta el momento, no se ha apreciado ninguna diferencia en cuanto al error estándar obtenido en cada una de las dos redes.

Nº ítems	Tipo de alumno	conceptos dominados	conceptos no dominados
40	0.8		
Preguntas modelo red bayesiana propuesto	0-2-4-6-8-21-23-25-27-29-95-9-11-13-15-17-31-35-37-39-41-43-45-47-49-51-65-67-69-71-74-76-78-80-83-85-89-91-93-96	0.47	0.46
Preguntas modelo red bayesiana de Millán	0-2-4-6-20-22-24-26-28-30-82-97-11-13-15-17-19-32-34-36-38-40-42-44-46-48-50-52-66-68-70-74-76-78-79-81-84-88-90-91	0.53	0.52

Tabla 7.7: Error estándar para un alumno con Tipo de alumno 0.8 tras 40 ítems

## CAPÍTULO 7. EXPERIMENTACIÓN DEL MODELO Y RESULTADOS

---

Para un alumno de Tipo 0.8, la secuencia de ítems que calcula el “modelo de red bayesiana con pesos propuesto” comparte un total de 21 conceptos distintos. En un alumno con *Tipo de alumno* igual a 0.8, el 80 %21, que es 16.8, son los conceptos realmente conocidos por un alumno de este tipo. El “modelo de red bayesiana con pesos propuesto” estima 11 conceptos como dominados, con lo que 11/33 es el porcentaje de conceptos dominados que estima el modelo. Tenemos, por tanto: El error estándar en el número de conceptos dominados es:

$$\begin{aligned} \text{Error estándar conceptos dominados} &= |16.8/21 \text{ real} - 11/33 \text{ estimado}| \\ &= |0.8 \text{ real} - 0.33 \text{ estimado}| = 0.47. \end{aligned}$$

En cuanto al número de conceptos no dominados, son 4.2/21 los conceptos que el alumno realmente no domina. El “modelo de red bayesiana con pesos propuesto” estima 22/33 como el porcentaje de conceptos no dominados. El error estándar en el número de conceptos no dominados es:

$$\begin{aligned} \text{Error estándar conceptos no dominados} &= |4.2/21 \text{ real} - 22/33 \text{ estimado}| \\ &= |0.3 \text{ real} - 0.66 \text{ estimado}| = 0.46. \end{aligned}$$

Si a un alumno con *Tipo de alumno* igual a 0.9, le presentamos 40 ítems calculados con el “modelo de red bayesiana de Millán”, obtenemos que la secuencia de ítems comparte un total de 21 conceptos distintos, de los cuales conoce el 80 %, que son 16.8 conceptos. El “modelo de red bayesiana de Millán” estima 9/33 como el porcentaje de conceptos dominados. El error estándar en el número de conceptos dominados es:

$$\begin{aligned} \text{Error estándar conceptos dominados} &= |16.8/21 \text{ real} - 9/33 \text{ estimado}| \\ &= |0.8 \text{ real} - 0.27 \text{ estimado}| = 0.53. \end{aligned}$$

En cuanto al número de conceptos no dominados, que son 4.2/21, el “modelo de bayesiana de Millán” estima 24/33 como el porcentaje de conceptos no dominados por el alumno. El error estándar en el número de conceptos dominados es:

$$\begin{aligned} \text{Error estándar conceptos no dominados} &= |4.2/21 \text{ real} - 24/33 \text{ estimado}| \\ &= |0.2 \text{ real} - 0.72 \text{ estimado}| = 0.52. \end{aligned}$$

Para un alumno de Tipo 0.6, la secuencia de ítems que calcula el “modelo de red bayesiana con pesos propuesto” comparte un total de 22 conceptos distintos. En un alumno con *Tipo de alumno* igual a 0.6, el 60 %22, que es 13.8, son los conceptos realmente conocidos por un alumno de este tipo. El “modelo de red bayesiana con pesos propuesto” estima 10 conceptos como

### 7.3. EXPERIMENTACIÓN CON ALUMNOS REALES

Nº ítems	Tipo de alumno		
40	0.6	conceptos dominados	conceptos no dominados
Preguntas modelo red bayesiana propuesto	0-2-6-8-21-23-25-27-29-95-9-11-13-15-17-19-31-33-35-37-39-41-43-45-47-49-51-67-69-73-75-77-79-81-84-86-88-90-92-94	0.47	0.46
Preguntas modelo red bayesiana de Millán	0-2-4-6-8-21-23-25-27-95-9-11-13-15-17-19-31-33-35-37-38-40-42-44-46-48-50-52-66-68-74-76-78-80-83-85-87-89-91-93	0.53	0.52

Tabla 7.8: Error estándar para un alumno con Tipo de alumno 0.6 tras 40 ítems

dominados, con lo que  $10/33$  es el porcentaje de conceptos dominados que estima el modelo. Tenemos, por tanto: El error estándar en el número de conceptos dominados es:

**Error estándar conceptos dominados** =  $|13.8/22 \text{ real} - 10/33 \text{ estimado}|$

$$= |0.6 \text{ real} - 0.30 \text{ estimado}| = 0.30.$$

En cuanto al número de conceptos no dominados, son  $8.8/22$  los conceptos que el alumno realmente no domina. El “modelo de red bayesiana con pesos propuesto” estima  $23/33$  como el porcentaje de conceptos no dominados. El error estándar en el número de conceptos no dominados es:

**Error estándar conceptos no dominados** =  $|8.8/22 \text{ real} - 23/33 \text{ estimado}|$

$$= |0.4 \text{ real} - 0.69 \text{ estimado}| = 0.29.$$

Si a un alumno con *Tipo de alumno* igual a 0.6, le presentamos 40 ítems calculados con el “modelo de red bayesiana de Millán”, obtenemos que la secuencia de ítems comparte un total de 26 conceptos distintos, de los cuales conoce el 60 %, que son 15.6 conceptos. El “modelo de red bayesiana de

## CAPÍTULO 7. EXPERIMENTACIÓN DEL MODELO Y RESULTADOS

---

Millán” estima 12/33 como el porcentaje de conceptos dominados. El error estándar en el número de conceptos dominados es:

$$\begin{aligned} \text{Error estándar conceptos dominados} &= |15.6/26 \text{ real} - 12/33 \text{ estimado}| \\ &= |0.6 \text{ real} - 0.36 \text{ estimado}| = 0.24. \end{aligned}$$

En cuanto al número de conceptos no dominados, que son 10.4/26, el “modelo de bayesiana de Millán” estima 21/33 como el porcentaje de conceptos no dominados por el alumno. El error estándar en el número de conceptos dominados es:

$$\begin{aligned} \text{Error estándar conceptos no dominados} &= |10.4/26 \text{ real} - 21/33 \text{ estimado}| \\ &= |0.4 \text{ real} - 0.63 \text{ estimado}| = 0.23. \end{aligned}$$

Nº ítems	Tipo de alumno	conceptos dominados	conceptos no dominados
40	0.4		
Preguntas modelo red bayesiana propuesto	0-2-3-5-7-21-23-25-83-97-8-10-12-14-16-30-32-34-38-40-42-44-46-48-50-52-66-68-70-74-76-77-79-81-85-87-89-91-93-96	0.16	0.15
Preguntas modelo red bayesiana de Millán	0-2-3-5-7-21-23-25-27-95-10-12-14-16-18-30-32-34-36-38-40-41-43-45-47-49-51-65-67-71-74-76-78-80-83-85-86-88-90-91	0.22	0.21

Tabla 7.9: Error estándar para un alumno con Tipo de alumno 0.4 tras 40 ítems

Para un alumno de Tipo 0.4, la secuencia de ítems que calcula el “modelo de red bayesiana con pesos propuesto” comparte un total de 25 conceptos

### 7.3. EXPERIMENTACIÓN CON ALUMNOS REALES

---

distintos. En un alumno con *Tipo de alumno* igual a 0.4, el 40 %25, que es 10, son los conceptos realmente conocidos por un alumno de este tipo. El “modelo de red bayesiana con pesos propuesto” estima 10 conceptos como dominados, con lo que  $8/33$  es el porcentaje de conceptos dominados que estima el modelo. Tenemos, por tanto: El error estándar en el número de conceptos dominados es:

$$\begin{aligned} \text{Error estándar conceptos dominados} &= |10/25 \text{ real} - 8/33 \text{ estimado}| \\ &= |0.4 \text{ real} - 0.24 \text{ estimado}| = 0.16. \end{aligned}$$

En cuanto al número de conceptos no dominados, son  $15/25$  los conceptos que el alumno realmente no domina. El “modelo de red bayesiana con pesos propuesto” estima  $25/33$  como el porcentaje de conceptos no dominados. El error estándar en el número de conceptos no dominados es:

$$\begin{aligned} \text{Error estándar conceptos no dominados} &= |15/25 \text{ real} - 25/33 \text{ estimado}| \\ &= |0.6 \text{ real} - 0.75 \text{ estimado}| = 0.15. \end{aligned}$$

Si a un alumno con *Tipo de alumno* igual a 0.4, le presentamos 40 ítems calculados con el “modelo de red bayesiana de Millán”, obtenemos que la secuencia de ítems comparte un total de 18 conceptos distintos, de los cuales conoce el 40 %, que son 7.2 conceptos. El “modelo de red bayesiana de Millán” estima  $6/33$  como el porcentaje de conceptos dominados. El error estándar en el número de conceptos dominados es:

$$\begin{aligned} \text{Error estándar conceptos dominados} &= |7.2/18 \text{ real} - 6/33 \text{ estimado}| \\ &= |0.4 \text{ real} - 0.18 \text{ estimado}| = 0.22. \end{aligned}$$

En cuanto al número de conceptos no dominados, que son  $10.8/18$ , el “modelo de bayesiana de Millán” estima  $27/33$  como el porcentaje de conceptos no dominados por el alumno. El error estándar en el número de conceptos dominados es:

$$\begin{aligned} \text{Error estándar conceptos no dominados} &= |10.8/18 \text{ real} - 27/33 \text{ estimado}| \\ &= |0.6 \text{ real} - 0.81 \text{ estimado}| = 0.21. \end{aligned}$$

Para un alumno de Tipo 0.2, la secuencia de ítems que calcula el “modelo de red bayesiana con pesos propuesto” comparte un total de 25 conceptos distintos. En un alumno con *Tipo de alumno* igual a 0.2, el 20 %25, que es 5, son los conceptos realmente conocidos por un alumno de este tipo. El “modelo de red bayesiana con pesos propuesto” estima 5 conceptos como dominados, con lo que  $5/33$  es el porcentaje de conceptos dominados que estima el modelo. Tenemos, por tanto: El error estándar en el número de conceptos dominados es:

CAPÍTULO 7. EXPERIMENTACIÓN DEL MODELO Y RESULTADOS

Nº ítems	Tipo de alumno		
40	0.2	conceptos dominados	conceptos no dominados
Preguntas modelo red bayesiana propuesto	0-2-4-6-20-22-24-26-28-83-97-9-11-13-15-17-32-34-38-40-42-44-46-48-50-52-66-68-70-74-76-78-80-84-86-88-89-91-93-96,	0.05	0.04
Preguntas modelo red bayesiana de Millán	1-3-5-7-21-23-25-27-29-95-8-10-12-14-16-32-38-40-42-44-45-47-49-51-53-67-69-71-73-75-77-79-81-84-86-88-89-91-93	0.02	0.01

Tabla 7.10: Error estándar para un alumno con Tipo de alumno 0.2

$$\begin{aligned} \text{Error estándar conceptos dominados} &= |5/25 \text{ real} - 5/33 \text{ estimado}| \\ &= |0.2 \text{ real} - 0.15 \text{ estimado}| = 0.05. \end{aligned}$$

En cuanto al número de conceptos no dominados, son 20/25 los conceptos que el alumno realmente no domina. El “modelo de red bayesiana con pesos propuesto” estima 18/33 como el porcentaje de conceptos no dominados. El error estándar en el número de conceptos no dominados es:

$$\begin{aligned} \text{Error estándar conceptos no dominados} &= |10/25 \text{ real} - 18/33 \text{ estimado}| \\ &= |0.8 \text{ real} - 0.84 \text{ estimado}| = 0.04. \end{aligned}$$

Si a un alumno con *Tipo de alumno* igual a 0.2, le presentamos 40 ítems calculados con el “modelo de red bayesiana de Millán”, obtenemos que la secuencia de ítems comparte un total de 22 conceptos distintos, de los cuales conoce el 20 %, que son 4.4 conceptos. El “modelo de red bayesiana de Millán” estima 6/33 como el porcentaje de conceptos dominados. El error estándar en el número de conceptos dominados es:

$$\begin{aligned} \text{Error estándar conceptos dominados} &= |4.4/22 \text{ real} - 6/33 \text{ estimado}| \\ &= |0.2 \text{ real} - 0.18 \text{ estimado}| = 0.02. \end{aligned}$$

### 7.3. EXPERIMENTACIÓN CON ALUMNOS REALES

---

En cuanto al número de conceptos no dominados, que son 17.6/22, el “modelo de bayesiana de Millán” estima 27/33 como el porcentaje de conceptos no dominados por el alumno. El error estándar en el número de conceptos dominados es:

$$\begin{aligned} \text{Error estándar conceptos no dominados} &= |17.6/22 \text{ real} - 27/33 \text{ estimado}| \\ &= |0.8 \text{ real} - 0.81 \text{ estimado}| = 0.01. \end{aligned}$$

Hasta aquí, hemos detallado el cálculo del error estándar en una ejecución de 40 ítems del test. Para poder comparar mejor el desempeño de una red con respecto a la otra y entender también su funcionamiento, será interesante conocer la evolución del valor del error estándar a lo largo de la ejecución de los 10 primeros ítems, 20 ítems, 30 ítems, hasta llegar a los 40 ítems. A continuación, mostramos los valores del error estándar obtenido tras ejecutar cada grupo de 10 ítems.

	10 ítems		20 ítems		30 ítems		40 ítems	
	domin.	no do.						
Modelo de red bayesiana propuesto	0.94	0.93	0.85	0.84	0.75	0.75	0.64	0.63
Modelo de red bayesiana de Millán	0.94	0.93	0.91	0.90	0.72	0.72	0.64	0.63

Tabla 7.11: Evolución del Error estándar para un alumno con Tipo de alumno 1.0

En la tabla 7.11, aparece el error estándar cometido por cada una de las dos redes bayesianas para un alumno con *Tipo de alumno* igual a 1.0, a lo largo de la ejecución de 10 ítems, 20 ítems, 30 ítems y 40 ítems del test. En los 10 primeros ítems, puede observarse cómo el error no varía entre los dos modelos de redes bayesianas, siendo el error estándar estimado por el “modelo de red bayesiana con pesos propuesto” de 0.94 en cuanto al número de conceptos dominados por el alumno, y de 0.93 en cuanto al número de conceptos no dominados. Después de la ejecución de 10 ítems más (20 ítems), puede observarse cómo el error es ligeramente mayor en el “modelo de red

## CAPÍTULO 7. EXPERIMENTACIÓN DEL MODELO Y RESULTADOS

bayesiana de Millán” que en el “modelo de red bayesiana con pesos propuesto”. Esta situación corresponde a un caso lógico por la forma de calcular las probabilidades ambas redes, como comentaremos más adelante, en el que el número de conceptos dominados por el alumno es algo mayor en el “modelo de red bayesiana con pesos propuesto” que en el “modelo de red bayesiana de Millán”. Tras 30 ítems, se aprecia que el error es por 3 centésimas mayor en el “modelo de red bayesiana con pesos propuesto” que en el “modelo de red bayesiana de Millán”. Este segundo caso, a diferencia del anterior, se corresponde con una situación en la que el número de conceptos dominados por el “modelo de red bayesiana de Millán” es algo mayor que por el “modelo de red bayesiana con pesos propuesto”. Este caso no es consistente con los valores que toman las probabilidades de los conceptos en ambas redes, como veremos, pero puede darse debido a la posibilidad de tener una secuencia de ítems diferente en ambas redes y donde pueda aparecer un ítem asociado con único concepto que nada más responderse correctamente incrementa en una unidad el número de conceptos dominados estimados por “el modelo de red bayesiana de Millán”. En cualquier caso, es más común el caso en el que es mayor el número de conceptos dominados por el “modelo de red bayesiana con pesos propuesto” que en el “modelo de red bayesiana de Millán”. Finalmente, después de 40 ítems, el error vuelve a ser el mismo en las dos redes.

	10 ítems		20 ítems		30 ítems		40 ítems	
	domin.	no do.						
Modelo de red bayesiana propuesto	0.84	0.83	0.77	0.77	0.72	0.71	0.60	0.59
Modelo de red bayesiana de Millán	0.87	0.86	0.75	0.74	0.72	0.71	0.60	0.59

Tabla 7.12: Evolución del Error estándar para un alumno con Tipo de alumno 0.9

En la tabla 7.12, aparece el error estándar cometido por cada una de las dos redes bayesianas para un alumno con *Tipo de alumno* igual a 0.9, a lo largo de la ejecución de 10 ítems, 20 ítems, 30 ítems y 40 ítems del test.

### 7.3. EXPERIMENTACIÓN CON ALUMNOS REALES

En este caso, el error cometido es el mismo en ambas redes, excepto tras 20 ítems, donde el error es ligeramente menor por 2 y 3 centésimas en el “modelo de red bayesiana de Millán” que en el “modelo de red bayesiana con pesos propuesto”. Otra vez es la situación en la que la secuencia de ítems produce un concepto dominado más en el primer modelo que en el segundo.

	10 ítems		20 ítems		30 ítems		40 ítems	
	domin.	no do.						
Modelo de red bayesiana propuesto	0.74	0.73	0.62	0.58	0.59	0.58	0.47	0.46
Modelo de red bayesiana de Millán	0.77	0.76	0.65	0.64	0.59	0.58	0.53	0.52

Tabla 7.13: Evolución del Error estándar para un alumno con Tipo de alumno 0.8

En la tabla 7.13, aparece el error estándar cometido por cada una de las dos redes bayesianas para un alumno con *Tipo de alumno* igual a 0.8, a lo largo de la ejecución de 10 ítems, 20 ítems, 30 ítems y 40 ítems del test. En este caso, siempre es o igual o ligeramente superior en el “modelo de red bayesiana de Millán” que en el “modelo de red bayesiana con pesos propuesto”.

En la tabla 7.14, aparece el error estándar cometido por cada una de las dos redes bayesianas para un alumno con *Tipo de alumno* igual a 0.6, a lo largo de la ejecución de 10 ítems, 20 ítems, 30 ítems y 40 ítems del test. Aquí el error es el mismo, excepto en el caso de 40 ítems, en el que el error vuelve a ser ligeramente superior en el “modelo de red bayesiana de Millán” que en el “modelo de red bayesiana con pesos propuesto” por la secuencia de ítems.

En la tabla 7.15, aparece el error estándar cometido por cada una de las dos redes bayesianas para un alumno con *Tipo de alumno* igual a 0.4, a lo largo

## CAPÍTULO 7. EXPERIMENTACIÓN DEL MODELO Y RESULTADOS

	10 ítems		20 ítems		30 ítems		40 ítems	
	domin.	no do.						
Modelo de red bayesiana propuesto	0.54	0.53	0.45	0.44	0.42	0.41	0.30	0.29
Modelo de red bayesiana de Millán	0.54	0.53	0.45	0.44	0.42	0.41	0.24	0.23

Tabla 7.14: Evolución del Error estándar para un alumno con Tipo de alumno 0.6

	10 ítems		20 ítems		30 ítems		40 ítems	
	domin.	no do.						
Modelo de red bayesiana propuesto	0.54	0.53	0.45	0.44	0.42	0.41	0.30	0.29
Modelo de red bayesiana de Millán	0.54	0.53	0.45	0.44	0.42	0.41	0.24	0.23

Tabla 7.15: Evolución del Error estándar para un alumno con Tipo de alumno 0.4

de la ejecución de 10 ítems, 20 ítems, 30 ítems y 40 ítems del test. Aquí puede observarse que el error es igual o ligeramente superior en el “modelo de red bayesiana con pesos propuesto” que en el “modelo de red bayesiana de Millán”, excepto tras 20 ítems, donde volvemos a tener un caso en el que el “modelo de red bayesiana con pesos propuesto” tiene un error ligeramente mayor que en el “modelo de red bayesiana de Millán”.

En la tabla 7.16, aparece el error estándar cometido por cada una de las dos redes bayesianas para un alumno con *Tipo de alumno* igual a 0.2, a lo largo de la ejecución de 10 ítems, 20 ítems, 30 ítems y 40 ítems del test. En este

### 7.3. EXPERIMENTACIÓN CON ALUMNOS REALES

	10 ítems		20 ítems		30 ítems		40 ítems	
	domin.	no do.						
Modelo de red bayesiana propuesto	0.17	0.13	0.11	0.10	0.08	0.07	0.05	0.04
Modelo de red bayesiana de Millán	0.17	0.16	0.11	0.10	0.11	0.10	0.02	0.01

Tabla 7.16: Evolución del Error estándar para un alumno con Tipo de alumno 0.2

último caso, puede observarse cómo el error cometido es igual o ligeramente superior en el “modelo de red bayesiana de Millán” que en el “modelo de red bayesiana con pesos propuesto”.

Antes de exponer las conclusiones extraídas de este estudio, y que justifican este trabajo y el uso del “modelo de red bayesiana con pesos propuesto” en lugar del “modelo de red bayesiana de Millán”, vamos a explicar los tres casos que hemos citado previamente que pueden darse en el cálculo del error estándar. Se trata de 3 casos:

- **Caso 1: Error estándar igual en los dos modelos:** Este caso es la situación en la que el error estándar es el mismo en las dos redes, tanto en la estimación del número de conceptos dominados por el alumno como en el número de conceptos no dominados. Después de analizar la secuencia de ítems y de conceptos implicados en las dos redes, este caso se ha observado que se da cuando ambas redes diagnostican el mismo número de conceptos dominados y el mismo número de conceptos no dominados. Esto hemos analizado, que se produce también cuando la secuencia de ítems en ambas redes es muy similar en cuanto a los ítems presentados.
- **Caso 2: Error estándar superior en el modelo de Millán que en el modelo propuesto:** Este caso es el que nos sirve para justificar la consideración del “modelo de red bayesiana con pesos propuesto” sobre el “modelo de red bayesiana de Millán” y el que parece más consistente tras examinar los valores que van tomando las probabilidades de los conceptos en ambas redes tras la ejecución de los ítems. Estas probabilidades las mostramos a continuación.

- **Caso 3: Error estándar superior en el modelo propuesto que en el modelo de Millán:** Este caso lo hemos catalogado como un caso excepcional que puede deberse a dos motivos fundamentales: en primer lugar, por asignar a los conceptos del ítem unos valores de los pesos iguales o que el peso del concepto menos importante de la pregunta sea un valor muy bajo, y en segundo lugar, cuando hay muchas diferencias en la secuencia de ítems ejecutada por cada red. Puede suceder, que al tratarse de una aplicación web, haya que recargar, en alguna ocasión una página y se muestre una pregunta distinta, que por casualidad, esté asociada con un único concepto, y que al responderse correctamente, aumente en una unidad el número de conceptos dominados por el “modelo de red bayesiana de Millán” que por el “modelo de red bayesiana con pesos propuesto”. También, como comentaremos en la sección 8.2.1, en las aplicaciones web los objetos que se almacenan en sesión a veces presentan problemas, no eliminándose bien entre una sesión de usuario y la siguiente. Tal es el caso de la secuencia de preguntas mostradas en cada red que se almacenan en la sesión. El primer motivo lo analizamos a continuación.

Con el fin de entender mejor estos tres casos, se muestran los valores de las probabilidades que van tomando los conceptos de los ítems implicados en las dos redes.

Se ha escogido un ejemplo en el que las dos redes calculan la misma secuencia de ítems. Puede verse, por ejemplo, para el ítem P0 del banco de preguntas, cómo la probabilidad del concepto C0 es de 0.067 en el “modelo de red bayesiana con pesos propuesto” y de 0.003 en el “modelo de red bayesiana de Millán”. Este caso donde la probabilidad del concepto es mayor en el “modelo de red bayesiana con pesos propuesto” que en el “modelo de red bayesiana de Millán” es el que hemos observado que se da en la mayoría de los casos a lo largo de la ejecución de los ítems. En esta situación, el número de conceptos diagnosticados como dominados por el “modelo de red bayesiana con pesos propuesto” es mayor que en el “modelo de red bayesiana de Millán”. No obstante, puede observarse cómo la probabilidad del concepto C4 es de 0.217 en el “modelo de red bayesiana con pesos propuesto” y de 0.11 en el “modelo de red bayesiana con pesos propuesto”. Este valor en el “modelo de red bayesiana de Millán” se obtiene porque tenemos los siguientes valores de probabilidades condicionadas, teniendo en cuenta que la pregunta P0 tiene asociados los conceptos C0 y C4, con los pesos  $w_0 = 0,5$  y  $w_4 = 0,5$  respectivamente.

### 7.3. EXPERIMENTACIÓN CON ALUMNOS REALES

MODELO BAYESIANO DE MILLÁN		MODELO BAYESIANO PROPUESTO	
Núm. pregunta	Conceptos de la pregunta	Núm. pregunta	Conceptos de la pregunta
0	P (C0) = 0.003= 0.1*0.03*(1/0.876)	0	P (C0) = 0.067= 0.7*0.03*(1/0.312)
	P (C4) = 0.217= 0.9*0.03*(1/0.124)		P (C4) = 0.11 =0.8*0.03*(1/0.218)
1	P (C0) = 0.00003 = 0.1*0.1*0.003*(1/0.897)	1	P (C0) = 0.144 =0.7*0.7*0.067*(1/0.326)
	P (C4) = 0.643= 0.9*0.9*0.217*(1/0.274)		P (C4) = 0.331 =0.8*0.8*0.11*(1/0.266)

Figura 7.12: Probabilidades de los conceptos tras 2 ítems. Parte izquierda (modelo de red bayesiana de Millán). Parte derecha (modelo de red bayesiana con pesos propuesto)

$$p(P0|C4)_{propuesta} = G(x_{propuesta}) = \frac{G((peso_{C4} * 3) * 13,716)}{3} = G(7,5438) = 0,8 \quad (7.1)$$

$$p(P0|C0)_{millan} = G(x_{millan}) = \frac{G((2 * 13,716))}{3} = G(9,144) = 0,9 \quad (7.2)$$

Es decir, cuando los pesos de los conceptos son “iguales” para los conceptos de la pregunta, podemos tener este caso en el que la probabilidad condicionada es mayor en el “modelo de red bayesiana de Millán” que en el “modelo de red bayesiana con pesos propuesto”, lo que produce un valor mayor en la probabilidad del concepto del primero. Sin embargo, este caso lo consideramos especial puesto que al igual que hacía Millán en [38], cada concepto tiene un “orden” distinto al resto dentro del mismo ítem, no existiendo situaciones de igualdad la mayor parte de las veces. En esta última situación, el número de conceptos diagnosticados como dominados es superior en el “modelo de red bayesiana de Millán” que en el “modelo de red bayesiana con pesos propuesto”.

Otro caso donde puede suceder que la probabilidad condicionada sea mayor en el “modelo de red bayesiana de Millán” que en el “modelo de red bayesia-

## CAPÍTULO 7. EXPERIMENTACIÓN DEL MODELO Y RESULTADOS

---

na con pesos propuesto” tiene lugar cuando generalmente para 2 conceptos, estamos calculando la probabilidad condicionada en el supuesto de conocer solamente el concepto menos importante de la pregunta y este concepto tiene asignado un peso de 0.1 o de 0.2. Para una pregunta P con dos conceptos C0 y C1 con pesos  $w_0 = 0,1$  y  $w_1 = 0,9$ , tendríamos los siguientes valores:

$$p(P0|C0)_{propuesta} = G(x_{propuesta}) = \frac{G((peso_{C0} * 3) * 13,716)}{3} = G(2,60) \quad (7.3)$$

$$p(P0|C0)_{millan} = G(x_{millan}) = \frac{G((1 * 13,716))}{3} = G(4,572) \quad (7.4)$$

Lo mismo ocurre para los valores  $w_0 = 0,2$  y  $w_1 = 0,8$  cuando se conoce únicamente el concepto C0. Pero, con 3 o más conceptos, se ha observado que ya no se da esta situación para por ejemplo, 3 conceptos con pesos  $w_0 = 0,1$ ,  $w_1 = 0,1$  y  $w_2 = 0,8$ . Como resultado, este caso sigue siendo poco ocasional.

$$p(P0|C0)_{propuesta} = G(x_{propuesta}) = \frac{G((peso_{C0} * 7) * 13,716)}{7} = G(2,88) \quad (7.5)$$

$$p(P0|C0)_{millan} = G(x_{millan}) = \frac{G((1 * 13,716))}{7} = G(1,95) \quad (7.6)$$

Además, como veíamos en la tabla 6.1, los valores de las probabilidades condicionadas y, como consecuencia directa, de las probabilidades de los conceptos, son mayores en el “modelo de red bayesiana con pesos propuesto” que en el “modelo de red bayesiana de Millán” para valores distintos asignados a los pesos de los conceptos del ítem.

Desde mi punto de vista, los resultados obtenidos son mejores en el “modelo de red bayesiana con pesos propuesto” que en el “modelo de red bayesiana de Millán” por los siguientes motivos:

- En primer lugar, el hecho de que el “modelo de red bayesiana con pesos propuesto” diagnostique con menor error el número de conceptos dominados y no dominados por el alumno, es beneficioso debido a que uno de los objetivos finales de nuestro trabajo es estimar con la mayor precisión posible el nivel de conocimiento del alumno, en especial el número de conceptos no dominados por el alumno. Este objetivo estaría cumplido con un nuevo modelo de red bayesiana que realiza una estimación mejor.

- En segundo lugar, el hecho de que el aprendizaje del alumno se divida, claramente en base a los resultados, entre el número de conceptos no dominados y el número de conceptos dominados, viene a confirmar que ambos modelos distinguen sin ninguna ambigüedad para cada concepto del sistema, si está dominado o no por el alumno. En mi opinión, este punto fortalece la decisión de haber optado por un modelo de red bayesiana de conceptos para el modelo del alumno.
- En tercer lugar y no menos importante, el número de conceptos no diagnosticados es un dato muy interesante que nos permite conocer qué conceptos están en el límite entre ser considerados como no dominados o por el contrario, dominados. Dada la evolución de los valores cuantitativos que van tomando las probabilidades de la red también examinadas en nuestro trabajo, se ha observado que los conceptos no diagnosticados tienen un valor de probabilidad más próximo al valor 0.7 (límite superior) que al valor de 0.3 (límite inferior). A partir del valor 0.7, el concepto ya es considerado como dominado. Y el hecho de que solamente el “modelo de red bayesiana con pesos propuesto” sea el que en algunas ocasiones, al determinar que un concepto es no diagnosticado, ponga en duda de si, con la administración de más ítems en un futuro, pueda ser un concepto dominado, es una forma de asegurarnos con la evolución del test, de que el alumno ha afianzado con total seguridad ese concepto.

A pesar de haber sacado estas conclusiones positivas que apoyan el “modelo de red bayesiana con pesos propuesto”, debemos comentar que estos resultados han sido extraídos a partir de una muestra pequeña de 100 ítems y 33 conceptos. En un futuro, como comentaremos en el siguiente capítulo, puede ser muy interesante volver a aplicar el estudio en una red con mayor número de ítems, y también con mayor número de conceptos. También es preciso tener en cuenta lo que sucede cuando un concepto está relacionado al mismo tiempo con muchos ítems y viceversa.

## Capítulo 8

# CONCLUSIONES Y TRABAJOS FUTUROS

### 8.1. Principales aportaciones

En el presente trabajo, el foco de estudio han sido los modelos bayesianos para el modelado del estudiante en los sistemas de tests informatizados. El centro de atención ha sido la mejora de la precisión en la estimación del nivel de conocimiento del alumno.

Se ha partido de un modelo de red bayesiana ya definido en [38]. Este modelo realizaba un cálculo de las probabilidades condicionadas de la red bastante sencillo asignando los valores estas probabilidades distribuidos uniformemente en un rango de valores que iba desde el valor del azar  $c$  hasta el valor máximo de probabilidad cuando se conocen todos los conceptos de la pregunta. El número de probabilidades va desde 1 hasta el número de combinaciones de estados de los conceptos de la pregunta.

Sucedía que como la probabilidad calculada representa la probabilidad de respuesta correcta condicionada a una combinación de conceptos, donde los conceptos están ordenados por orden de importancia en la pregunta, en este trabajo, nos hemos planteado la duda de si la asignación uniforme en los valores de probabilidad a lo largo de un rango, en realidad, lo que hace es considerar que cada concepto tiene una importancia proporcional al siguiente concepto de la pregunta, sin considerar la diferencia de importancia entre un concepto y el siguiente o entre el concepto y cualquier otro de la pregunta.

En este trabajo, hemos modificado los valores de las probabilidades condicionadas de una pregunta dada la combinación de conceptos asociados, de tal modo que el valor de la probabilidad condicionada esté directamente relacionado con el *peso real* del concepto en la pregunta, siendo este parámetro *peso*

de cada concepto un parámetro que ha que definir antes de ejecutar la red y que como representa una proporción de la pregunta, es un valor entre 0 y 1. El número de estas probabilidades también es un valor comprendido entre 1 y el número de combinaciones de los conceptos de la pregunta menos 1. El peso del concepto, por tanto, será el factor que ordene el concepto dentro de la pregunta.

La razón por la que hemos considerado que el peso de los conceptos es clave en el cálculo de las probabilidades condicionadas, es porque nos parece lógico pensar que si el alumno conoce un concepto con peso grande sobre la pregunta, la probabilidad de responder correctamente la pregunta será mayor si la hacemos depender de este peso, y en caso contrario, si peso del concepto que conoce el alumno es pequeño, la probabilidad de responder correctamente la pregunta será menor si depende a su vez del concepto.

Para referirnos indistintamente a los dos modelos de redes bayesianas descritos, desde el principio del trabajo hemos renombrado los dos sistemas como “modelo de red bayesiana de Millán” al modelo de red bayesiana definido en [37] y del que partimos, y “modelo de red bayesiana con pesos propuesto” al modelo de red bayesiana que presentamos en este trabajo como una propuesta de mejora del de Millán.

Con el propósito de evaluar los dos sistemas y comparar su eficacia en la estimación del nivel de conocimiento del alumno, se han desarrollado los algoritmos de diagnóstico de ambas redes, que calculan los valores de las probabilidades a posteriori de los conceptos del sistema, para tener al final de la ejecución de un test de preguntas, información sobre el conocimiento del alumno en cada uno de los conceptos.

Se han realizado experimentos tanto con alumnos simulados como con alumnos reales en ambas redes, y nos ha parecido fundamental la comprobación con alumnos reales, puesto que dan una idea más fidedigna del desempeño de ambas redes, ya que la respuesta del alumno a cada ítem del test no se simula, sino que se adecúa a su perfil. Tras los experimentos, hemos observado que el “modelo de red bayesiana con pesos propuesto” obtiene, por regla general, valores mayores en las probabilidades a posteriori de los conceptos y se ha podido comprobar que estos mayores numéricos más altos, son debidos a que los valores de las probabilidades condicionadas en este modelo también superan a las que se tenían en el “modelo de red bayesiana de Millán”.

Para cuantificar con medidas de evaluación cercanas a la realidad educativa los valores de las probabilidades de los conceptos, hemos definido como medidas de evaluación que dependen de estas probabilidades: el número de conceptos dominados por el alumno, el número de conceptos no dominados por el alumno y el número de conceptos no diagnosticados.

Como resultado, hay un mayor número de conceptos diagnosticados como

dominados por el alumno en el “modelo de red bayesiana con pesos propuesto”, y hemos examinado, que este número se aproxima más al número real de conceptos dominados por el alumno que en el “modelo de red bayesiana de Millán”. Se ha calculado el error estándar de la estimación en los dos modelos y se ha podido ver que, en la mayoría de las ocasiones, el error estándar es menor en el “modelo de red bayesiana con pesos propuesto” que en el “modelo de red bayesiana de Millán”. Análogamente, también es mejor la estimación en el número de conceptos no dominados por el alumno en el “modelo de red bayesiana con pesos propuesto”.

## 8.2. Trabajos futuros

A partir de este trabajo, se plantean varias líneas de investigación, que pasamos a comentar.

### 8.2.1. Mejora del entorno real

Después de la realización de los experimentos con alumnos reales, se nos ha planteado como posible punto de mejora la búsqueda de un contexto web más óptimo. La aplicación web que hemos implementado en este trabajo, utiliza el objeto sesión del usuario para almacenar la lista de las preguntas que se le van administrando, de tal manera, que siguiendo el algoritmo que selecciona la siguiente pregunta del Anexo I, nunca se le presenta el mismo ítem para responder. El resultado del algoritmo de selección de pregunta es el buscado, pero como las páginas web adolecen, en ocasiones, de ser recargadas, se puede perder una pregunta que sin haberse cargado adecuadamente en la página, ya había sido eliminada de la sesión del usuario. Pensamos, por tanto, que puede ser muy útil investigar qué modo de almacenamiento se podría utilizar para optimizar la aplicación web. Insistimos en este punto porque debemos recordar que la secuencia de los ítems que se presentan al alumno puede influir sobre el número final conceptos dominados y no dominados deducidos por la red bayesiana, desde el punto de vista de que si el algoritmo selecciona para mostrar un ítem en lugar de otro, el valor de las probabilidades a posteriori de los conceptos puede variar si los conceptos relacionados son diferentes o en menor número entre un ítem y otro. En caso de darse esta situación, la diferencia no es muy significativa resultando de un concepto más estimado como dominado en una red con respecto a la otra, pero dado también que la estimación del nivel de conocimiento es casi similar en ambas redes, es un factor a considerar en una futura línea de investigación.

### 8.2.2. Uso de estándares

En el presente trabajo, se han utilizado las redes bayesianas como forma de modelado del estudiante en un test informatizado adaptativo, de tal manera que las preguntas que se le muestran al alumno se adaptan al conocimiento que va adquiriendo sobre los conceptos del sistema. Se trata, por tanto, de una adaptación del contenido.

Pero en el campo de los tests adaptativos informatizados, no podemos obviar la existencia de estándares que permitir definir tests con la posibilidad de un doble objetivo de adaptación del contenido y adaptación de la navegación.

Tras hacer un estudio del arte de los estándares actuales existentes, hemos deducido que la especificación IMS QTI: ASI Information Model [27], es la que merece más nuestro interés por dos motivos:

- La capacidad de diseñar tests adaptativos capaces de evaluar y ofrecer alimentación automática al usuario.
- La característica de interoperabilidad, que hace factible transportar los tests definidos entre distintos sistemas de e-learning que soportan la especificación, o entre diferentes unidades de aprendizaje definidas con la misma especificación.

El modelo IMS QTI se puede utilizar para crear un test adaptativo donde cada pregunta tiene un nivel de complejidad asociado. Esta característica es fundamental considerarla en este trabajo, debido a que el eje principal del trabajo es el modelado del alumno en un test adaptativo en base a su grado de conocimiento.

El proceso de adaptación con respecto al nivel de complejidad busca presentar la siguiente pregunta al usuario tomando en consideración el resultado obtenido y dicho nivel de complejidad. De este modo, si la pregunta contestada es incorrecta, la nueva habilidad estimada del alumno es menor que la que tenía, y por el contrario, si la respuesta es correcta, la siguiente pregunta se elige adecuada a un nivel de habilidad superior.

En el caso de necesitar el diseño de un test con ítems adaptados al nivel de complejidad del alumno o cualquier otra característica está la especificación IMS QTI: ASI Selection and Ordering [28].

La especificación IMS QTI ASI: Selection and Ordering es parte de la especificación IMS QTI: ASI Information Model porque es posible integrar el elemento más alto de la jerarquía Selection and Ordering, es decir, *selection\_ordering* dentro de un elemento *Assesment* o *Section*, que son objetos para definir un test de evaluación en IMS QTI: ASI Information Model.

Un test adaptativo que sigue la especificación ASI Selection and Ordering,

## CAPÍTULO 8. CONCLUSIONES Y TRABAJOS FUTUROS

es una etiqueta XML con nombre *questestinterop*, que es la raíz de QTI y contiene un conjunto de *assessments*, *sections* e *items*.

Podemos definir dentro de un elemento *questestinterop* una sección con una etiqueta *section*. Dentro de un elemento de tipo *section*, hay un elemento *selection\_ordering* que define la forma en la que se seleccionan para mostrarse los ítems del tests, mediante un conjunto de reglas de ordenación y selección que se aplican a los ítems.

Un ejemplo de definición de un *questestinterop* se muestra en la siguiente figura:

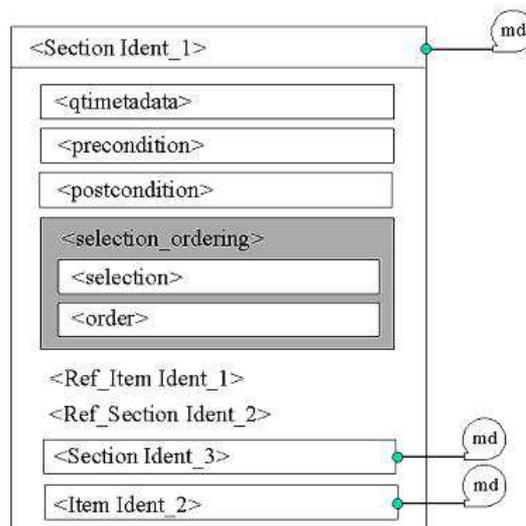


Figura 8.1: Estructura ASI Selection and Ordering [28]

En la figura 4.1, la abreviatura *md* hace referencia a que dentro del elemento al que apunta, se pueden definir metadatos.

Los elementos que aparecen en la estructura tienen las siguientes funciones:

- `qtimetadata`: contiene la definición de metadatos específica del objeto, que es la información usada posteriormente por los algoritmos de selección y ordenación.
- `precondition`: define las precondiciones que deben ser “true” para que los objetos sean procesados por el motor QTI.
- `selection_ordering`: define las reglas de ordenación que se aplican a los hijos contenidos en un objeto.

Se puede ver también que los ítems y las secciones pueden hacerse referencia a ellas si están definidas en un fichero XML aparte.

El elemento *selection\_ordering* tiene un atributo que es el tipo de secuencia (*sequence\_type*), que hace referencia a la regla de selección que se puede definir para los ítems del test y que puede ser de dos tipos:

- Normal (Normal): cada objeto solamente se puede mostrar una vez, sin ninguna repetición.
- Repetición (Repeat): cada objeto se puede mostrar un número determinado de veces.

Si no se especifica el atributo *sequence\_type*, la selección por defecto es del tipo Normal.

Dentro del elemento *selection\_ordering*, se define el subelemento *order*, que hace referencia a la regla de ordenación que se puede definir para los ítems del test y que puede ser de tres tipos:

- Fijas(Fixed): los ítems siguen la ordenación definida por la estructuración y combinación de secciones e ítems dentro de un test. Se define usando el atributo *order\_type=Fixed*.
- Secuenciales(Sequential): los ítems se muestran en el orden en el que son seleccionados. Se define usando el atributo *order\_type=Sequential*.
- Aleatorias(Random): los ítems seleccionados se muestran en orden aleatorio, que cambia con cada instanciación del test. Se define usando el atributo *order\_type=Random*.

La regla de ordenación que se aplica por defecto es la secuencial. El orden aleatorio se aplica sobre los ítems que no han sido definidos con un orden.

El elemento *selection\_ordering* también puede contener subelementos con nombre *selection*, donde cada elemento selección es una regla de selección de los ítems. Estas reglas pueden combinarse entre sí mediante las estructuras *and\_selection*, *or\_selection*, *not\_selection*.

Un subelemento fundamental dentro del elemento *selection\_ordering*, es el elemento *selection\_metadata*, que representa un atributo que se define posteriormente en los ítems, para seleccionarlos en función del valor que le asignemos. Este elemento *selection\_metadata*, tiene un atributo *md\_name* que contiene el nombre del atributo y un atributo *mdoperator* que es un operador de comparación utilizado para aplicar la regla de selección de cada ítem.

Tras examinar todos los estándares de la especificación IMS QTI, consideramos como especificación muy eficaz para la definición de tests adaptativos la especificación QTILite [29], puesto que puede contener ítems de

distintos tipos (preguntas con opciones múltiples, preguntas de asociación de ideas...) para la confección de un test rico y variado, y también porque esta especificación puede interactuar con el elemento *selection\_ordering* de la especificación QTI Selection and Ordering recién explicada.

En este trabajo, consideramos como futura línea de investigación muy interesante, la definición de un test adaptativo de tipo *questestinterop*, que presente al alumno ítems sobre los conceptos que se han deducido como no dominados por el modelo de red bayesiana. El test *questestinterop* puede definirse con reglas de selección y ordenación de los ítems en función de los conceptos que se necesiten reforzar en el aprendizaje. El objetivo será, por tanto, utilizar esta especificación para definir un test que sirva de apoyo en el aprendizaje del alumno. En este test, por ejemplo, se puede definir cada ítem asociado con un determinado concepto, y asociar a cada ítems los siguientes dos atributos:

- *qmd\_topic*: hace referencia al área o tópico al que pertenece el ítem.
- *qmd\_levelofdifficulty*: hace referencia al nivel de dificultad del ítem, que en este caso es el nivel de dificultad del concepto. Dado el sistema de preguntas y conceptos de la red bayesiana cuyo cálculo de probabilidades sigue el modelo IRT, es fácil saber el nivel de dificultad de un concepto, que puede calcularse como el nivel de dificultad más bajo de todas sus preguntas relacionadas. Recordamos, en este punto, que el modelo IRT calibra las preguntas del test con un parámetro  $b$  de dificultad del ítem.

Una figura de un test futuro que podría diseñarse como sistema de refuerzo del aprendizaje para alumno que acaba de ejecutar un test que sigue el “modelo de red bayesiana con pesos propuesto” es el siguiente:

```

-<questestinterop>
  -<section ident="IMS_QTIV1p2_I_SAO_06">
    -<selection_ordering>
      -<selection>
        -<and_selection>
          <selection_metadata mdname="qmd_topic" mdoperator="EQ">sistemas </selection_metadata>
          <selection_metadata mdname="qmd_levelofdifficulty" mdoperator="EQ">basic </selection_metadata>
        </and_selection>
      </selection>
      <order order_type="Random"/>
    </selection_ordering>
  -<item ident="item_1_concepto_01">
    -<itemmetadata>
      -<qtimetadafield>
        -<fieldlabel>qmd_levelofdifficulty</fieldlabel>
        <fieldentry>advanced</fieldentry>
      </qtimetadafield>
      -<qtimetadafield>
        -<fieldlabel>qmd_topic</fieldlabel>
        <fieldentry>sistemas</fieldentry>
      </qtimetadafield>
    </qtimetadafield>
    </itemmetadata>
    ...ejemplo de item
  </item>
  +<item ident="item_2_concepto_01">
  +<item ident="item_3_concepto_01">
</section>
</questestinterop>

```

Figura 8.2: Ejemplo de test definido con la especificación QTI LITE [29]

En este trabajo, se han identificado como tópicos: sistemas, procesos, memoria, periféricos y archivos.

En el ejemplo del test de la figura 4.2, la etiqueta *selection\_ordering* configura que los ítems seleccionados del test sean los que cumplan que el atributo *qmd\_topic* tenga como valor “sistemas” y que el atributo *qmd\_levelofdifficulty* sea igual a “advanced”.

Como acabamos de comentar, los ítems de los que se compone un *questestinterop* siguen la especificación IMS QTILite. Esta especificación es un subconjunto de IMS QTI: ASI Information Model [27], lo que implica que QTILite no soporta todas las características de la especificación completa de QTI, pero una instancia que cumple con QTILite también será compatible con la especificación QTI. Un ejemplo de característica que no soporta QTILite son los metadatos.

Un ítem dentro del elemento *questestinterop* lo definimos para que contenga un elemento *itemmetadata* (especificación IMS QTI: ASI Information Model) que contiene a su vez tantos subelementos *qtimetadafield* como atributos por los que queremos que el algoritmo de selección elija los ítems para mostrarse en el orden que se haya definido en el test. Para el resto de ele-

mentos que son necesarios para definir un ítem relativos a su tipo y material de presentación si seguimos la especificación QTILite.

La estructura de un ítem QTILite se muestra en la siguiente figura:

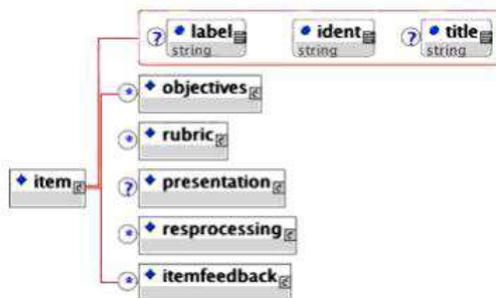


Figura 8.3: Estructura XML de un ítem QTILite [29]

En esta estructura, puede verse que un ítem QTILite se compone de 5 elementos:

- objectives.
- rubric.
- presentation.
- reprocessing.
- itemfeedback.

Dentro de estos elementos, el que permite definir el tipo de ítem es el de *presentation*, que contiene las instrucciones para la presentación del ítem en el test. Dentro de *presentation*, está el elemento *response\_lid*, que contiene las instrucciones para las respuestas de cada ítem en caso de ser el ítem seleccionado por el alumno. Y a destacar, dentro del elemento *response\_lid*, está el elemento *render\_choice*, que define a un ítem de tipo respuesta múltiple y cuya estructura se muestra en la siguiente figura:



Figura 8.4: Estructura de un ítem de respuesta múltiple en QTILite [29]

Asimismo, dentro de un ítem, están los elementos *resprocessing*, que incluyen los subelementos *outcomes* y *rescondition*. El subelemento *outcomes* define las variables que se van a utilizar en la “calificación” del ítem por el algoritmo de “scoring” (calificación). Por su parte, el elemento *rescondition* permite determinar la realimentación que se presenta al usuario en función de su respuesta al ítem, porque permite discriminar si el usuario ha respondido correctamente o no y asignar un valor a las variables que intervienen en la calificación. Un ejemplo de ítem QTILite de tipo respuesta múltiple (render\_choice) con sus elementos *feedback* en función de la opción seleccionada por el usuario es el siguiente:

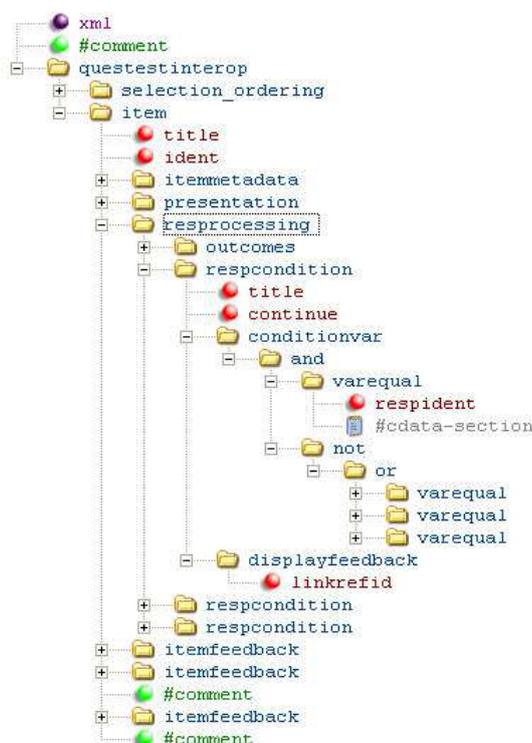


Figura 8.5: Estructura de un ítem con realimentación *feedback* en la especificación QTILite [29]

La segunda forma de uso de IMS QTI es la definición de reglas de navegación de unos ítems hacia otros en función de la respuesta que haya dado el alumno a cada ítem. La definición de estas reglas de adaptación se ha seguido en el desarrollo de la herramienta ADASAT [20] que genera cursos que siguen la especificación IMS QTI: Assessment Test, Section and Item Information Model [30].

Un test adaptativo en la especificación IMS QTI: ASI Information Model es un conjunto de ítems de evaluación (*assessment*) que muestra información a un usuario junto con información sobre cómo se evalúa el ítem. La evaluación o calificación del usuario en el sistema QTI tiene lugar cuando las respuestas del usuario se procesan por unas reglas de procesamiento. Además, puede configurarse que se puntúen de forma distinta.

Los ítems de evaluación (*assessmentItem*) pueden ser de dos tipos: ítems adaptativos e ítems no adaptativos. Los ítems adaptativos son capaces de adaptar su apariencia su puntuación en función de la respuesta del usuario. Se da la posibilidad de mostrar el mismo ítem varias veces. Cada ítem y el test en su totalidad tienen un tiempo de sesión y el resultado de evaluación del usuario se muestra cuando se agota la sesión del test, independientemente de que el usuario haya respondido a todas las preguntas.

Para la definición del test adaptativo con reglas de navegación, se puede seguir la especificación IMS QTI: Assesment Test, Section and Item Information Model.

Siguiendo esta especificación, un test comprende la definición de un elemento *assessmentTest* que incluye las secciones de *attributs*, *outcomeDeclaration*, *timeLimits*, *testPart*, *outcomeProcessing* y *testFeedback*.

Además, un *assesmentItem* contiene un elemento llamado *itemBody*. El *itemBody* contiene un objeto *interaction*, y dentro de la interacción está el objeto *response\_variable* que representa la respuesta del usuario, tiene, por ejemplo, como atributo si la respuesta ha sido correcta o incorrecta.

El cuerpo de un ítem contiene texto, imágenes, objetos multimedia, e interacciones que describen su contenido y estructura. Estos elementos pueden combinarse con plantillas de estilo a partir de reglas de estilo por defecto que ofrece el sistema. La especificación IMS QTI usa una plantilla XHTML de manera que sólo algunos elementos de XHTML se permiten en un ítem de evaluación (*assessmentItem*). Además de los ítems, un test se puede estructurar en secciones y subsecciones y establecer una navegación entre ellas, mediante una serie de reglas que se definen mediante elementos pre-condiciones (pre-conditions) y reglas de ramificación (branching rules). Por otro lado, la unidad más grande que representa el ítem en su totalidad se modela con la clase de objetos *testPart* que tiene como atributos las precondiciones (pre-Condition) y las reglas (branchRule). Por tanto, las etiquetas de precondición y de bifurcación pueden definirse en los niveles de *testPart*, *sectionPart*, *assessmentSection* y *assessmentItemRef*. También existen los fragmentos (*fragments*) que son partes de dentro de un ítem que pueden ser tratadas de forma independiente.

Un elemento precondición es un elemento opcional en cada uno de estos niveles que es evaluado justo antes de entrar a cada nivel. Más de una pre-

condición puede ser asociada a la misma unidad del test. Un elemento de bifurcación determina el siguiente elemento o sección que será procesado a continuación dependiendo el resultado de una comparación lógica. Al igual que sucede con las precondiciones, más de una bifurcación lógica puede asociarse a una unidad del test.

Siguiendo la especificación IMS QTI: Assessment Test, Section and Item Information Model, se podría plantear como otra línea de trabajo futura, la definición de un test que defina ítems correspondientes a los conceptos del sistema. Este test de tipo *testPart* diferiría del test definido como *questestinterop* que seguía la especificación QTILite, en que el ítem que muestra en cada momento, se presenta de forma adaptativa a la respuesta a cada ítem administrado. Con el *questestinterop* ya definíamos un test con un orden ya determinado de los ítems asociados a los conceptos que el “modelo de red bayesiana con pesos propuesto” diagnosticaba como conceptos no dominados por el alumno. Como resultado, sería un test definido tras cuya ejecución, tendríamos una puntuación para el alumno que podría ser comparada con el número de conceptos deducidos como dominados y no dominados por la red bayesiana.

Para construir un test adaptativo en la navegación con la especificación IMS QTI: Assessment Test, Section and Item Information Model, se pueden seguir los siguientes pasos:

1. Se define un *testPart* para un test.
2. Dentro del *testPart*, se definen secciones *assessmentSection* para agrupar los ítems de acuerdo a su grado de complejidad (alto, medio o bajo), más un ítem especial de nivel medio que podría presentarse inicialmente al alumno con el fin de determinar el nivel de conocimiento inicial del alumno en función de su respuesta correcta o incorrecta, para mostrarle posteriormente una sección de nivel de complejidad inferior o superior (bajo o alto).
3. Se definen las instrucciones de bifurcación *branchRule* que evalúan la condición dada y dirigen el procesamiento del siguiente elemento de acuerdo al resultado (falso o verdadero) de la condición. Esta condición representa si la respuesta al ítem ha sido correcta o incorrecta. La condición se especifica con la estructura *match* usando el valor de la variable *firstNormalItem.unifiedResult*. Si el resultado de la condición es negativo (valor de *firstNormalItem.unifiedResult* = N), entonces se producirá una bifurcación a la sección que presenta una información de retroalimentación al estudiante.

## CAPÍTULO 8. CONCLUSIONES Y TRABAJOS FUTUROS

---

La herramienta ADASAT [20] ya define un *testPart*, donde por cada ítem dentro de cada sección, define el siguiente ítem XML a desplegar en función de la respuesta correcta o incorrecta del alumno al ítem anterior. Un extracto de *testPart* de ADASAT se muestra en la siguiente figura:

```
<?xml version="1.0" encoding="UTF-8"?>
<testPart navigationMode="linear" submissionMode="individual">
  <itemSessionControl showFeedback="true" maxAttempts="1"
  showSolution="false"/>
  <assessmentSection identifier="sectionA" title="section A" visible="true">
    <assessmentItemRef identifier="item034" href="item034.xml">
      <itemSessionControl maxAttempts="0"/>
    </assessmentItemRef>
    <assessmentItemRef identifier="item160" href="item160.xml">
      <!-- Despliega el ítem solo si la respuesta previa tiene un valor de Y-->
      <preCondition>
        <match>
          <variable identifier="item034.RESPONSE"/>
          <baseValue baseType="identifier">Y</baseValue>
        </match>
      </preCondition>
      <!--Salta al ítem 656 si la respuesta es N -->
      <branchRule target="item656">
        <match>
          <variable identifier="item034.RESPONSE"/>
          <baseValue baseType="identifier">N</baseValue>
        </match>
      </branchRule>
    </assessmentItemRef>
  </assessmentSection>
</testPart>
```

Figura 8.6: *testPart* de la herramienta ADASAT [20] con precondiciones y reglas de bifurcación

En resumen, hemos presentado dos posibles futuras líneas de investigación que utilizan estándares para la definición de tests adaptativos que son:

- La definición de un test *questestinterop* que siga la especificación QTI-Lite. Este test se compone de preguntas de respuesta múltiple (render\_choice) con unas reglas de selección y ordenación de ítems, mediante el elemento *selection\_ordering* de la especificación IMS QTI: ASO Selection and Ordering. Este test se presenta al alumno después de ejecutar un test modelado con el “modelo de red bayesiana con pesos propuesto” que se presenta en este trabajo. El objetivo es reforzar aquellos conceptos que la red bayesiana ha diagnosticado como no dominados por el alumno. Es por ello que los ítems del test, se definen de modo que su material de presentación o texto, hace referencia a un concepto no dominado por el alumno. Puede haber varios ítems que estén relacionados en su contenido con el mismo concepto. Esta línea de investigación nos parece útil por la posibilidad de diseñar tests que

sean interoperables y capaces de presentación realimentación al usuario para reforzar su aprendizaje.

- La definición de un test *testPart* que siga la especificación IMS QTI: Assessment Test, Section and Item Information Model, que a priori, no conozca la secuencia de ítems que se le van a presentar al alumno, porque se decidirá dinámicamente con cada respuesta del alumno a cada ítem del test. Un test de este tipo contiene, por tanto, reglas de navegación que determinan en función de la respuesta a cada ítem (precondition) el siguiente ítem que se muestra al alumno. Esta línea de investigación, a diferencia de la anterior, no aprovecha los resultados de la red bayesiana en cuanto a que ya conoceríamos el conjunto de conceptos no dominados por el alumno, pero en contra, sí puede servir para comparar el nivel de conocimiento del alumno con un *testPart* con el de un test adaptativo bayesiano.

Para la ejecución de los tests que siguen la especificación IMS, se necesitan herramientas que la soporten. Hay varias herramientas de e-learning precisamente debido al estándar que se pueden utilizar (QTITools, Moodle...). Cada una de ellas, además de los ficheros XML que definen el test, llevan a cabo un empaquetado de un conjunto de ficheros de configuración del test dando como resultado un fichero .zip. Esta forma de agrupación y estructuración de los recursos de un test o curso educativo siguen la especificación IMS Content Packaging [31]. En el caso de la herramienta qplayer [44], el fichero desde donde se hace referencia a cada test definido es el Qmanifest.xml. Además, en qplayer hay una subcarpeta con el nombre XML donde se almacenan los ficheros de test de tipo XML. Desde la página web de la herramienta qplayer, es posible descargarse una versión gratuita que consiste en un fichero .zip con los archivos de un curso adaptativo que ya viene creado. El contenido de la carpeta qplayer\_demo descargada es el siguiente:

1. Qmanifest.xml: es un fichero de manifiesto con la descripción general del curso y que contiene las referencias a los ficheros de tests que serán archivos XML.
2. Qplayer\_config.xml: es un fichero de configuración del curso, en el que se da valor a propiedades generales como: el lenguaje del curso, el modo de navegación, la posibilidad de mostrar las respuestas, la realimentación, el modo de presentación de las preguntas y entre otras propiedades, una referencia a la URL del fichero manifiesto.
3. Una subcarpeta XML, que contiene los archivos XML que corresponden a cada test de tipo *questestinterop*.

4. Una subcarpeta IMG que contiene los archivos de tipo imagen incluidos en los tests.
5. Un fichero index.html que es el fichero principal del curso. Este fichero es esencial porque para ejecutar los tests, basta con hacer doble click sobre él.

### 8.2.3. Relaciones de prerrequisito

Las relaciones de prerrequisito se establecen entre los conceptos de la red bayesiana[37].

Sería interesante partir de la estructura de red bayesiana del “modelo de red bayesiana con pesos propuesto” que se compone de relaciones de dependencia causal entre los nodos concepto y los nodos pregunta y añadirle relaciones entre los conceptos.

Estas relaciones entre los conceptos simbolizarían que para conocer un concepto dado es preciso conocer los nodos concepto que influyen directamente en él. De este modo, se puede deducir más rápidamente el conocimiento de un concepto dado porque podemos suponer que se conoce una vez que se conocen todos sus conceptos prerrequisito. Análogamente, una vez que se ha diagnosticado que el alumno no conoce un concepto dado, ya no haría falta seguir investigando si conoce los conceptos de los que el concepto es prerrequisito.

Sin embargo, la introducción de estas relaciones complica la estructura de la red bayesiana y la cambia por completo, por lo que es un problema que habría que estudiar detenidamente.

### 8.2.4. Equilibrio de contenido

En el banco de ítems que hemos definido, nos hemos preocupado de que fueran variados y de que pertenecieran a áreas del dominio diferenciadas, para que el conjunto fuera lo más equilibrado posible, sin embargo, no hemos utilizado ningún método teórico para controlar la exposición de los ítems del test como el criterio de Sympson-Hetter [34].

Como la red bayesiana selecciona como siguiente ítem a ser presentado al alumno aquél que maximice la probabilidad de que se responda correctamente dados los conceptos conocidos que se tengan hasta el momento (criterio de máxima utilidad de la pregunta [38] y cuyo algoritmo A se incluye en el Anexo I), se propone como futuro trabajo, definir los conceptos de tal modo que pertenezcan a diferentes áreas y refinar el algoritmo A para que añada al criterio de selección como segunda prioridad, el equilibrio de los conceptos.

Esta red podría compararse con un sistema de tests adaptativos informatizados que tratan el equilibrio de contenido como SIETTE [11].

# Apéndice A

## Anexo I

A continuación, mostramos los algoritmos de diagnóstico implementados en el presente trabajo para poder comparar los dos modelos bayesianos, “modelo de red bayesiana propuesta” y “modelo de red bayesiana de Millán”.

Ambos algoritmos utilizan un procedimiento común que se ocupa de seleccionar la siguiente pregunta del test, que hemos nombrado “obtenerSi-guientePregunta” y que se muestra a continuación de los dos algoritmos de diagnóstico.

---

**Algoritmo 5** Algoritmo de diagnóstico del modelo de red bayesiana de Millán.

---

**entrada:** red bayesiana  $(D, P)$  definida en un conjunto de variables  $X$

```
1: para  $i \leftarrow 1, 50$  hacer
2:    $dominados = 0$ 
3:    $nodominados = 0$ 
4:    $nodiagnosticados = 0$ 
5:   mientras  $numpreguntas \leq 100$  and  $dominados \leq 33$  hacer
6:      $idpregunta \leftarrow obtenerSiguientePregunta()$ 
7:      $subconceptos \leftarrow idconceptosdeidpregunta$ 
8:      $probabilidadesconceptos \leftarrow probconceptosdeidpregunta$ 
9:     si  $idpregunta$  es correcta then
10:       $preguntasbiencontestadas[idpregunta] = -1$ 
11:     else si pregunta no es correcta then
12:       $preguntasbiencontestadas[idpregunta] = -2$ 
13:     fin si
14:      $total \leftarrow Math.pow(2,0, subconceptos.size())$ 
15:     para  $c \leftarrow 1, total - 1$  hacer
16:        $Stringbinario \leftarrow Integer.toBinaryString(c)$ 
17:        $x \leftarrow (c * 13,716) / (total - 1)$ 
18:        $probabilidad \leftarrow 1 - ((1 - ci) * (1 + Math.exp((-1,7 * ai) * (x - bi)))) / (1 + Math.exp((1,7 * ai) * (x - bi)))$   $\triangleright$  almacenar la probabilidad de la pregunta condicionada al concepto  $p(P/C)$ 
19:        $conceptos_preguntas[c][idpregunta] \leftarrow probabilidad$ 
20:     fin para
21:   fin mientras
22: fin para
```

---

---

```

23: para  $j \leftarrow 1, \text{subconceptos.size}()$  hacer
24:   para  $k \leftarrow 1, 100$  hacer
25:     si  $p(C_j) = 0,0$  then
26:        $p(P|C) \leftarrow p(P|C) * 0,03$ 
27:        $p(P|\neg C) \leftarrow p(P|\neg C) * 0,97$ 
28:     else si  $p(C_j) \neq 0,0$  then
29:        $p(P|C) \leftarrow p(P|C) * p(C_j)$ 
30:        $p(P|\neg C) \leftarrow p(P|\neg C) * p(C_j)$ 
31:     fin si            $\triangleright$  Si la pregunta se ha contestado correctamente
32:   si  $\text{preguntasbiencontestadas}[k] = -1$  then
33:      $p(P|C) \leftarrow p(P|C) * \text{conceptos}_p\text{reguntas}[j][k]$ 
34:      $p(P|\neg C) \leftarrow p(P|\neg C) * (1 - \text{conceptos}_p\text{reguntas}[j][k])$ 
35:      $\text{sumaprob} \leftarrow p(P|C) + p(P|\neg C)$ 
36:      $p^*(C_j) \leftarrow p(P|C) / \text{sumaprob}$ 
37:   fin si            $\triangleright$  Si la pregunta no se ha contestado correctamente
38:   si  $\text{preguntasbiencontestadas}[k] = -2$  then
39:      $p(\neg P|C) \leftarrow p(\neg P|C) * \text{conceptos}_p\text{reguntas}[j][k]$ 
40:      $p(\neg P|\neg C) \leftarrow p(\neg P|\neg C) * (1 - \text{conceptos}_p\text{reguntas}[j][k])$ 
41:      $\text{sumaprob} \leftarrow p(\neg P|C) + p(\neg P|\neg C)$ 
42:      $p^*(C_j) \leftarrow p(\neg P|C) / \text{sumaprob}$ 
43:   fin si
44: fin para
45: fin para
46: para  $p \leftarrow 1, p^*(C_p)$  hacer
47:   si  $p(C_j) \leq 0,3$  then
48:      $\text{nodominados} \leftarrow \text{nodominados} + 1$ 
49:   fin si
50:   si  $p(C_j) \geq 0,7$  then
51:      $\text{dominados} \leftarrow \text{dominados} + 1$ 
52:   fin si
53:   si  $p(C_j) > 0,3 \text{ and } p(C_j) < 0,7$  then
54:      $\text{nodiagnosticados} \leftarrow \text{nodiagnosticados} + 1$ 
55:   fin si
56: fin paradevolver:  $\text{dominados}, \text{nodominados}, \text{nodiagnosticados}$ 

```

---

---

El algoritmo de diagnóstico del “modelo de red bayesiana de Millán” descrito en el algoritmo 0 tiene como centro principal un bucle programado While que se ejecuta siempre y cuando el número de conceptos dominados por la red sea menor de 33 conceptos (total de conceptos del sistema) o bien, se haya mostrado el total de 100 ítems del test. Lo primero que hace el algoritmo es seleccionar una pregunta para mostrarse en el test que se obtiene tras la llamada a la función obtenerSiguientePregunta cuyo algoritmo aparece al final de este Anexo. Una vez que el alumno responde la pregunta, en función de su respuesta (correcta o incorrecta) los conceptos que estén relacionados con la pregunta actualizan su valor. Los dos casos que pueden darse son:

- Si la respuesta a la pregunta ha sido correcta, es decir, tenemos una evidencia positiva, entonces se calcula la probabilidad de tener una respuesta correcta condicionada a conocer el concepto,  $p(p^+|C)$ .
- Si la respuesta a la pregunta ha sido incorrecta, es decir, tenemos una evidencia negativa, entonces se calcula la probabilidad de tener una respuesta incorrecta condicionada a conocer el concepto,  $p(p^-|C)$ .

Estas probabilidades condicionadas se calculan con el modelo logístico de 3 parámetros de la TRI evaluando la función  $G(x)$  (ver ecuación 4.16) con el valor  $c * G(x^*)/2^p - 1$ , para  $p$  conceptos de la pregunta. En esta expresión el factor  $c$  toma los valores desde  $0,2^p - 1$ . De este modo, el valor de  $c$  se corresponde con cada una de las posibles combinaciones binarias (conocido=1, desconocido=0) de los conceptos de la pregunta.

Como hay que tener en cuenta todos los conceptos padre de la pregunta, se calculan un total de probabilidades condicionadas igual a  $2^p$ . Tras calcularse todas las probabilidades condicionadas de la pregunta, se almacenan para poder ser utilizadas por el penúltimo proceso del algoritmo.

El penúltimo proceso del algoritmo consiste en actualizar los valores de las probabilidades a posteriori de todos los conceptos del sistema. Para calcular la probabilidad de cada concepto, se aplica la expresión de la ecuación 6.13. Después de este proceso, todos los conceptos tendrán un valor de probabilidad entre 0 y 1.

Por último, se consulta la probabilidad calculada para concepto, dando lugar a tres situaciones:

- Si la probabilidad a posteriori del concepto es menor de 0,3, entonces se suma una unidad al número de conceptos no dominados del sistema.
- Si la probabilidad a posteriori del concepto es mayor de 0,7, entonces se suma una unidad al número de conceptos dominados del sistema.

**Algoritmo 6** Algoritmo de diagnóstico del modelo de red bayesiana con pesos propuesto.

---

**entrada:** red bayesiana  $(D, P)$  definida en un conjunto de variables  $X$

```

para  $i \leftarrow 1, 50$  hacer
2:    $dominados \leftarrow 0$ 
      $nominados \leftarrow 0$ 
4:    $nodiagnosticados \leftarrow 0$ 
     mientras  $numpreguntas \leq 100$  and  $dominados \leq 33$  hacer
6:      $idpregunta \leftarrow obtenerSiguientePregunta()$ 
        $subconceptos \leftarrow idconceptosdeidpregunta$ 
8:      $probabilidadesconceptos \leftarrow probconceptosdeidpregunta$ 
       si  $idpregunta$  es correcta then
10:       $preguntasbiencontestadas[idpregunta] \leftarrow -1$ 
       fin si
12:     si  $idpregunta$  no es correcta then
        $preguntasbiencontestadas[idpregunta] \leftarrow -2$ 
14:     fin si
        $total \leftarrow Math.pow(2, subconceptos.size())$ 
16:     para  $c \leftarrow 1, total - 1$  hacer
        $Stringbinario \leftarrow Integer.toBinaryString(c)$ 
18:        $cont \leftarrow binario.length() - 1$ 
       para  $b \leftarrow 1, binario.length()$  hacer
20:         si  $binario[b] == 1$  then
            $pesoconcepto \leftarrow probabilidadesconceptos[cont]$ 
22:         else si  $binario[b] == 0$  then
            $pesoconcepto \leftarrow (1 - probabilidadesconceptos[cont])/10$ 
24:          $pesofinal \leftarrow pesofinal + pesoconcepto$ 
            $cont = cont - 1$ 
26:         fin si
       fin para
28:        $x \leftarrow (pesofinal * (total - 1) * 13,716) / (total - 1)$ 
        $probabilidad \leftarrow 1 - ((1 - ci) * (1 + Math.exp((-1,7 * ai) * (x - bi)))) / (1 + Math.exp((1,7 * ai) * (x - bi)))$ 
30:        $conceptos_preguntas[c][idpregunta] \leftarrow probabilidad;$ 
       fin para
32:     fin mientras
fin para

```

---

---

```

34: para  $j \leftarrow 1, \text{subconceptos.size}()$  hacer
    para  $k \leftarrow 1, 100$  hacer
36:     si  $p(C_j) = 0,0$  then
         $p(P|C) \leftarrow p(P|C) * 0,03$ 
38:      $p(P|\neg C) \leftarrow p(P|\neg C) * 0,97$ 
    else si  $p(C_j) \neq 0,0$  then
40:      $p(P|C) \leftarrow p(P|C) * p(C_j)$ 
         $p(P|\neg C) \leftarrow p(P|\neg C) * p(C_j)$ 
42:     fin si
    si  $\text{preguntasbiencontestadas}[k] = -1$  then
44:      $p(P|C) \leftarrow p(P|C) * \text{conceptos\_preguntas}[j][k]$ 
         $p(P|\neg C) \leftarrow p(P|\neg C) * (1 - \text{conceptos\_preguntas}[j][k])$ 
46:      $\text{sumaprob} \leftarrow p(P|C) + p(P|\neg C)$ 
         $p^*(C_j) \leftarrow p(P|C) / \text{sumaprob}$ 
48:     fin si
    si  $\text{preguntasbiencontestadas}[k] = -2$  then
50:      $p(\neg P|C) \leftarrow p(\neg P|C) * \text{conceptos\_preguntas}[j][k]$ 
         $p(\neg P|\neg C) \leftarrow p(\neg P|\neg C) * (1 - \text{conceptos\_preguntas}[j][k])$ 
52:      $\text{sumaprob} \leftarrow p(\neg P|C) + p(\neg P|\neg C)$ 
         $p^*(C_j) \leftarrow p(\neg P|C) / \text{sumaprob}$ 
54:     fin si
    fin para
56: fin para
    para  $p \leftarrow 1, p^*(C_p)$  hacer
58:     si  $p(C_j) \leq 0,3$  then
         $\text{nodominados} \leftarrow \text{nodominados} + 1$ 
60:     fin si
    si  $p(C_j) \geq 0,7$  then
62:      $\text{dominados} \leftarrow \text{dominados} + 1$ 
    fin si
64:     si  $p(C_j) > 0,3 \text{ and } p(C_j) < 0,7$  then
         $\text{nodiagnosticados} \leftarrow \text{nodiagnosticados} + 1$ 
66:     fin si
fin paradevolver:  $\text{dominados}, \text{nodominados}, \text{nodiagnosticados}$ 

```

---

---

**Algoritmo 7** Algoritmo de selección de la siguiente pregunta.

---

**entrada:** red bayesiana  $(D, P)$  definida en un conjunto de variables  $X$

```

para  $i \leftarrow 1, 100$  hacer
    obtener los conceptos de la pregunta  $i \rightarrow$  array subconceptos y array
    probabilidadesconceptos
3:   para  $c \leftarrow 1, subconceptos.size()$  hacer
         $peso \leftarrow probabilidadesconceptos[c]$ 
         $peso \leftarrow peso * (subconceptos.size() - 1)$ 
6:    $x \leftarrow (peso * 13,716) / (subconceptos.size() - 1)$ 
         $probabilidad \leftarrow 1 - ((1 - ci) * (1 + Math.exp((-1,7 * ai) * (x - bi)))) / (1 + Math.exp((1,7 * ai) * (x - bi)))$ 
         $probabilidadcorrecta \leftarrow probabilidad * p(C_c) + probabilidad * (1 - p(C_c))$ 
9:    $probabilidadincorrecta \leftarrow (1 - probabilidad) * (1 - p(C_c)) + (1 - probabilidad) * p(C_c)$ 
        si  $probabilidadcorrecta > probabilidadincorrecta$  then
             $probabilidadespregunta[c] \leftarrow probabilidadcorrecta$ 
12:  fin si
        fin para
    fin para
15:  $probabilidadmaxima \leftarrow 0,0$ 
    para  $p \leftarrow 1, probabilidadespreguntas.size()$  hacer
        si  $probabilidadespreguntas[p] > probabilidadmaxima$  then
18:      $probabilidadmaxima \leftarrow probabilidadespreguntas[p]$ 
         $idpregunta \leftarrow preguntas[p]$ 
        fin si
21: fin paradevolver:  $idpregunta$ 

```

---

- 
- Si la probabilidad a posteriori del concepto es mayor de 0,3 y menor de 0,7, entonces se suma una unidad al número de conceptos no diagnosticados del sistema.

El algoritmo devuelve el número de conceptos no dominados, dominados y no diagnosticados.

El algoritmo de diagnóstico del “modelo de red bayesiana con pesos propuesto” descrito en el algoritmo 0 tiene como centro principal un bucle programado While que se ejecuta siempre y cuando el número de conceptos dominados por la red sea menor de 33 conceptos (total de conceptos del sistema) o bien, se haya mostrado el total de 100 ítems del test. Lo primero que hace el algoritmo es seleccionar una pregunta para mostrarse en el test que se obtiene tras la llamada a la función obtenerSiguientePregunta cuyo algoritmo aparece al final de este Anexo. Una vez que el alumno responde la pregunta, en función de su respuesta (correcta o incorrecta) los conceptos que estén relacionados con la pregunta actualizan su valor. Los dos casos que pueden darse son:

- Si la respuesta a la pregunta ha sido correcta, es decir, tenemos una evidencia positiva, entonces se calcula la probabilidad de tener una respuesta correcta condicionada a conocer el concepto,  $p(p^+|C)$ .
- Si la respuesta a la pregunta ha sido incorrecta, es decir, tenemos una evidencia negativa, entonces se calcula la probabilidad de tener una respuesta incorrecta condicionada a conocer el concepto,  $p(p^-|C)$ .

Estas probabilidades condicionadas se calculan con el modelo logístico de 3 parámetros de la TRI evaluando la función  $G(x)$  (ver ecuación 4.16) con el valor  $peso * G(x^*) * (2^p - 1)/2^p - 1$ , para  $p$  conceptos de la pregunta. En esta expresión el peso se calcula como la suma del peso de cada concepto considerado en una combinación lineal de los conceptos de la pregunta de la siguiente forma:

- Si la posición del concepto en la combinación binaria es un 1, entonces a la variable peso se suma la probabilidad del concepto en la pregunta.
- Si la posición del concepto en la combinación binaria es un 0, entonces a la variable peso se suma el complementario de la probabilidad del concepto en la pregunta dividiendo el resultado por 10. La decisión de dividir por 10 es arbitraria y se corresponde con el peso de no conocer el concepto en la pregunta.

Como hay que tener en cuenta todos los conceptos padre de la pregunta, se calculan un total de probabilidades condicionadas igual a  $2^p$ . Tras calcularse

todas las probabilidades condicionadas de la pregunta, se almacenan para poder ser utilizadas en el penúltimo proceso del algoritmo.

El penúltimo proceso del algoritmo consiste en actualizar los valores de las probabilidades a posteriori de todos los conceptos del sistema. Para calcular la probabilidad de cada concepto, se aplica la expresión de la ecuación 6.13. Después de este proceso, todos los conceptos tendrán un valor de probabilidad entre 0 y 1.

Por último, se consulta la probabilidad calculada para concepto, dando lugar a tres situaciones:

- Si la probabilidad a posteriori del concepto es menor de 0,3, entonces se suma una unidad al número de conceptos no dominados del sistema.
- Si la probabilidad a posteriori del concepto es mayor de 0,7, entonces se suma una unidad al número de conceptos dominados del sistema.
- Si la probabilidad a posteriori del concepto es mayor de 0,3 y menor de 0,7, entonces se suma una unidad al número de conceptos no diagnosticados del sistema.

El algoritmo devuelve el número de conceptos no dominados, dominados y no diagnosticados.

El algoritmo de selección de la siguiente pregunta descrito en el algoritmo A recorre en un bucle programado for todas las preguntas del sistema. Por cada pregunta, obtiene para todos los conceptos de la pregunta, la probabilidad de la pregunta condicionada al concepto  $p(P|C)$ . De las probabilidades  $p(P|C)$  correspondientes a cada pregunta, seleccionamos la probabilidad máxima. Y de todas las probabilidades máximas, nos quedamos con la probabilidad máxima, es decir, con la pregunta cuyo valor máximo de  $p(P|C)$  es el valor mayor.

El algoritmo devuelve esta pregunta como el siguiente ítem a mostrar en el test.

---

# Bibliografía

- [1] AHA project, disponible en: <http://aha.win.tue.nl/>
- [2] Andersen, S.K., Olesen, K.G., Jensen, F.V., Jensen, F. (1985). HUGIN-a Shell for Building Bayesian Belief Universes for Expert Systems. *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI-89)*, 1080-1085. Morgan Kaufmann Publishers, San Mateo, CA.
- [3] Brusilovsky, P. (2002). From Adaptative Hypermedia to the Adaptative Web. En P. Brusilovsky and M. T. Maybury (eds.) *Communications of the ACM 45 (5), Special Issue on the Adaptive Web*, 31-33.
- [4] Boticario and Gaudioso, 2003. Sistemas Interactivos de Enseñanza/Aprendizaje. En Sanz y Torres (eds.) *Sistemas adaptativos*, (pp. 109-149). Madrid: Editores.
- [5] Castillo, E., Gutiérrez J.M., Hadi, A.S. (1996). Sistemas expertos y Modelos de Redes Probabilísticas. *Monografías de la Academia de Ingeniería*. Algunos conceptos sobre grafos (pp. 128-191).
- [6] Chan, T.W. (1991). Integration-Kid: A Learning Companion System. En J. Mylopoulos, and R. Reiter, (eds.) *Proceedings of the Twelfth International Conference on Artificial Intelligence Volume 2*, Sydney, Australia, 904-1099. Morgan Kaufmann Inc.
- [7] Chin, D.N. (1994). KNOME: Modeling What the User Knows in UC. En A. Kobsa and W. Wahlster (eds.) *User Models in Dialog Systems*, 74-107. Springer, Berlin, Heidelberg.
- [8] Clancey, W.J., Letsinger, R. (1984). Reconfiguring a rule-based system for application to teaching. En W.J. Clancey, and E.H. Shortliffe, (eds.) *Readings in Medical Artificial Intelligence : the First Decade*, pp.361-381. Reading, MA : Addison-Wesley.

- 
- [9] Cumming, G., Self, J. (1991). Learner Models in Collaborative Intelligent Educational Systems. En Peter Goodyear (ed.), *Teaching Knowledge and Intelligent Tutoring*, 85-104, Norwood, N.J.: Ablex.
- [10] Conejo, R., Millán, E., Pérez de la Cruz, J.L., Trella, M. (2001). Modelado del alumno: un enfoque bayesiano. *Inteligencia Artificial. Revista iberoamericana de Inteligencia Artificial*, 192(5), 50-58. Valencia (España): Asociación Española para la Inteligencia Artificial.
- [11] Conejo, R., Guzmán, E., Millán, E., Trella M., Pérez de la Cruz, J.L, Ríos, A. (2004). SIETTE: A Web-Based Tool for Adaptive Testing. *International Journal of Artificial Intelligence in Education*, 14, 1-33. Málaga: Departamento de Lenguajes y Ciencias de la Computación, Facultad de Informática.
- [12] Conejo, R., Guzmán, E., Pérez de la Cruz, J.L, Millán, E. (2007). Estudio de un mecanismo para la administración adaptativa de ayudas en la realización de tests. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 33(11), 13-23. Valencia (España): Asociación Española para la Inteligencia Artificial.
- [13] Conati, C., VanLeh, K. (1996). POLA: a student modeling framework for Probabilistic On-Line Assesment of problem solving performance. *Proceedings of the fifth International Conference UM96*, 75-82. SpringerWienNewYork.
- [14] Conati, C., Gertner, A.S., VanLehn, K., Drudzel, M. (1997). On-Line Student Modeling for Coached Problem Solving Using Bayesian Networks. In Jameson A., Paris C., Tasso C., (eds.) *User Modeling; Proceedings of the sixth International Conference UM97*, 231-242. SpringerWienNewYork.
- [15] Corbett, A.T, Anderson, J.R, O'Brien, A.T. (1995). Cognitively Diagnostic Assesment. En P.D Nichols, S.F. Chipman, R.L. Brennan, (eds.) *Student Modeling in the ACT Programming Tutor* (pp. 19-27). New Jersey, NY: Lawrence Erlbaum Associates, Inc.
- [16] Cooper, G., Herskovits, E. (1992). A Bayesian Method for the Induction of Probabilistic Networks from Data. *Machine Learning*, 9, 309-347. Kluwer Academic Publishers, Boston.
- [17] DERIVE disponible en: <http://www.gratisprogramas.org/descargar/derive-6-full-incluye-manual/>

## BIBLIOGRAFÍA

---

- [18] ELVIRA Proyecto disponible en: <http://www.ia.uned.es/ elvira/>
- [19] Gyftodimos, E., Flach, P.A. (2002). Hierarchical Bayesian Networks: A Probabilistic Reasoning Model for Structured Domains. *In Proc Workshop on Development of Representations, International Conference on Machine Learning (ICML)*, 1-8. Cambridge, MA, USA.
- [20] Gonzalo, H. Generador de pruebas objetivas adaptadas a las preferencias de presentación de los usuarios. Universidad de Salamanca. Departamento de Informática e Automática. PhD Thesis. (Sep 2010).
- [21] Gürer, D.W., desJardins, M., Schlager, M. (1985). Representing a Student's Learning States and Transitions. *Association for the Advancement of Artificial Intelligence Technical Report SS-95-05*, pp. 1-9.
- [22] Hambleton R.K., Jane Rogers H. (1991). Fundamentals of Item Response Theory. En D.S. Foster. Concepts, Models and Features, (pp. 7-31). California: Sage Publications, Inc.
- [23] Heckerman D. (1992). The Certainty Factor Model. In S. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, 131-138. Wiley, New York.
- [24] Hoppe, U., Verdejo, F., Kay, J. (2003). Artificial Intelligence in Education: shaping the future of learning through intelligent technologies. H.U. Hoppe et al. (eds.), (pp. 65-73). IOS Press, Amsterdam.
- [25] HUGIN LITE disponible en: <http://www.hugin.com/productservices/demo/hugin-lite/>
- [26] HUGIN API JAVA disponible en: <http://download.hugin.com/webdocs/manuals/api-manual.pdf>
- [27] IMS Question and Test Interoperability: ASI Information Model Specification. Final Specification Version 1.2 disponible en: <http://www.imsglobal.org/question/qtiv1p2>
- [28] IMS Question and Test Interoperability: ASI Selection and Ordering. Final Specification Version 1.2 disponible en: <http://www.imsglobal.org/question/qtiv1p2>
- [29] IMS Question and Test Interoperability QTILite Specification. Final Version 1.2 disponible en: <http://www.imsglobal.org/question/qtiv1p2>

- 
- [30] IMS Question and Test Interoperability Assesment Test, Section and Item Information Model. Version 2.1. Public Draft revision 2 Specification disponible en: <http://www.imsglobal.org/question/qtiv2p1pd2>
- [31] IMS Content Packaging Specification Primer disponible en: <http://imsglobal.org/content/packaging/cp1p2pd2>
- [32] Jensen, F.V., Chamberlain, B., Nordahl, T., Jensen. F. (1991). Analysis in HUGIN of Data Conflict, In P.P. Bonissone, M. Henrion, L.N. Kanal, and J.F. Lemmer, (eds.) *Uncertainty in Artificial Intelligence*, 6, 546-554. Elsevier Science Publishers B.V.
- [33] JAVABAYES disponible en :<http://www.cs.cmu.edu/~fgcozman/Research/JavaBayes/Home/>
- [34] Leung C., HuaHua C., Hau K. (2003). Computerized Adaptive Testing: A Comparison of Three Content Balancing Methods. *The Journal of Technology, Learning, and Assessment*, 2(5), 1-16.
- [35] Martin, J., VanLehn, K. (1995). Student assessment using Bayesian nets. *International Journal of Human-Computer Studies*, 42, 575-591.
- [36] Mendel, J. M. (1995). Fuzzy Logic Systems for Engineering: A Tutorial. *Proceedings of the IEEE* 83(3), 345-377.
- [37] Millán, E. Sistema bayesiano para modelado del alumno. Universidad de Málaga. Departamento de Lenguajes y Sistemas informáticos. PhD Thesis. (May 2000).
- [38] Millán, E., Pérez-de-la-Cruz, J.L., (2002). A Bayesian Diagnostic Algorithm for Student Modeling and its Evaluation. En Kluwer Academic Publishers (Eds). *User Modeling and User-Adapted Interaction*, 12, 281-330.
- [39] Millán, E., Pérez-de-la-Cruz, J.L., (2003). García, F. Dynamic versus Static Student Models Based on Bayesian Networks: An Empirical Study. En V. Palade, R.J., Howlett, and L.C. Jain (Eds). *Computer Science*, 1337-1344.
- [40] Mitrovic, A. (Freiburg, 1998). A knowledge-based Teaching System for SQL. In *Proc. of EDMEDIA/ED-TELECOM'98*, 1027-1032.
- [41] NETICA Application disponible en: <http://www.norsys.com/netica.html>

## BIBLIOGRAFÍA

---

- [42] Olea J., Ponsoda V. (1996). Tests adaptativos informatizados. In Muñiz, J.(ed). *Psicometría*, 1-139. Madrid: Universitas.
- [43] Pearl J. (1988). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. En Morgan Kaufmann Publishers (eds.); edición 2nd (1 de junio de 1988). *Uncertainty in AI Systems: an overview* (pp. 1-26). California: Editores.
- [44] Qplayer software disponible en: <http://www.e-teach.ch/qplayer.php>.
- [45] Reye, J. (2004). Student Modelling based on Belief Networks. En IOS Press (eds.). *International Journal of Artificial Intelligence in Education*, 14, 1-33.
- [46] Rojas, A. (2001). Pasado, presente y futuro de los Tests Adaptativos Informatizados: entrevista con Isaac I. Bejar. *Psicothema*, 4(13), 685-690.
- [47] Self, J. (1990). Bypassing the Intractable Problem of Studing Modelling. En C. Frasson and G. Gauthier (eds.). *Intelligent Tutoring Systems: at the Crossroads of Artificial Intelligence and Education*, 107(23). Norwood, N.J.:Ablex.
- [48] Self, J. (1994). Formal Approaches to Student Modelling. En McCalla, G.I. and Greer, J. (eds.). *Student Modelling: the key to the individualized knowledge-based instruction*, (pp. 295-352).
- [49] SMILE disponible en: <http://genie.sis.pitt.edu/>
- [50] Spearman, C. (1904). The proof and measurement of association between two things. *International Journal of Epidemiology*, 5(39), 1137-1150.
- [51] Suraweera P., Mitrovic, A., Cerri, S., Gouarderes, G., Paraguacu, F. (Francia, 2002). KERMIT: A Constraint-Based Tutor for Database Modeling. In: S. Cerri, G. Gouarderes and F. Paraguacu (eds.). *Proc. ITS 2002, LCNS 2363, 2002*, 377-387.
- [52] Weiss, S.M, Kulikowski, C.A, Safir, A. (1977). A model-based consultation system for the long-term management of glaucoma, *Proceedings de International Joint Conference on Artificial Intelligence*, Massachusetts, Vol 2, 826-833.

- [53] Weber, G. (1999). Adaptative Learning Systems in the World Wide Web. *Proceedings of 7th International Conference on User Modeling UM99*, Banff, Canada, 371-378.
- [54] Weber, G. and Spetch (1997). M. User Modelling and Adaptative Navigation Support in WWW-Based Tutoring Systems. En A. Jameson, C. Paris, and C. Tasso (eds.). *User Modeling: Proceedings of the Sixth International Conference UM97*, 289-300. SpringerVerlag, Wien.
- [55] Zadeh, L. Fuzzy sets (1965). *Information and Control*, 8(3), (pp. 338-353). Academic Press.