

Aprendizaje computacional de mapas conceptuales en términos
de modelos gráficos probabilistas. Aplicación al diseño de un
sistema recomendador para la asignatura de Teoría de
Autómatas I

David Alfonso Gaspar

Página en blanco

Resumen

Las redes bayesianas permiten modelar la relación incierta entre diferentes variables aleatorias, con la ventaja sobre otros métodos de tener una representación gráfica y de permitir el uso de información de expertos. Por otra parte, los mapas conceptuales representan, también gráficamente, relaciones entre conceptos de un mismo dominio, sin apenas imponer restricciones sobre su estructura, lo que las ha llevado a ser una herramienta muy utilizada en entornos educativos.

Este trabajo propone un método semi-automático de generación de mapas conceptuales o competenciales a partir de redes bayesianas aprendidas automáticamente de respuestas de exámenes tipo test de una materia determinada. Nuestra hipótesis es que estas respuestas contienen conocimiento del dominio, el cual puede representarse de forma visual, teniendo una doble utilidad: en primer lugar, estos mapas pueden enriquecer a los creados manualmente por un experto, sugiriendo nuevas conexiones o subrayando la importancia de las ya identificadas; y, en segundo lugar, se propone la utilización de los mapas obtenidos como núcleo de un sistema recomendador que proporcione al alumno una representación gráfica de su conocimiento de la materia, ayudándole a localizar sus fortalezas y debilidades, así como planteándole actividades pedagógicas adecuadas a su nivel de conocimiento.

Para conseguir los objetivos anteriores, se han diseñado diversos experimentos y algoritmos que trabajan con los datos de resultados de exámenes tipo test de la asignatura de “Teoría de Automatas I” impartida en la UNED, aplicando diversas técnicas de Inteligencia Artificial y Minería de Datos. También se ha diseñado un prototipo web de sistema recomendador, basado en los mapas conceptuales generados en la primera fase del trabajo, con el fin de poder valorar su utilidad en un contexto de educación a distancia.

Índice

1. Introducción	6
1.1. Objetivos	6
1.2. Motivaciones	7
1.3. Metodología	8
1.4. Herramientas	8
1.5. Organización del trabajo	8
2. Estado de la cuestión	9
2.1. Mapas conceptuales	9
2.1.1. Definición y características	9
2.1.2. Generación semi-automática	9
2.1.3. Artículos relevantes	11
2.2. Redes bayesianas	13
2.2.1. Aprendizaje automático	14
2.2.2. Algoritmos de búsqueda heurística	16
2.2.3. Tratamiento de los valores ausentes	17
2.2.4. Artículos relevantes	18
2.3. Sistemas Recomendadores	19
2.3.1. Definición, tipos y aplicaciones	19
2.3.2. Artículos relevantes	20
2.4. Aportaciones al estado de la cuestión	22
3. Estudio y clasificación de exámenes tipo test	23
3.1. Nomenclatura de los exámenes	23
3.2. Preparación de los datos de respuestas	23
3.2.1. Datos en crudo	23
3.2.2. Datos originales	24
3.3. Análisis estadístico de las respuestas	24
3.3.1. Porcentajes y valores medios	24
3.3.2. Correlaciones entre las preguntas	25
3.4. Clasificación manual de preguntas	25
3.4.1. Conceptos de la materia	26
3.4.2. Categorías	26
3.5. Grafos de dependencias	28
4. Generación semi-automática de mapas conceptuales	31
4.1. Datos de entrada	31
4.1.1. Descripción de los datos originales	31
4.1.2. Atributos	32
4.1.3. Preprocesado de los datos originales	33
4.1.4. Filtrado de posibles fuentes de ruido	34
4.2. Aprendizaje estructural de redes bayesianas	34
4.2.1. Algoritmos	35

4.2.2.	Métricas de evaluación de redes bayesianas	36
4.2.3.	Experimentos con datos de competencias generales	37
4.2.4.	Experimentos con datos de competencias específicas	40
4.2.5.	Experimentos con datos de conceptos por tema	45
4.3.	Construcción de mapas conceptuales a partir de redes bayesianas	46
4.3.1.	Mapas conceptuales creados manualmente	46
4.3.2.	Paso 1: Algoritmo de simplificación de redes bayesianas	49
4.3.3.	Paso 2: Transformación de red bayesiana en mapa conceptual	50
4.3.4.	Paso 3: Asignación manual de frases de enlace	50
4.3.5.	Resultados	50
5.	Sistema recomendador genérico basado en mapas conceptuales generados semi-automáticamente por métodos bayesianos	53
5.1.	Análisis	53
5.1.1.	Arquitectura	53
5.1.2.	Funcionalidades	55
5.2.	Diseño	56
5.2.1.	Tipo de sistema recomendador	57
5.2.2.	Descripción de los ítems	57
5.2.3.	Perfil del alumno	57
5.2.4.	Recomendación de ítems de aprendizaje	59
5.2.5.	Visualización de mapas de conocimiento	60
5.3.	Prototipo	61
5.3.1.	Funcionalidad	64
5.3.2.	Evaluación	65
6.	Metodología general	68
7.	Conclusiones	70
7.1.	Principales aportaciones	70
7.2.	Problemas encontrados	70
7.3.	Trabajo futuro	71
A.	Datos estadísticos de las preguntas de los exámenes	77
B.	Correlaciones entre las preguntas de los exámenes	77
C.	Software para redes bayesianas	78
D.	Mapas conceptuales generados de forma semi-automática	85

1. Introducción

Los Mapas Conceptuales permiten representar y organizar el conocimiento de forma gráfica [NC08]. Su función original era facilitar el seguimiento de la evolución de las estructuras cognitivas de los niños durante su aprendizaje, aunque actualmente su uso se ha extendido a multitud de campos, muchos de ellos relacionados con la educación como, por ejemplo, la validación del conocimiento adquirido o como herramienta de aprendizaje [CFU03]. En los últimos años, ha surgido un especial interés por su construcción de forma automática a partir de conjuntos de datos, ya que su creación manual resulta laboriosa y depende de la existencia de expertos en la materia sobre la que verse el mapa [LLL09].

Por otra parte, las Redes Bayesianas son un modelo gráfico probabilista que representa en forma de grafo las relaciones de independencia entre un conjunto de variables aleatorias [Dí08]. Las redes bayesianas se utilizan en Inteligencia Artificial como herramientas para razonar ante situaciones de incertidumbre, utilizando las evidencias disponibles para inferir probabilidades de otras variables; es lo que se conoce como *propagación de la evidencia*. Su construcción puede ser manual o automática, siendo, en el segundo caso, una forma de aprendizaje automático que permite determinar la estructura del grafo asociado a la red bayesiana así como de sus parámetros. Las redes bayesianas tienen aplicaciones en medicina, ingeniería, informática, visión artificial y muchos otros campos, ya que permiten abordar problemas de diagnóstico, clasificación, predicción y control.

Los Sistemas Recomendadores están orientados a ayudar a los usuarios a encontrar ítems de su interés, dentro de un conjunto normalmente muy extenso (e.g. películas, páginas web, música, etc.) Los dos tipos de sistemas recomendadores más importantes son los colaborativos y los basados en ítems, siendo éstos últimos los de mayor interés para este trabajo, ya que se basan en los atributos de los ítems y en un modelo del usuario, para recomendar otros ítems a éste [CJFM09]. En el ámbito del Aprendizaje Mejorado por la Tecnología (en inglés, *Technology Enhanced Learning*), los sistemas recomendadores tienen diversas aplicaciones, desde la recomendación de secuencias de aprendizaje (dentro de un conjunto de recursos educativos) hasta el descubrimiento de cursos de interés para el alumno [MDV⁺11].

1.1. Objetivos

El primer objetivo de este trabajo consiste en la definición de un método de análisis de resultados de respuestas de cuestionarios de exámenes basado en modelos gráficos probabilísticos. La hipótesis de investigación es que estas respuestas de exámenes encapsulan un conocimiento del dominio de aprendizaje que se puede representar gráficamente mediante redes bayesianas, permitiendo a los docentes obtener información de las relaciones entre los conceptos y competencias de una materia determinada, por ejemplo, determinando aquellos que presentan más dificultades o cuyo conocimiento es más importante dentro de la misma. Dos nodos enlazados en la red sugerirán que los alumnos suelen mostrarse competentes, o no, en ambas competencias.

Posteriormente, se diseñará un procedimiento de generación semi-automática de mapas conceptuales, utilizando las redes bayesianas obtenidas anteriormente. Estos mapas conceptuales estarán formados por conceptos o competencias de aprendizaje de la materia en cuestión, y podrán ser utilizados para enriquecer otros generados de forma manual por un experto, su-

giriendo nuevas conexiones o subrayando la importancia de las ya identificadas.

Como segundo objetivo, se estudiará la utilización de los resultados anteriores en entornos digitales educativos; en particular, se diseñará un sistema recomendador introspectivo que proporcione un soporte personalizado a los estudiantes mediante una herramienta visual, como es un mapa conceptual, tanto de su conocimiento del dominio, como de sus fortalezas y debilidades en las competencias de la materia; permitiendo al alumno realizar un análisis de su propio conocimiento. Además, se recomendarán al alumno actividades de aprendizaje que le permitan mejorar sus conocimientos y competencias en las partes de la materia donde el alumno tenga más debilidades. La hipótesis en este punto es que los ítems recomendados serán adecuados al nivel de conocimiento del alumno y le guiarán en el adecuado aprendizaje de la materia.

Finalmente, se sintetizarán las conclusiones de los puntos anteriores en forma de metodología general, de forma que se facilite la aplicación de estas técnicas en otros dominios de conocimiento.

1.2. Motivaciones

A los docentes con frecuencia les cuesta ver las dificultades de los alumnos e identificar los caminos de aprendizaje más intuitivos, estructurando los programas docentes según criterios convencionales basados en la perspectiva de los expertos conocedores de la materia. Los mapas conceptuales utilizados en docencia muestran también usualmente esta perspectiva. Sería de gran interés para la comunidad docente disponer de mapas conceptuales que reflejasen los caminos de aprendizaje más naturales para un aprendiz que desconoce la materia, y tener la posibilidad de generar estos mapas automáticamente.

Por otro lado, las pruebas de evaluación realizadas por los alumnos llevan implícita información valiosa sobre las fortalezas y debilidades de los estudiantes (individuales y colectivas) en lo que respecta a las competencias en la materia y sobre los distintos vínculos que relacionan estas competencias. Son pues de gran utilidad para definir estrategias pedagógicas y caminos de aprendizaje. Sin embargo, la interpretación de estas pruebas puede ser difícil y laboriosa cuando se dispone de un gran número de alumnos. Es claro el interés de proporcionar a los docentes herramientas que les permiten extraer y explotar este conocimiento de forma semi-automática.

La obtención de mapas conceptuales para su uso en sistemas recomendadores tiene, finalmente, gran interés, al tratarse de una herramienta de uso común en los ámbitos educativos. Además, de acuerdo a la opinión de los docentes en la materia consultados, uno de los problemas más importantes del aprendizaje de la teoría de autómatas es que los alumnos van avanzando en el estudio sin haber afianzado las competencias necesarias en cada paso.

Por otra parte, los métodos probabilistas tienen interesantes aplicaciones en el campo de los sistemas educativos adaptativos en línea, pudiendo dar soporte a la adaptación de las estrategias pedagógicas en función de datos de interacción de los alumnos con los sistemas, o de repositorios de datos que contengan información académica de un gran número de alumnos. En el caso de este proyecto los métodos probabilistas se aplicarán a repositorios de respuestas a cuestionarios de examen.

1.3. Metodología

La experimentación realizada en este trabajo se basa en la asignatura de “Teoría de Autómatas I” impartida en la UNED dentro de la carrera de Ingeniería Informática (extinguida el pasado curso académico, 2012/2013), ya que se cuenta con expertos en dicha materia y con los resultados de cuestionarios de exámenes de varios años. Se trabajará de forma inductiva, buscando validar en un dominio concreto las hipótesis descritas anteriormente, para finalmente establecer un procedimiento general de construcción automática de mapas conceptuales y sistemas recomendadores.

Como etapa final, se planteará la creación de un prototipo de sistema recomendador web que demuestre el funcionamiento descrito en el trabajo, así como sus posibles aplicaciones.

1.4. Herramientas

A lo largo del trabajo se describen las herramientas utilizadas en cada una de las etapas, siendo las principales: el lenguaje de programación Python [VRF03], que facilita el procesado de cadenas y la reutilización de código; el lenguaje R [IG96], que dispone de una gran cantidad de módulos para análisis estadístico de datos; y el lenguaje GNU/Octave [Eat08], que proporciona una sintaxis similar a MATLAB, lo que permite trabajar con matrices de forma sencilla y potente.

Para la realización del prototipo de sistema recomendador, se han reutilizado la mayoría de las clases realizadas en Python para los experimentos de generación de mapas conceptuales a partir de redes bayesianas, integrándolas en la plataforma Google App Engine [Zah09], que permite realizar un rápido despliegue de aplicaciones web aprovechando la infraestructura de computación en la nube de Google.

1.5. Organización del trabajo

En la sección 2, se realiza una revisión del estado de la cuestión en torno a: mapas conceptuales y su construcción automática, aprendizaje automático de redes bayesianas (prestando especial atención al uso en la literatura de exámenes tipo test), y sistemas recomendadores (especialmente en el ámbito educativo). En la sección 3, se describen los datos en crudo de los que se ha partido en este trabajo, cómo se han preprocesado para poder utilizarlos posteriormente, los resultados de un análisis estadístico simple y, por último, la clasificación manual de las preguntas de los tests de examen, incluyendo los grafos de dependencias entre competencias de la asignatura construidos. Posteriormente, en la sección 4, se describe el método propuesto para la generación de mapas conceptuales a partir de redes bayesianas aprendidas de forma semi-automática. La sección 5 describe un sistema recomendador genérico basado en mapas conceptuales, que tiene el objetivo de guiar al alumno en el aprendizaje y la práctica de una materia; se detalla también el prototipo realizado para la asignatura de “Teoría de Autómatas I”. La sección 6 describe una metodología para aplicar las técnicas descritas en este trabajo a otras asignaturas. Finalmente, la sección 7 describe las conclusiones del trabajo y las líneas de investigación que han quedado abiertas.

2. Estado de la cuestión

2.1. Mapas conceptuales

2.1.1. Definición y características

Los mapas conceptuales son una herramienta gráfica que permite representar y organizar el conocimiento, facilitando el aprendizaje [NC08]. Descritos originalmente por Joseph D. Novak en su libro “Learning How to Learn” [NG84], se planteaban como una herramienta educativa para facilitar tanto el aprendizaje como la enseñanza, impulsando el pensamiento basado en conceptos y preposiciones (entendidos como regularidades) y su expresión visual en forma de mapas conceptuales.

Un mapa conceptual suele estar formado por: *conceptos*, que a menudo consisten en palabras o frases rodeadas por un círculo, y *enlaces*, representados por líneas que conectan dos conceptos. A su vez, los enlaces pueden tener *frases de enlace*, que describen la relación entre los conceptos.

Sobre los elementos básicos de un mapa conceptual, se establece el concepto de *preposición*, que consiste en una frase u oración, formada por dos o más *conceptos* unidos por sus *frases de enlace*, que tiene un significado completo. En la figura 1 se muestra un ejemplo de mapa conceptual que contiene todos los elementos descritos.

A continuación, se listan algunas características generales de los mapas conceptuales:

- Giran en torno a un dominio de conocimiento acotado, que suele definirse mediante una *pregunta de enfoque*. Son el reflejo de la estructuras de conocimiento que tiene la persona que ha realizado el mapa sobre el tema; por lo que resulta difícil que dos personas realicen el mismo mapa conceptual, lo que contribuye a la dificultad en su creación de forma manual, aun por expertos en el dominio [KCB10].
- Suelen tener una *estructura jerárquica* con los conceptos más generales arriba y los más concretos abajo, aunque no tiene porqué ser así; de hecho, es posible añadir enlaces entre nodos de diferentes zonas de un mapa conceptual: los denominados *enlaces cruzados*. Otras topologías posibles son las circulares, en árbol, en red y otras [VLC08].
- La única restricción a la hora de crear un mapa conceptual es que las *frases de enlace* tengan sentido (en caso de estar presentes), es decir, que se puedan formar *proposiciones* correctas. Por lo demás, no hay apenas restricciones formales en cuanto a la estructura que pueden tener [NC08].
- Se utilizan mayoritariamente para presentar información en entornos educativos con el fin de mejorar el proceso de aprendizaje o incluso de evaluarlo [CFU03].

2.1.2. Generación semi-automática

Uno de los objetivos de este trabajo es la generación automática de mapas conceptuales, por lo que se han investigado los diversos trabajos realizados hasta la fecha en este área. En [KCB10] se describe el estado de la cuestión de la generación automática de mapas conceptuales y se plantean tres criterios a utilizar cuando se estudia un método de generación

conceptuales [GG08].

Resultados Desde el punto de vista de los resultados obtenidos, los mapas conceptuales generados pueden clasificarse según:

- Su *estructura*, que se refiere al tipo de estructura del grafo generado. Por ejemplo, en forma de árbol.
- Si son *completos o incompletos*, que corresponde a que los nombres de los nodos sean comprensibles y las preposiciones existan y estén formadas adecuadamente.
- El *método de construcción* utilizado, que puede ser automático, semi-automático o manual; hace hincapié en la implementación del mecanismo de generación, i.e. si es independiente de herramientas de terceros, o, si no lo es, qué nivel de interoperabilidad tiene con ellas.
- La *evaluación* realizada, que puede ser subjetiva u objetiva. La primera se basa en el criterio de expertos humanos, mientras que la segunda lo hace en criterios objetivos asociados a las proposiciones, niveles de jerarquía, cantidad de ramificaciones, *cross-links* (enlaces entre distintos segmentos de un mapa conceptual) u otros de interés.

Será interesante clasificar los mapas generados en este trabajo teniendo en cuenta los criterios anteriores, a excepción del método de construcción, más preocupado de la utilidad y facilidad de uso del software en entornos reales, que de los propios algoritmos de generación. Especialmente, en este trabajo se planteará el uso de una evaluación objetiva basada en la clasificación topológica descrita en [VLC08].

Técnicas de generación de mapas En los métodos analizados en [KCB10], se utilizan diversas técnicas estadísticas y de aprendizaje por computador, como asociaciones apriorísticas, procesado de lenguaje natural, clusterización de palabras en documentos o reglas difusas, pero en ninguno se plantea el uso de redes bayesianas como estructura intermedia entre los datos y los mapas conceptuales.

2.1.3. Artículos relevantes

Generación de mapas conceptuales a partir de ontologías

En [GG08] se describe un algoritmo que genera mapas conceptuales a partir de ontologías OWL ya existentes. En cierta forma, existe una similitud con este trabajo, si las redes bayesianas se construyeran de forma manual por un experto, como es el caso de las ontologías del artículo referenciado.

Las ontologías tienen similitudes con los mapas conceptuales, ya que ambos representan estructuras de conocimiento. Sin embargo, aunque las ontologías son a priori más potentes, los mapas conceptuales tienen la ventaja de ser más sencillos de construir, ya que las primeras requieren aprender lenguajes formales y herramientas específicas.

El resultado del trabajo descrito es un conjunto de mapas conceptuales con preposiciones, ya que aprovecha los diferentes tipos de relaciones que se describen en las ontologías que

utiliza como base del algoritmo de construcción. En la figura 2 se muestra uno de los 6 mapas conceptuales correctos que generaron en este artículo, de un total de 10 ontologías, ya que con 4 tuvieron diversos problemas con la sintaxis OWL.

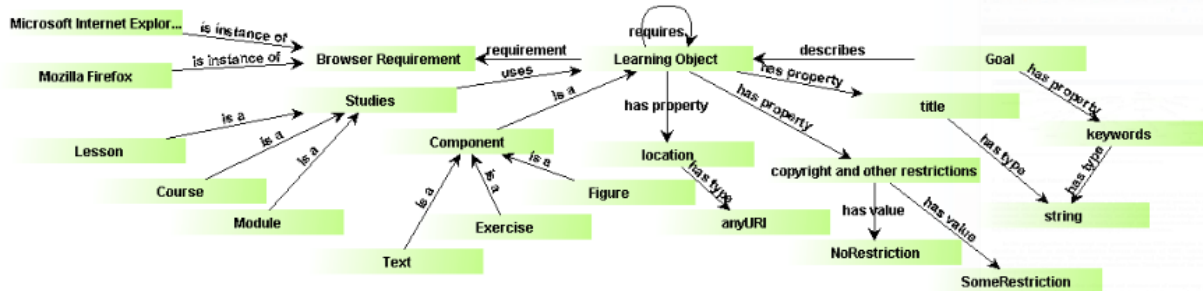


Figura 2: Mapa conceptual mostrado como resultado en [GG08] tras aplicar el método descrito a una ontología relacionada con el campo de las ciencias de la computación.

Construcción basada en reglas difusas para sistemas de aprendizaje adaptativos

Un sistema adaptativo es un sistema interactivo que adapta su comportamiento a cada usuario a partir de la adquisición de un modelo del usuario y la aplicación de algún tipo de mecanismo de inferencia, aprendizaje o toma de decisiones [SJ07].

En [BC08] se presenta un método para construir de forma automática mapas conceptuales utilizando reglas difusas y registros de tests de los estudiantes. Se evalúan los grados de relevancia de las relaciones entre los diferentes conceptos y se plantea la posible utilidad de este método en sistemas de aprendizaje adaptativos.

En dicho método, cada pregunta puede tener uno o más conceptos asociados, indicando que este concepto es relevante para responder a la pregunta. El método comienza por utilizar conjuntos difusos para obtener una tabla que relaciona los resultados de cada alumno en cada pregunta con los del resto de alumnos. A partir de esta tabla, aplicando un razonamiento difuso (i.e. una serie de reglas difusas), se determina el grado de relación entre cada par de preguntas. Finalmente, para crear el mapa conceptual, se crea en primer lugar un grafo de preguntas, en el que todos los nodos están relacionados con el resto formando un grafo completo; la dirección entre cada par de preguntas se establece en función de la nota media de los alumnos en cada una (desde la que tiene más nota a la que tiene menos) y el valor asociado al enlace es igual al grado de relación entre pares de preguntas calculado en el paso anterior. Para transformar el grafo de preguntas en un mapa conceptual se hace simple sustitución directa de preguntas por los conceptos asociados a las mismas, eliminando enlaces duplicados calculando su media.

En la figura 3, se muestra un mapa conceptual entre conceptos obtenido aplicando el método anterior a un conjunto de datos artificial y muy reducido de 5 preguntas y 5 conceptos.

Comentarios En primer lugar, el artículo no menciona cómo podría aplicarse su método a un sistema adaptativo; sólo sugiere que puede ser de utilidad en este tipo de sistemas. Además, el hecho de no aplicarlo a datos reales hace pensar que su comportamiento no es adecuado o que resulta demasiado difícil de realizar.

En cuanto al algoritmo, la elección del criterio para establecer la dirección de los enlaces, que se basa en qué pregunta ha tenido una nota media más alta, no se explica en el artículo, por lo que resulta un tanto arbitraria. Además, el algoritmo está limitado a conjuntos de datos sin valores ausentes, lo que supone una limitación importante a la hora de agregar resultados de exámenes realizados por diferentes alumnos; como veremos a lo largo del presente trabajo, donde sí se tiene en cuenta esta posibilidad y se plantean diversas soluciones.

Por otro parte, los mapas conceptuales generados no tienen frases de enlace, lo cual es muy común en los métodos de generación revisados en este trabajo, ya que su creación suele depender de su existencia previa, como vimos en el artículo comentado en la sección anterior. En nuestro trabajo se da una guía para generarlos de forma manual.

Finalmente, el hecho de trabajar con reglas y valores difusos hace que las etapas intermedias del algoritmo sean más difíciles de visualizar y, por tanto, de comprender por un humano; por el contrario, en nuestro trabajo se utilizan redes bayesianas, lo que permitirá mostrar visualmente estos resultados intermedios.

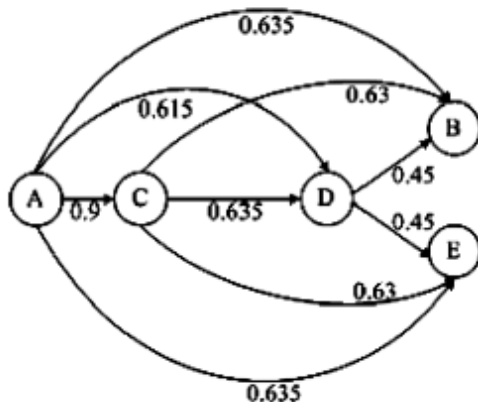


Figura 3: Mapa conceptual mostrado como resultado en [BC08] tras aplicar el algoritmo descrito en ese mismo artículo a un conjunto de datos artificial formado por 5 preguntas, 5 conceptos (A, B, C, D y E) y 10 alumnos.

2.2. Redes bayesianas

Las redes bayesianas permiten modelar el conocimiento de un determinado dominio, representando las dependencias probabilísticas entre las variables que lo componen [KN11]. Una vez construidas, permiten calcular la probabilidad a posteriori de una o más variables, llamadas “variables de interés”, condicionada a cierta evidencia, formada por las “variables observadas”; es lo que se conoce como propagación de la evidencia. En el ámbito de este trabajo, el interés se centrará en las relaciones de dependencia e independencia que las redes bayesianas definen entre sus diferentes variables.

Definición Las redes bayesianas [Dí08, RN10] son un modelo gráfico probabilista que consta de tres elementos: un conjunto de variables aleatorias (discretas o continuas), X ; un grafo

dirigido acíclico $\mathcal{G} = (X, \mathcal{A})$, en que cada nodo representa una variable aleatoria X_i ; y una distribución de probabilidad sobre X , $P(X)$, que puede ser factorizada así:

$$P(x) = \prod_i P(x_i | pa(X_i))$$

donde las $P(x_i | pa(X_i))$ son las probabilidades condicionadas de cada nodo con respecto a sus padres, que se obtienen a partir de $P(X)$.

Mapas de independencia Hemos visto que la distribución de probabilidad P se puede factorizar según el grafo dirigido acíclico G . Además, sabemos que esta propiedad es equivalente a decir que G es un mapa de independencias de P , es decir, que para todo trío de subconjuntos $\{A, B, C\}$ de X se cumple:

$$I_G(A, B|C) \implies P(a, b|c) = P(a|c) \cdot P(b|c)$$

Esto quiere decir que si dos subconjuntos A y B están *separados direccionalmente* [Pea88, Pea00], es decir, no existe ningún camino activo entre ellos, entonces son condicionalmente independientes en sentido probabilista.

Interpretaciones probabilista y causal del grafo Los enlaces de una red bayesiana pueden tener una interpretación causal o probabilista: en la primera, un enlace $A \rightarrow B$ significa que existe un mecanismo causal (determinista o no determinista) que hace que la presencia de A produzca B ; en la segunda, el grafo se entiende como un mapa de independencias de la distribución de probabilidad, donde la dirección de los enlaces no es importante, ya que representan una correlación matemática [Dí08]. Por tanto, podrá haber grafos equivalentes desde el punto de vista probabilista, pero no desde el punto de vista causal.

Grafos equivalentes en sentido probabilista Lo que se conoce en inglés como *Markov equivalence* quiere decir que dos modelos causales (redes bayesianas) son equivalentes en sentido probabilista si y solo si contienen las mismas variables y cualquier distribución de probabilidad que se pueda representar con uno, se puede representar también con el otro [KN11]. Se llama clase de equivalencia de Markov a cada uno de los grupos de grafos que son equivalentes en sentido probabilista.

Por ejemplo, los grafos $A \rightarrow B \rightarrow C$, $A \leftarrow B \leftarrow C$ y $A \leftarrow B \rightarrow C$ son equivalentes en sentido probabilista, ya que pueden representar las mismas distribuciones de probabilidad y las mismas relaciones de independencia. Sin embargo, como hemos visto anteriormente, la interpretación causal de estos grafos es muy diferente. Es por esta razón que deberemos prestar una especial atención al tipo de aprendizaje que realizan los distintos algoritmos que veremos a continuación, ya que no será lo mismo trabajar con clases de equivalencia que tener en cuenta la dirección de los enlaces, a la hora de utilizar el aprendizaje bayesiano en nuestro trabajo.

2.2.1. Aprendizaje automático

Hay dos formas de construir una red bayesiana: de forma manual o mediante aprendizaje automático. En la primera, es un experto en el dominio el que establece las relaciones entre

los nodos y los parámetros; en la segunda, se utilizan bases de datos y algoritmos para estimar la estructura de la red y sus parámetros.

El aprendizaje automático trata de construir el modelo que mejor represente la realidad de los datos [Dí08]. Existen dos tipos distintos de aprendizaje:

1. **Aprendizaje estructural**, que se refiere a las dependencias e independencias entre las variables, es decir, a la estructura gráfica de la red.
2. **Aprendizaje paramétrico**, que se refiere al aprendizaje de los valores de las probabilidades condicionadas de la red.

En función del problema, será necesario sólo el primero, el segundo (siempre que se conozca la estructura de la red) o ambos. En este trabajo, nos vamos a centrar en el primer tipo, ya que nuestro objetivo es aprender de forma automática estructuras de redes bayesianas y transformarlas en mapas conceptuales de un determinado dominio de conocimiento.

Tipos de aprendizaje estructural Existen dos tipos principales [KN12]: los basados en tests de independencia (en inglés se llaman “constraint-based algorithms”), como el algoritmo PC [SGS93], y los basados en búsqueda heurística y métricas de calidad, como K2 [CH92].

Los algoritmos basados en tests de independencia buscan relaciones de dependencia mediante técnicas estadísticas y se utilizan para construir la red. Estos algoritmos evitan la complejidad de las redes aprendidas de forma explícita mediante umbrales (e.g. máximo número de padres por nodo) y suelen ser muy intensivos en cálculo y poco escalables a redes grandes.

Los basados en búsqueda heurística, utilizan este tipo de búsqueda para recorrer el espacio de estados en busca del grafo que optimice una determinada *métrica* (medida de calidad de la red). Estos algoritmos tienen un mayor potencial para evitar el sobreajuste gracias al uso de métricas, que suelen incorporar términos que favorecen las redes menos complejas. Por otro lado, la mayoría de éstas asignan el mismo valor a redes equivalentes en sentido probabilista, lo que dificultará la interpretación de los enlaces de la red por un experto. Otro ejemplo de algoritmo de este tipo es el conocido como Hill Climbing (HC) [Mar03], que será utilizado en este trabajo.

Otro tipo de algoritmos son los conocidos como híbridos, que tratan de aprovechar los puntos fuertes de los dos tipos principales. Un algoritmo de este tipo es el conocido como Max-Min Hill Climbing (MMHC) [TBA06], que también será utilizado en este trabajo.

Métricas y clases de equivalencia Ejemplos de métricas son: K2 [CH92] y BIC (Bayesian Information Criterion) [Sch78]. Las métricas utilizadas en los algoritmos de búsqueda heurística suelen realizar una interpretación probabilista de la estructura de las redes bayesianas, de forma que todos los grafos que están en la misma clase de equivalencia representan las mismas relaciones de independencia. En concreto, la métrica BIC, que se utilizará en este trabajo, es de este tipo, por lo que los arcos de las redes aprendidas no deberán ser interpretados en sentido causal, ya que la métrica no distingue entre diferentes redes dentro de la misma clase de equivalencia. Además, esta métrica tiene la característica de favorecer redes más sencillas, evitando así el sobreajuste.

Aprendizaje de redes causales Bajo determinadas condiciones de aprendizaje [Pea00], se puede realizar una interpretación causal de los enlaces de las redes bayesianas [TBA06]. Debido a esta propiedad, éstas se han utilizado en diversos estudios para obtener hipótesis causales, por ejemplo en [NDS07] se utiliza para la búsqueda de causas globales en espacios con alta dimensionalidad. Sin embargo, existe mucha controversia en cuanto a la posibilidad de obtener redes causales simplemente utilizando datos observacionales (i.e. datos sobre los que no se ha intervenido).

En lo que respecta a este trabajo, trataremos los resultados del aprendizaje estructural de redes bayesianas desde el punto de vista probabilístico, suponiendo que los enlaces representan correlaciones entre variables y no mecanismos causales. Por esta razón, la interpretación de la causalidad de los enlaces quedará en manos de los expertos que, en el momento de establecer las frases de enlace de los mapas conceptuales, deberán revisar también la orientación de los enlaces.

2.2.2. Algoritmos de búsqueda heurística

Hill Climbing (HC) En primer lugar, el algoritmo voraz de búsqueda heurística (en inglés, “score-based”) Hill Climbing (HC) [Mar03] es un algoritmo sencillo que resulta útil cuando no hay expertos en el dominio que conozcan la estructura, así como para realizar descubrimiento del dominio (i.e. el caso planteado en este trabajo).

El algoritmo realiza una búsqueda sobre el estado de posibles redes partiendo de una red generada al azar, realizando en cada iteración una modificación sencilla (e.g. añadir un enlace, eliminarlo o cambiar la orientación) y continuando con aquella que obtenga un mejor resultado para la métrica utilizada. Como se puede observar, este algoritmo no asegura que encuentre un máximo global, aunque existen diferentes técnicas para intentar evitar caer en máximos locales (e.g. realizar perturbaciones aleatorias en la red).

De las muchas disponibles, se utilizará la métrica BIC (Bayesian Information Criterion) [Sch78], que tiene la característica de favorecer redes más sencillas evitando el sobreajuste, aunque no distingue entre redes dentro de la misma clase de equivalencia.

Algoritmos híbridos: Max-Min Hill Climbing El algoritmo Max-Min Hill Climbing (MMHC) [TBA06] es un algoritmo de tipo híbrido, que realiza una primera fase de búsqueda del esqueleto de la red (i.e. sin dirección en los enlaces) mediante un algoritmo llamado Max-Min Parents and Children (MMPC) que determina los padres y los hijos de cada nodo de forma local mediante tests de independencia; y una segunda, en la que realiza una búsqueda heurística utilizando el algoritmo Hill Climbing y la métrica BIC acotada al esqueleto obtenido en la primera fase.

Este algoritmo ha sido utilizado en problemas con alta dimensionalidad y con el objetivo de descubrir relaciones causales en los datos [KN11, NDS07]. Al ser un algoritmo *híbrido*, trata de aprovechar las ventajas de los dos tipos principales, es decir, de búsqueda heurística y de tests de independencia. Además, no requiere una ordenación inicial de los nodos, por lo que resulta más sencillo de utilizar que otros algoritmos, obteniendo resultados iguales o mejores, en un tiempo de ejecución mucho menor [AN11].

2.2.3. Tratamiento de los valores ausentes

Los valores ausentes son un problema ámpliamente tratado en el campo de la minería de datos y la inferencia estadística [Pig01], ya que éstos pueden aparecer en cualquier conjunto de datos, independientemente del tratamiento posterior que se vaya a realizar con ellos. Muchos de los estudios al respecto realizados durante la segunda mitad del siglo XX estaban relacionados con las ciencias sociales, donde este problema resulta muy común, e.g. cuestionarios incompletos.

Los conjuntos de datos que se utilizan en el trabajo presentan en la mayoría de ocasiones valores ausentes, por lo que se han estudiado las diferentes alternativas disponibles a la hora de realizar el aprendizaje de redes bayesianas en su presencia.

Tipos de datos ausentes En función de los mecanismos que hayan dado lugar a la ausencia de los datos, se puede hablar de: *missing completely at random (MCAR)*, *missing at random (MAR)* y *not missing at random (NMAR)* [LR02]. En MCAR, la ausencia de los datos no se asocia con otros datos del conjunto o fuera del conjunto, mientras que en MAR se supone que como mucho las ausencias se derivan de datos presentes en el conjunto (pero no fuera); y, por último, NMAR en la que existe algún motivo para los valores ausentes que no está presente en los datos. La mayoría de métodos para tratar los valores ausentes trabajan con la hipótesis de que estos son MCAR o, al menos, MAR.

En nuestros conjuntos de datos, podemos suponer que los valores ausentes se producen totalmente al azar (MCAR), ya que se derivan de que el examen que realizó un alumno tuviera o no preguntas con determinadas categorías o conceptos.

Alternativas Existen varias opciones para tratar los valores ausentes en el aprendizaje de redes bayesianas [Hec08, LH08]:

1. *Eliminar las instancias con valores ausentes*, aunque es una opción sencilla, da lugar a pérdida de información y si muchas de las instancias tienen valores ausentes, resulta inaplicable.
2. *Imputar los valores ausentes* utilizando algún método de imputación que puede ser unidimensional, como la media, o multidimensional, como los métodos basados en distancias (e.g. *k-nearest neighbor*) o en covarianzas [TKF11]. Además, cuando un método de imputación puede tener en cuenta la incertidumbre inherente a los datos, se puede utilizar para realizar imputación múltiple [VB12], donde se realizan m imputaciones para cada valor ausente, dando lugar a m conjuntos de datos que muestran la incertidumbre existente en cuanto al valor que esos datos tienen realmente. Posteriormente, se trabaja con dichos conjuntos de forma independiente y se combinan los resultados obtenidos en uno solo.
3. *Aprender la estructura de la red bayesiana utilizando los datos con valores ausentes*. Uno de los métodos más conocido es el llamado *Structural EM* (SEM) [FGG97], basado en el uso del algoritmo *Expectation-Maximization* (EM) para calcular el valor de la métrica asociada a la estructura de la red en cada iteración de la búsqueda heurística.

2.2.4. Artículos relevantes

Redes bayesianas para construir ontologías y validar conocimiento

En [CDS10] se presenta un método para construir ontologías “ligeras” (i.e. sólo incluyen conceptos y relaciones) orientado a dominios de conocimiento similares a los que tiene una asignatura o un curso universitario, como la que vamos a utilizar en este trabajo, y también en un contexto de educación a distancia.

Se propone un algoritmo basado en que los profesores construyan de forma manual una ontología y, a partir de ésta, se realice un mapeo directo para construir una red bayesiana con los mismos nodos que la ontología, donde cada nodo puede tomar el valor Sí o No, según el alumno conozca o no la respuesta a la pregunta. Después, utilizando resultados de cuestionarios de alumnos, se determina si los enlaces propuestos por los profesores tienen un valor de entropía cruzada alto, lo que indicará si las relaciones se ven más o menos confirmadas por los resultados de los exámenes. La entropía se calcula para todo los posibles pares de nodos, con lo que es posible sugerir a los profesores nuevos enlaces en la ontología.

Basándose en la ontología construida y en la red bayesiana equivalente (realizada mediante mapeo directo), se plantea su uso en una herramienta de validación del conocimiento, que las utiliza para inferir el estado de conocimiento de los alumnos a partir de cuestionarios realizados a lo largo del curso. Esto permite conocer a los profesores el conocimiento actual de los alumnos y, al sistema, proponer preguntas y recursos de aprendizaje en función de este conocimiento.

El artículo presenta resultados basados en el uso o no del sistema propuesto por alumnos a lo largo del curso de una asignatura y concluye que los aquellos que lo utilizaron obtuvieron mejores resultados en el examen final.

Comentarios El algoritmo que proponen se aplica a ontologías de reducido tamaño (4-8 conceptos) que corresponden a temas de diferentes asignaturas. La validación se realiza con entre 300 y 500 cuestionarios, aunque no sabemos de cuántas preguntas cada uno. En cambio, nosotros planteamos utilizar ontologías (grafos de dependencias) de muchos más conceptos (desde 9 hasta casi 40) y utilizar los resultados de exámenes de fin de curso, en lugar de cuestionarios realizados ad-hoc, lo que dará lugar a dificultades en la fusión de dichos resultados, como veremos a lo largo del trabajo.

Por otro lado, el artículo plantea las redes bayesianas como una herramienta auxiliar para validar una ontología y, posteriormente, como representación del modelo del usuario en un sistema tutor inteligente [SB82]. Sin embargo, es interesante comentar que en otros artículos más antiguos [CDSV04], los mismos autores planteaban el aprendizaje automático de ontologías utilizando aprendizaje estructural de redes bayesianas, mediante el uso de varios algoritmos de aprendizaje estructural (todos ellos basados en tests de independencia) cuyos resultados se fusionaban en una única red mediante un algoritmo de votación. Además, en este caso sí que planteaban el uso de exámenes finales. Dado que el artículo comentado tiene sólo dos páginas (y no se ha encontrado otro más extenso), suponemos que los resultados fueron poco satisfactorios y que este camino fue abandonado, optándose por la construcción manual de las ontologías. Aunque el enfoque de nuestro trabajo puede ser parecido, nosotros plantearemos el uso de algoritmos de aprendizaje estructural más recientes que permiten trabajar con más variables y más datos, además de evaluar diversos tipos de preprocesado y

agregación de los datos para determinar cuáles dan lugar a mejores resultados.

2.3. Sistemas Recomendadores

2.3.1. Definición, tipos y aplicaciones

Los sistemas recomendadores buscan ayudar al usuario a encontrar ítems de su interés dentro de amplios catálogos [CJFM09]. Su popularidad ha crecido gracias a su uso en populares comercios electrónicos, como Amazon¹, donde los usuarios reciben recomendaciones de productos en función de sus compras anteriores, los productos visitados, etc. Sin embargo, su ámbito de aplicación es mucho más extenso y los ítems puede ser casi cualquier cosa, desde películas de cine hasta páginas web.

Tipos de sistemas recomendadores Los principales tipos de sistemas recomendadores son los siguientes:

- *Filtrado colaborativo*, donde dados los gustos del usuario (i.e. qué ítems prefiere), se puede determinar cuánto apreciara un nuevo ítem, sabiendo lo que usuarios con gustos parecidos han opinado sobre éste. Otra alternativa es recomendar ítems similares a los que el usuario ya ha mostrado interés.
- *Filtrado basado en el contenido*, donde las recomendaciones se basan tanto en una descripción de los ítems como en un perfil de los intereses del usuario. Sobre esta información, se establece un método para comparar ítems con perfiles, de forma que se determinen los ítems más adecuados. Para representar el modelo del usuario, se pueden usar desde vectores con pesos hasta reglas inductivas. Otra alternativa es aprender estos modelos usando técnicas de aprendizaje automático.
- *Filtrado híbrido*, en el que se combina información de los dos tipos anteriores.

Sistemas recomendadores en educación Como hemos dicho, el ámbito de aplicación de un sistema recomendador es muy extenso, así que no es de extrañar que hayan tenido mucha aceptación en entornos educativos. Por ejemplo, en lo que se conoce como Aprendizaje Mejorado por la Tecnología (en inglés, “Technology Enhanced Learning”), se han utilizado para mejorar el aprendizaje, tanto de individuos como de instituciones [MDV⁺11].

El contexto del sistema recomendador es muy relevante, ya que no es lo mismo que se enmarque dentro de un curso presencial o semi-presencial, que en uno totalmente online. También es importante tener en cuenta si el curso lo imparte una institución educativa, como una universidad, o si tiene un carácter más informal, donde la evaluación podría no ser tan importante o incluso no existir.

A la hora de diseñar un sistema recomendador es importante plantearse adecuadamente la meta que se quiere conseguir. Las principales tareas donde se han aplicado son: recomendar ítems de aprendizaje dentro de un curso online, otros cursos online, recursos sobre un tema o proponer una secuencia de aprendizaje para lograr un objetivo. El entorno educativo tiene consideraciones diferentes a las que puede tener uno comercial, como, por ejemplo, se puede buscar el aprendizaje de nuevos ítems o el refuerzo de los ya conocidos.

¹<http://www.amazon.es>

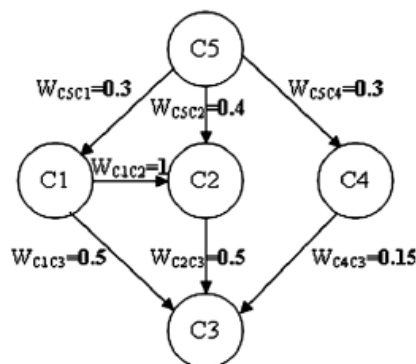


Figura 4: Mapa conceptual aprendido a partir de relaciones apriorísticas derivadas de resultados de preguntas con el algoritmo descrito en [LLL09]. C_i significa “Concepto i ” y $W_{C_iC_j}$ hace referencia al peso del enlace entre C_i y C_j .

2.3.2. Artículos relevantes

Recomendación de conceptos a partir de mapas conceptuales

En [LLL09], se describe un sistema de diagnóstico inteligente de conceptos (ICDS) que se basa en la creación dinámica de un mapa conceptual a partir de resultados de preguntas tipo test, utilizando técnicas de aprendizaje apriorístico, típicas de la minería de datos; Además, el sistema realiza recomendaciones personalizadas de conceptos de estudio a los alumnos, basándose en el algoritmo RIP (Remedial Instruction Path o *ruta de instrucción correctiva*) [CLW01] utilizando el mapa conceptual creado previamente.

El algoritmo planteado es el siguiente: los conceptos se asocian manualmente a las preguntas por los profesores, utilizando un peso para cada concepto asignado, lo que permite darles más o menos importancia. Después se aplica un algoritmo apriorístico, llamado “Apriori”, a los datos de resultados de preguntas, dando lugar a una serie de reglas asociativas que establecen relaciones entre preguntas. Posteriormente, dichas reglas son transformadas para que en lugar de relacionar preguntas relacionen conceptos, asignándoles además un peso basado en la clasificación manual realizada por el profesor. Finalmente, se construye el mapa conceptual haciendo corresponder cada regla con un enlace del mapa, eliminando los ciclos que puedan aparecer, prefiriendo aquellos enlaces que tengan más fuerza.

A partir del mapa conceptual construido, el sistema propone a cada alumno el estudio de una serie de conceptos utilizando el algoritmo RIP, que calcula el camino de aprendizaje más adecuado dados los conceptos en los que ha fallado. En la figura 4, se muestra uno de los mapas conceptuales aprendidos por el sistema.

El sistema ha sido evaluado en un entorno real con 245 alumnos, llegando a las siguientes conclusiones: las preguntas demasiado fáciles o difíciles daban lugar a un mapa conceptual demasiado simple o complejo, respectivamente, que no representaba el estado real del aprendizaje del alumno; las preguntas de memorizar tampoco daban lugar a mapas conceptuales representativos; y, por último, la mayoría de los grupos con los que probaron el sistema tuvieron un progreso significativo en su aprendizaje, siendo los alumnos que tenían un menor conocimiento los más beneficiados de disponer la ruta de instrucción correctiva.

Comentarios En cuanto al proceso de creación del mapa, el algoritmo no plantea la existencia de valores ausentes en los datos, por lo que para cada conjunto de alumnos que realiza los tests, se generará un mapa diferente. Sin embargo, en nuestro trabajo sí se tiene cuenta este caso, lo que permitirá agregar datos de diferentes exámenes o pruebas. Por otro lado, el hecho de usar relaciones apriorísticas dificulta la visualización de las etapas intermedias del algoritmo de generación del mapa conceptual, lo cual sí resulta posible en nuestro trabajo, al utilizar redes bayesianas como estructura intermedia.

Si valoramos el sistema completo, hay que destacar que el nuestro permite recomendar no sólo conceptos para reforzar el aprendizaje, como plantea este artículo, sino también ejercicios prácticos en forma de preguntas. Además, nuestro sistema resulta más flexible, ya que permite trabajar con diferentes tipos de conceptos (e.g. competencias de aprendizaje) y decidir cuáles se utilizan para recomendar qué cosas (i.e. preguntas o conceptos). Además, permite a los alumnos visualizar gráficamente el conocimiento mediante colores, lo cual puede ayudarles a decidir en qué áreas de la materia han de concentrarse; mientras que el del artículo sólo permite visualizar el mapa de conceptos a los profesores y con una visión general del conocimiento de todos los alumnos.

Sistema Tutor Inteligente basado en redes bayesianas y resultados de tests

En [KN11, capítulo 11.3], se describe un sistema tutor inteligente orientado a niños cuyo dominio de conocimiento es el sistema decimal. El núcleo es una red bayesiana que se utiliza para modelar el dominio de conocimiento y se utiliza para: (a) diagnosticar qué tipos de errores de comprensión son más probables que tenga el alumno dadas sus respuestas, (b) actualizar los parámetros de la red con la nueva información, y (c) determinar qué tipo de preguntas es más probable que conteste incorrectamente un alumno para proponerle nuevos juegos (preguntas) o ítems teóricos.

En cuanto a la red bayesiana, los investigadores han realizado en paralelo un proceso de construcción manual y automático de la red, lo que les ha permitido comparar los resultados de cada uno y aprovecharlos en el otro. Gracias a que el dominio de conocimiento está muy estudiado, existen ya tests específicos y clasificaciones de los tipos de errores de comprensión muy elaboradas y precisas, por lo que los nodos de la red están basados directamente en éstas.

Por otra parte, cuentan con un gran conjunto de datos de resultados de estudiantes que han realizado un test muy conocido en este dominio, llamado test DCT (Decimal Test Comparison); estos datos se utilizan tanto para el aprendizaje paramétrico como para el estructural, siendo este último realizado mediante un algoritmo llamado CaMML (Causal discovery via MML), específicamente diseñado para descubrir relaciones causales. Aunque los métodos automáticos generan redes que dan buenos resultados, los expertos han preferido utilizar sus redes bayesianas, ya que las relaciones aprendidas automáticamente no son fáciles de comprender y resultan demasiado distintas a las creadas por los expertos de forma manual.

El sistema fue evaluado con 25 estudiantes, de los que 17 mejoraron en alguna medida sus ideas equivocadas del dominio, mientras que el resto se mantuvo igual. Resulta interesante que el trabajo se preocupa también de la usabilidad de la aplicación y de cómo pueden mejorar el aprendizaje de los estudiantes; por ejemplo, se dan pistas en determinadas situaciones.

Comentarios La potencia del sistema descrito radica en que un grupo de expertos ha realizado un proceso de ingeniería de conocimiento muy metódico para dar con la red bayesiana que mejor funcionara en un dominio tan específico como es el sistema decimal. Por el contrario, nuestra propuesta resulta más general, ya que, aunque utilizamos una asignatura a lo largo del trabajo, todos los pasos podrían aplicarse directamente a cualquier otra asignatura, como se describe en la sección. El hecho de ser más general puede hacerla más débil frente a soluciones más específicas.

Por otro lado, en lugar de utilizar redes bayesianas como núcleo de nuestro sistema recomendador, pudiendo haber buscado la forma de aprovechar su capacidad de diagnóstico; hemos preferido utilizar una herramienta más flexible, como son los mapas conceptuales, ya que hemos planteado un método para generarlos de forma semi-automática y, además, porque nos permiten poder utilizar nuestro sistema recomendador en ausencia de suficientes datos o de redes bayesianas adecuadas, ya que el profesor siempre podrá introducir manualmente los mapas conceptuales que se usarán en el sistema.

2.4. Aportaciones al estado de la cuestión

Las principales aportaciones de este trabajo al estado de la cuestión se pueden resumir en los siguientes puntos:

- El método propuesto para la generación de mapas conceptuales de forma semi-automática, utilizando como herramienta intermedia redes bayesianas aprendidas de forma automática, resulta novedoso en la literatura.
- Se ha tenido en cuenta el problema de los valores ausentes al mezclar resultados de diferentes exámenes en un único conjunto de datos, planteándose dos alternativas: (1) un novedoso algoritmo de fusión de estructuras de redes bayesianas y, (2) el tratamiento de los valores ausentes mediante técnicas más tradicionales. En los trabajos encontrados en la literatura que utilizan resultados de test para aprender redes bayesianas, no se plantea este problema y se limitan a aprender redes para cada conjunto de datos por separado.
- Aunque, como hemos visto en la sección 2.2.4, existían estudios que aprendían estructuras de redes bayesianas a partir de resultados de exámenes, no habían experimentado con conjuntos de datos como los utilizados en este trabajo (casi 1500 resultados de tests), ni con algoritmos de aprendizaje más recientes o con un mayor número de nodos.
- El sistema recomendador propuesto tiene la característica novedosa de utilizar mapas conceptuales para actualizar el perfil de los alumnos, suponiendo relaciones propedéuticas en dichos mapas.

3. Estudio y clasificación de exámenes tipo test

La asignatura de “Teoría de Autómatas I” se imparte en la UNED y se enmarca dentro de la carrera de Ingeniería en Informática. El conjunto de datos formado por los resultados de los exámenes realizados por los alumnos durante los años 2004 a 2006 en dicha asignatura, constituye la base de este trabajo, por lo que se ha realizado un estudio previo de los mismos, con el objeto de conocer mejor dichos datos.

En las siguientes secciones, se describe la nomenclatura utilizada para identificar los exámenes, se revisa el formato de los datos en crudo, se comentan los pasos necesarios para preparar los datos para su posterior procesado, se muestran los datos estadísticos obtenidos junto con algunas observaciones a los mismos, se explica el criterio con que se han seleccionado los exámenes utilizados en este trabajo y, finalmente, se realizan diversas clasificaciones de las preguntas de dicho exámenes.

3.1. Nomenclatura de los exámenes

En cada curso académico existen dos convocatorias en las que los alumnos pueden examinarse: febrero y septiembre. En febrero hay dos exámenes: primera semana y segunda semana. En septiembre ocurre de forma similar, aunque se le da otro nombre a los exámenes: el original y el de reserva.

Cada examen tiene tres variaciones (i.e. las mismas preguntas, pero desordenadas). Por tanto, nos encontramos con seis exámenes diferentes en cada convocatoria, a los que se les denomina con una letra:

- A, el examen de la primera semana (en septiembre, el original).
- B y C, variaciones del examen A.
- D, el examen de la segunda semana (en septiembre, el reserva).
- E y F, variaciones del examen D.

En el resto del trabajo se hará referencia a cada examen por su año, convocatoria (febrero o septiembre) y semana (1 o 2), de forma que la semana 1 corresponderá con las variaciones A, B y C, y la semana 2 con las D, E y F.

3.2. Preparación de los datos de respuestas

Para poder realizar el análisis estadístico de los datos, y como paso necesario dentro del trabajo, se ha partido de los datos en crudo para obtener unos “datos limpios”, que puedan ser cargados directamente por diferentes aplicaciones informáticas.

3.2.1. Datos en crudo

Los datos en crudo (u originales), que contienen las respuestas de los alumnos, se encuentran en bases de datos relacionales de tipo Microsoft Access Database (extensión .mdb). Existe una base de datos por curso académico que contiene las siguientes tablas:

- **PRUEBAS_PROGRAMA:** Contiene un listado de las pruebas realizadas, correspondiendo cada una con una convocatoria, y si la prueba es o no de tipo test.
- **PLANTILLATEST:** Contiene la plantilla con las respuestas correctas de cada variación de cada prueba. Además, incluye el año de la prueba y el tipo de algoritmo usado para evaluar el test.
- **RESPUESTAS_TEST_ALUMNOS:** Contiene una fila por cada examen realizado por un alumno. Se indican las opciones del test elegidas, así como el número de respuestas en blanco, de errores y aciertos, y la nota final.

Las relaciones entre las tablas se establecen mediante el identificador de prueba (**ID_PRUEBA**), el tipo de examen (**TIPO_EXAMEN**) y la semana (**SEMANA**). Utilizando estas relaciones se podrían obtener los resultados de un examen concreto mediante sentencias SQL (lenguaje de consulta estructurado).

3.2.2. Datos originales

El contenido de las bases de datos se ha extraído mediante las utilidades MDBTools ². En primer lugar, se han exportado las tablas de las bases de datos a ficheros en formato CSV, siglas en inglés de *valores separados por comas*. Después, estos ficheros se han procesado para obtener un fichero por variación de examen, es decir, por cada año, convocatoria y variación. En total, doce ficheros por año, cada uno con las correspondientes respuestas.

Posteriormente, se ha establecido manualmente la correspondencia entre las preguntas de los exámenes de las variaciones B C y E F, con A y D, respectivamente.

El resultado son diversos ficheros por examen, a partir de los cuales es posible construir una matriz de datos para cada examen distinto, donde las filas corresponden a cada alumno que ha realizado el examen, y las columnas a cada una de las preguntas del mismo, independientemente de la variación concreta (A, B, C, D, E o F) a la que haya respondido.

El resultado son los datos de las respuestas realizadas por los alumnos para cada uno de los cuatro exámenes de un curso académico (F1, F2, S1 y S2), para cada uno de los años estudiados.

3.3. Análisis estadístico de las respuestas

A partir de los datos limpios, se han obtenido algunos resultados estadísticos básicos de los exámenes, con el objetivo de comprender mejor las características de los datos. Para ello, se ha utilizado la herramienta GNU/Octave [Eat08], capaz de leer directamente los valores de ficheros CSV permitiéndonos realizar operaciones numéricas sobre matrices y vectores de forma sencilla.

3.3.1. Porcentajes y valores medios

Se han obtenido resultados a nivel de examen, para hacernos una idea agregada de los datos de cada uno de ellos (ver Tabla 1) y, además, se han obtenido resultados a nivel de

²MDB Tools: Información en <http://mdbtools.sourceforge.net/>

pregunta, para conocer qué preguntas resultaban más difíciles, qué porcentaje de alumnos dejaba las preguntas en blanco, etc. Los datos de cada pregunta se muestran en el anexo A.

Año	Conv.	Semana	Nº alum.	% Nota ≥ 5	Nota media	% blanco	Nº alum. año
2004	F	1	120	65	5.4847	17	382
2004	F	2	190	73	6.1867	16	
2004	S	1	72	35	3.9104	22	
2004	S	2	0	0	0.0000	0	
2005	F	1	117	50	4.6180	21	514
2005	F	2	237	32	3.9141	22	
2005	S	1	155	25	3.2441	22	
2005	S	2	5	40	4.2040	26	
2006	F	1	112	71	5.7392	20	524
2006	F	2	290	54	5.0067	19	
2006	S	1	122	35	3.7084	24	
2006	S	2	0	0	0.0000	0	

Tabla 1: Datos estadísticos básicos de los exámenes de 2004 a 2006. Se muestran en color rojo aquellos que han sido descartados por haber sido realizados por pocos o ningún alumno.

3.3.2. Correlaciones entre las preguntas

A nivel de cada examen, si suponemos que las respuestas de los alumnos son repeticiones de un experimento aleatorio, podemos obtener sus probabilidades asociadas a partir de las frecuencias observadas.

Posteriormente calculamos el coeficiente de correlación de Pearson [Wik13], suponiendo que las variables aleatorias pueden tomar los valores *acierto* o *fallo*, para obtener las *dependencias lineales* entre dichas preguntas, es decir, cuándo el acierto o fallo de una pregunta suele estar asociado al acierto o fallo de otra.

En la figura 5 se muestra un grafo de preguntas que muestra las correlaciones en el examen 2004F1, estando los enlaces etiquetados con el valor de la correlación. En general, para todos los exámenes, las preguntas que aparecen correlacionadas se enmarcan en el mismo tema de la asignatura. Sin embargo, muchas preguntas que se enmarcan en el mismo tema, no aparecen correlacionadas, aunque muchos de los conceptos son compartidos. Hay que destacar que en varios exámenes apenas aparecen correlaciones.

Los resultados completos se encuentran en el anexo B, incluyendo descripciones de las funciones de Octave utilizadas, así como las partes más importantes de los programas escritos.

3.4. Clasificación manual de preguntas

Una vez estudiados, seleccionados y limpiados, los datos de los resultados de los exámenes, se han clasificado las preguntas de los mismos con el objetivo de asignarles etiquetas (concretamente, conceptos o categorías) que puedan utilizarse en el posterior aprendizaje de redes bayesianas y construcción de mapas conceptuales.

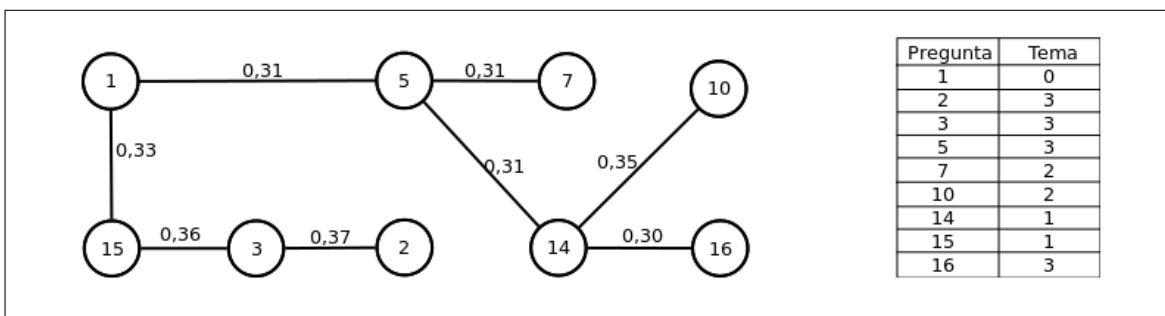


Figura 5: Grafo de correlaciones entre las preguntas del examen 2004F1. Se muestran las correlaciones con valor mayor o igual que 0,30. Las preguntas con un mayor valor de correlación son la 2 y la 3, ambas relacionadas con el mismo tema de la asignatura: las máquinas de turing.

3.4.1. Conceptos de la materia

A partir de los enunciados de los exámenes y de sus soluciones, se han asignado uno o más conceptos a las preguntas de los exámenes bajo estudio. El criterio utilizado ha consistido en asociar aquellos que resultaría necesario comprender para responder correctamente a la pregunta.

Para llevar a cabo esta tarea, se han estudiado diferentes fuentes sobre teoría de autómatas [Bro93, Hop01, Par08] y teoría de conjuntos [Lar02]. A la vez que se ha estudiado la materia, se ha aprovechado para crear mapas conceptuales de forma manual (descritos en la sección 4.3.1), con el objetivo de familiarizarse con ellos. Además, se ha preferido asignar conceptos que aparecieran en los mapas conceptuales creados manualmente para poder compararlos entre sí.

En la Tabla 2 se muestra un listado de todos los conceptos utilizados, incluyendo el número de veces que han sido asignados a alguna pregunta.

3.4.2. Categorías

Se han realizado diversas categorizaciones de las preguntas de los exámenes con el objetivo de utilizarlas posteriormente para los objetivos de este proyecto. Las categorías planteadas han sido las siguientes:

1. *Competencias generales y específicas.* Corresponden a competencias dentro de la asignatura de Teoría de Autómatas I. Para ello, nos hemos basado en las competencias que se espera que obtenga el alumno tras cursar esta asignatura. Un ejemplo de competencia general sería “funcionamiento de los autómatas finitos y de pila”, mientras que una competencia específica de podría ser “determinar equivalencia entre autómatas finitos”.
2. *Temas.* Corresponde a asignar como categoría aquellos temas en que se enmarca la pregunta (e.g. expresiones regulares). Este puede ser uno o más de los cuatro primeros capítulos del libro [Bro93], que corresponden al temario impartido en la asignatura bajo estudio. Ejemplo: “Capítulo 1: Autómatas finitos y lenguajes regulares”. Por lo tanto, una pregunta puede pertenecer a uno o más de: 0, 1, 2 y 3.

cadenas (59)	foma normal de chomsky (10)
autómatas finitos (35)	gramáticas regulares (10)
lenguajes regulares (33)	terminación anormal (10)
máquinas de turing (30)	estrella de kleene (9)
equivalentes (26)	transiciones (8)
lenguajes independientes del contexto (25)	gramáticas estructuradas por frases (6)
deterministas (21)	reconocer (6)
expresiones regulares (21)	representación (6)
lenguajes formales (19)	concatenación (5)
no deterministas (19)	operaciones (5)
autómatas de pila (17)	conjuntos (4)
definición por conjunto (17)	inversa (4)
lenguajes estructurados por frases (16)	lenguajes generales (4)
poder de reconocimiento (16)	mecanismo de control (4)
gramáticas independientes del contexto (15)	máquina de turing compleja (4)
lema de bombeo (15)	tesis de turing (4)
cardinalidad (14)	intersección (3)
complemento (14)	aceptable (2)
jerarquía de lenguajes (14)	bloques básicos sobre lenguajes (2)
derivación (11)	bucle (2)
propiedades de cierre (11)	construcción de algoritmos (2)
unión (11)	gramáticas formales (2)
decidible (10)	pila (2)
estados (10)	

Tabla 2: Listado de los conceptos asignados a las preguntas, ordenados de mayor a menor número de asignaciones. Entre paréntesis, el n^o de preguntas que tienen el concepto asignado. Recordemos que se han estudiado 144 preguntas (9 exámenes con 16 preguntas cada uno).

3. *Tipo*. Corresponde al tipo de pregunta en función de si son exclusivamente memorísticas, de establecer relaciones entre conceptos con un cierto componente práctico o si se trata de resolver ejercicios puramente prácticos. Esta categorización resulta un tanto subjetiva, ya que todas las preguntas necesitan de un conocimiento teórico para poder ser comprendidas y, posteriormente, resueltas. Sin embargo, sí es cierto que los exámenes suelen contener preguntas eminentemente teóricas, en cuya resolución sólo influye el grado de conocimiento de la materia, pero no la práctica o la experiencia.
4. *Dificultad*. Esta categorización se podría obtener a partir de los resultados de los exámenes, estableciendo una serie de intervalos en el porcentaje de alumnos que respondieron correctamente a cada pregunta y aplicando algún tipo de normalización teniendo en cuenta el conocimiento general de los alumnos que han respondido a cada pregunta.

En términos de los objetivos de este trabajo, es decir, de la construcción semi-automática de mapas conceptuales y del diseño de sistemas recomendadores, estas categorizaciones han de ayudarnos a establecer relaciones entre las preguntas de los distintos exámenes y sus conceptos asociados. Vamos a analizarlas desde este punto de vista:

- La primera y la segunda (i.e. *competencias* y *temas*) establecen relaciones entre preguntas del mismo contexto teórico de la asignatura, por lo que sus conceptos asociados serán cercanos y podrían aparecer en un mismo mapa conceptual. Por lo tanto, tendrá sentido buscar relaciones entre dichos conceptos.
- La tercera y la cuarta (i.e. *tipo* y *dificultad*), sin embargo, nos dan información sobre características no relacionadas con la asignatura. De hecho, estas mismas categorías podrían aplicarse a preguntas de examen de otras asignaturas, por lo que no tendrá sentido plantearnos el aprendizaje de redes bayesianas a partir de estas, ya que son demasiado generales. En el caso de la tercera, podríamos obtener relaciones entre los cuatro temas de la asignatura, mientras que en el caso de la cuarta, si discretizáramos los valores, podríamos obtener relaciones entre preguntas muy difíciles y preguntas fáciles, por ejemplo. Sin embargo, estas categorías sí podrían tener utilidad en otras tareas; por ejemplo, en un sistema recomendador que generase exámenes adaptados al nivel de aprendizaje del alumno o para realizar el aprendizaje filtrando sólo aquellas preguntas que correspondan a un tema determinado.

A la luz del análisis anterior, se han clasificado las preguntas de los exámenes asignándoles una competencia específica, una o más competencias generales, uno o más temas de la asignatura y el tipo de la pregunta. Las competencias generales y específicas utilizadas se pueden encontrar en los grafos de competencias (Figuras 6 y 7) que se describen en la siguiente sección.

3.5. Grafos de dependencias

A raíz de la categorización manual realizada utilizando competencias generales y específicas, se ha planteado la construcción por un experto, de un grafo de dependencias para cada uno de los tipos de competencias comentados. Los enlaces de estos grafos pueden verse como dependencias entre competencias, es decir, caminos de aprendizaje que el alumno debería seguir para aprender las diferentes competencias de la asignatura. En la figura 6, se muestra el grafo de dependencias de competencias generales y, en la figura 7, el de competencias

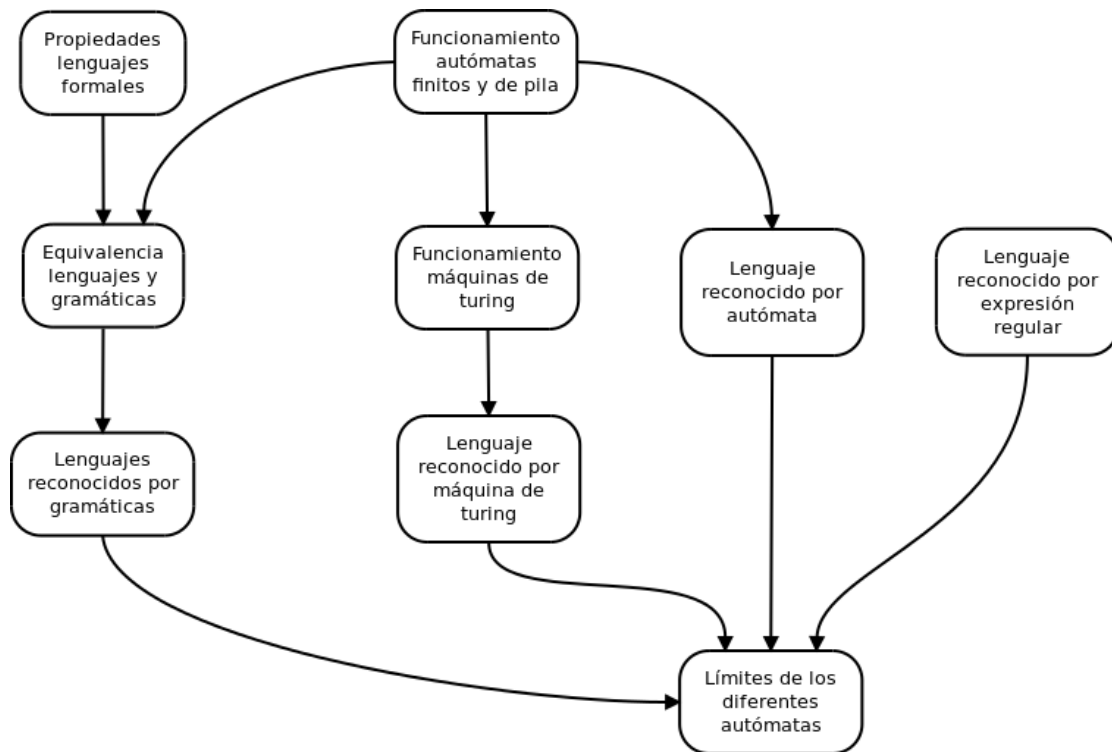


Figura 6: Grafo de dependencias entre competencias generales de la asignatura de Teoría de Autómatas I, construido de forma manual.

específicas. Durante la siguiente etapa del trabajo, estos grafos servirán de referencia de lo que un experto consideraría relaciones de dependencia apropiadas entre las competencias de la asignatura. También será posible completar estos grafos con nuevos enlaces obtenidos al generar mapas conceptuales de forma semi-automática.

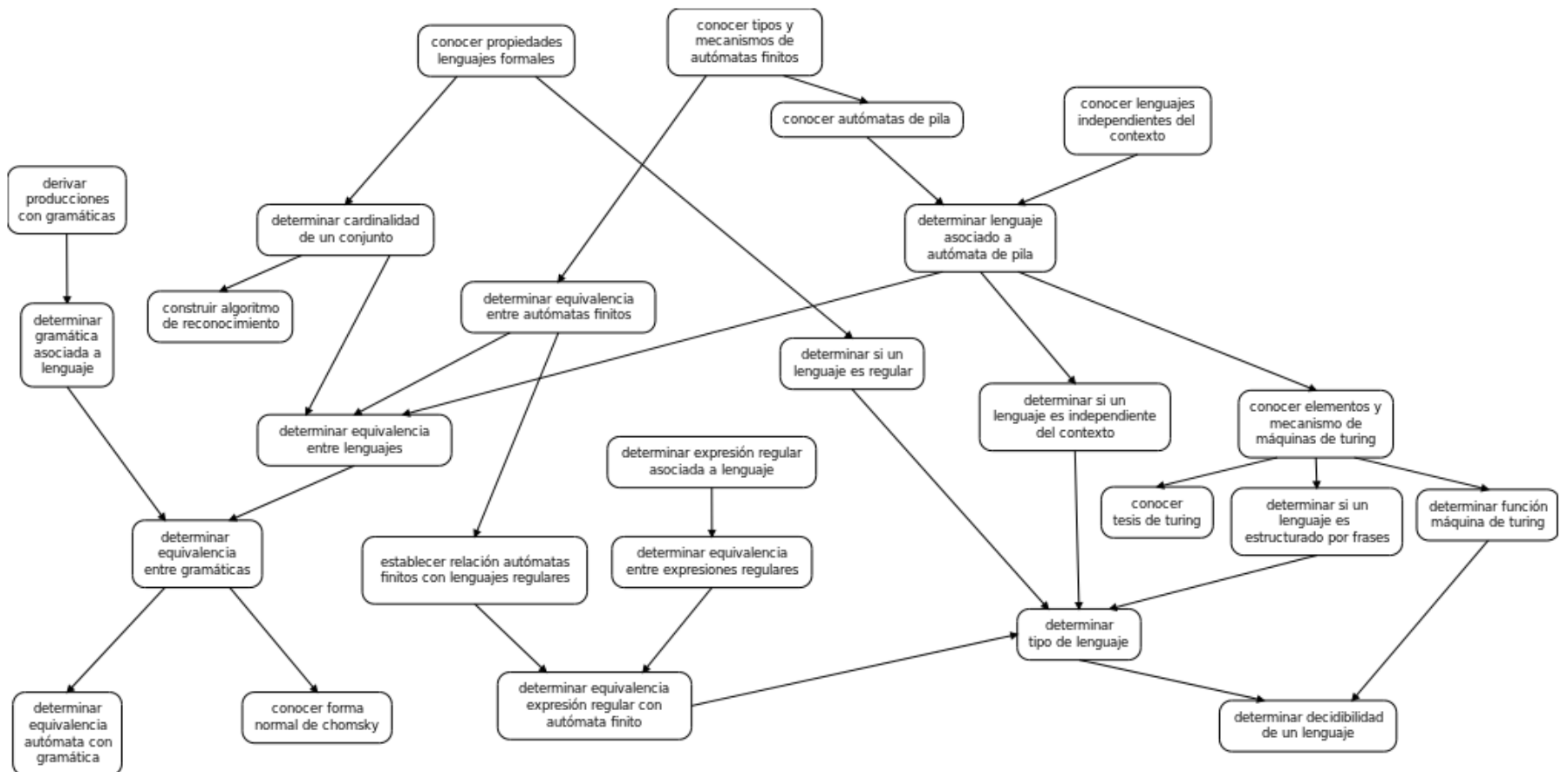


Figura 7: Grafo de dependencias entre competencias específicas de la asignatura de Teoría de Autómatas I, construido de forma manual.

4. Generación semi-automática de mapas conceptuales

En esta sección se describen los experimentos realizados para generar de forma semi-automática mapas conceptuales para la asignatura de Teoría de Autómatas I utilizando técnicas de aprendizaje estructural de redes bayesianas, así como de minería de datos. En primer lugar, se describen los datos de entrada para el aprendizaje, revisando diferentes alternativas y formas de preprocesado. Después, se describen los experimentos de aprendizaje estructural de redes bayesianas utilizando diversas configuraciones y conjuntos de datos. Se evalúan sus resultados mediante diferentes métricas. Por último, se describe el algoritmo utilizado para generar mapas conceptuales a partir de las redes bayesianas del paso anterior, así como los resultados obtenidos.

4.1. Datos de entrada

En esta sección, se estudian diferentes alternativas para los datos de entrada del aprendizaje automático. Recordemos que el objetivo es obtener relaciones que puedan traducirse en enlaces en un mapa conceptual. Para ello, se utilizarán las clasificaciones de las preguntas de los exámenes bajo estudio, realizadas de forma manual tal y como se describe en la sección 3.4.

4.1.1. Descripción de los datos originales

Los datos originales fueron obtenidos para cada examen (ver sección 3.2.2), pero para facilitar su descripción, supondremos que están agrupados en un único conjunto de datos que contiene todos los datos disponibles. Este conjunto estaría formado por instancias cuyos atributos corresponden a las preguntas de los exámenes y los valores a las respuestas que ha dado un alumno a dichas preguntas, es decir, tenemos una fila por alumno que ha realizado algún examen y dieciséis columnas por examen. El resultado es una matriz de la siguiente forma:

$$M = \begin{bmatrix} r_1^1 & r_2^1 & \dots & r_{16}^1 \\ r_1^2 & r_2^2 & \dots & r_{16}^2 \\ & & \dots & \\ r_1^{n_1} & r_2^{n_1} & \dots & r_{16}^{n_1} \\ & & \dots & \\ & & & \dots & r_1^{n_1+\dots+n_{j-1}+1} & r_1^{n_1+\dots+n_{j-1}+1} & \dots & r_1^{n_1+\dots+n_{j-1}+1} \\ & & & & r_1^{n_1+\dots+n_{j-1}+2} & r_2^{n_1+\dots+n_{j-1}+2} & \dots & r_{16}^{n_1+\dots+n_{j-1}+2} \\ & & & & & & \dots & \\ & & & & r_1^{n_1+\dots+n_{j-1}+n_j} & r_2^{n_1+\dots+n_{j-1}+n_j} & \dots & r_{16}^{n_1+\dots+n_{j-1}+n_j} \end{bmatrix}$$

siendo $n_1 + \dots + n_j$ el n° total de alumnos que realizaron los exámenes desde 2004 hasta 2006, ordenados cronológicamente (i.e. 2004F1, 2004F2, ..., 2006S1). M es una matriz de 1415 filas por 144 columnas, en total 203.760 elementos, de los que conocemos 22.640 (el 9%), siendo el resto valores ausentes, ya que cada instancia (fila) sólo nos da información de las 16 preguntas a las que respondió cada alumno en el examen correspondiente. El caso de que un alumno haya realizado varios exámenes no es relevante para este trabajo, ya que en cualquier

caso los realizará con diferentes competencias, por lo que seguirán siendo de interés tratadas por separado.

En cuanto a los valores que toman los elementos r de la matriz M , son los siguientes: el n° de la respuesta que eligió el alumno (1, 2 o 3) o si la respuesta quedó en blanco (0). Este convenio se ha establecido en el contexto de este trabajo y se implementa en la etapa de preparación de los datos (sección 3.2.2). Por tanto, el valor 1 corresponde a la respuesta A, el valor 2 a la B, el valor 3 a la C y el 0 a la ausencia de respuesta.

Además de la matriz M , disponemos de un vector s con las respuestas correctas, que toma la forma:

$$s = \left[s_1^1 \ s_2^1 \ \dots \ s_{16}^1 \ s_1^2 \ s_2^2 \ \dots \ s_{16}^2 \ \dots \ s_1^9 \ s_2^9 \ \dots \ s_{16}^9 \right]$$

siendo s^i las respuestas correctas a las 16 preguntas del examen i . Los valores que pueden tomar los elementos de s son únicamente el número de la respuesta correcta (1, 2, o 3), ya que en este caso no tiene sentido el valor 0.

4.1.2. Atributos

Los datos de entrada que utilizaremos en el aprendizaje automático están formados por instancias o elementos, los cuales tienen unos atributos que los describen. En el caso de los datos de la matriz M definida en la sección 4.1, estos corresponden con las preguntas de los exámenes (i.e. cada columna corresponde a una pregunta distinta).

Cuando se aprende la estructura de una red bayesiana, los atributos de los datos de entrada corresponden con los nodos de dicha red, por lo que dichos atributos definirán entre qué tipo de elementos se establecerán relaciones. En el caso de los datos originales, se obtendrían redes bayesianas cuyos nodos corresponderían a preguntas, lo cual no resulta de interés para el objetivo de nuestro trabajo. Por esta razón, lo que vamos a plantear son distintas alternativas para que las redes aprendidas sean de mayor interés y aprovechen los datos de la mejor forma posible. A continuación, se describen las opciones que utilizaremos.

Competencias generales y específicas Cada atributo/nodo corresponderá a competencias (generales o específicas) con las que han sido clasificadas manualmente las preguntas de los diferentes exámenes.

Este tipo de atributos permite relacionar los datos de todos los exámenes, ya que las clasificaciones son globales para todos los exámenes, como vimos en la sección 3.4.2. El hecho de poder utilizar todos los datos como entrada para el aprendizaje tiene la parte positiva de que nos permite utilizar mayor información, y la parte negativa de que da lugar a valores ausentes (ya que no siempre todas las categorías aparecen en alguna pregunta de cada examen). Otra opción es utilizar los datos de cada examen de forma independiente, evitando así los valores ausentes, aunque limitando las competencias que podrán aparecer en las redes a las que aparezcan en cada examen y dando lugar a varias redes (una por examen) que luego será necesario mezclar en una sola.

Conceptos Cada atributo/nodo corresponderá a un concepto de los asignados manualmente a cada una de las preguntas de los diferentes exámenes bajo estudio, tal y como se describe en

la sección 3.4.1. Estos atributos también permiten relacionar datos de distintos exámenes y tienen el problema de los valores ausentes. De igual forma que en el caso de las competencias, se pueden utilizar los datos de exámenes de forma independiente.

Estos atributos plantean el problema de que algunos de ellos tienen diferentes significados en función del contexto (tema de la asignatura) en que se encuentren. Además, si se utilizaran todos juntos generarían un grafo con una gran cantidad de nodos/conceptos (todos aquellos que han sido utilizados para clasificar alguna pregunta), que con toda seguridad resultaría complejo de evaluar (por un humano o mediante alguna métrica) y poco fiable, debido al problema descrito anteriormente.

Por esta razón, los datos de conceptos se separarán por tema, lo que dará lugar a redes bayesianas con nodos pertenecientes a un mismo contexto de la asignatura, como veremos en la sección 4.2.5.

4.1.3. Preprocesado de los datos originales

Los valores de la matriz M tal y como se obtienen de las bases de datos nos dan información de la respuesta concreta que han elegido los alumnos al realizar los exámenes.

Transformación de los valores La primera etapa del preprocesado consiste en transformar los valores originales en valores discretos que indiquen si el resultado ha sido *acierto* o *fallo*. En cuanto a las *respuestas en blanco*, vamos a entender que son *fallos*, ya que si el alumno decidió no contestar es porque su conocimiento no era suficiente como para arriesgarse a dar una respuesta. El resultado de esta etapa es una matriz similar a la original pero con los valores pertenecientes al conjunto $\{0, 1\}$, siendo 0 el fallo y 1 el acierto.

Transformación de los atributos En segundo lugar, se transforman los atributos originales en los descritos en la sección 4.1.2. Como ya hemos comentado, el uso de competencias y conceptos dará lugar a una matriz para cada clasificación, las cuales tendrán la siguiente forma:

$$M_C = \begin{bmatrix} v_1^1 & v_2^1 & \dots & v_i^1 \\ v_1^2 & v_2^2 & \dots & v_i^2 \\ v_1^3 & v_2^3 & \dots & v_i^3 \\ \dots & \dots & \dots & \dots \\ v_1^{n_1+\dots+n_j} & v_2^{n_1+\dots+n_j} & \dots & v_i^{n_1+\dots+n_j} \end{bmatrix}$$

siendo $n_1 + \dots + n_j$ el n° total de alumnos que realizaron los exámenes desde 2004 hasta 2006, ordenados cronológicamente (i.e. 2004-F-1, 2004-F-2, ..., 2006-S-1), de la misma forma que en la matriz M ; y siendo i el n° de clases en la categoría C , que podrá ser competencias generales, específicas o conceptos.

Los valores de los elementos de las diferentes matrices M_C se obtienen a partir de la clasificación manual de cada pregunta para cada una de las clasificaciones utilizadas y de los valores 0 o 1 que tiene cada una de las preguntas respondidas por cada alumno.

Los valores pueden tener distintas formas:

- *Discretos binarios*, es decir, *acierto* o *fallo*, igual que los datos originales, sólo que en este caso representan si se ha acertado una determinada clasificación (i.e. un concepto o una competencia). Por ejemplo, dado el concepto “máquina de turing”, el valor indicaría si se han respondido correctamente las *suficientes* preguntas como para considerar que el alumno conoce dicho concepto (y por tanto lo ha acertado). El porcentaje de preguntas de una determinada categoría necesarios para considerarla acertada vendrá dado por un umbral (e.g. 75 %). Se estudiarán diferentes umbrales para determinar cuál resulta más adecuado para nuestros datos.
- *Discretos multinomiales*, por ejemplo, *todas*, *alguna* o *ninguna*. En este caso, estamos ante valores nominales con tres clases. Nos permitiría obtener información del tipo: “Si no se responde *alguna* pregunta de la categoría A, no se responde *ninguna* de la B”.
- *Continuos*, por ejemplo, porcentual en el rango $[0, 1]$. En este caso, serían valores numéricos que indicarían el porcentaje de las preguntas de cada categoría que han sido respondidos correctamente. Este tipo de valores tiene los mismos problemas que el anterior.

En este trabajo, se realizarán experimentos con datos discretos binarios y datos continuos; y, dentro de los primeros, se evaluarán diferentes umbrales de preguntas acertadas. Los resultados permitirán determinar qué tipo de atributos resulta más adecuado para nuestros objetivos. Los datos discretos multinomiales no se estudian en este trabajo y se dejan como experimento futuro, a realizar en función de los resultados obtenidos.

4.1.4. Filtrado de posibles fuentes de ruido

Si observamos el análisis estadístico básico de los resultados de los exámenes (sección 3.3), se puede observar que 3 preguntas (de las 144 estudiadas) han sido respondidas correctamente por un número muy alto de alumnos (entre el 95 % y el 100 % de aciertos). Debido a que estas preguntas podrían aparecer correlacionadas casi con cualquier otra, se ha planteado su filtrado antes de realizar el aprendizaje de las redes bayesianas.

Por otro lado, otras 5 preguntas se han clasificado con la competencia específica “pregunta-variada”, la cual no es realmente una competencia, sino que indica que la pregunta incluye competencias variadas. Normalmente, son preguntas cuyas posibles respuestas no están relacionadas y basta con conocer la adecuada para responder correctamente (o conocer todas menos una y acertar por descarte). A causa de esto, suelen tener asociados conceptos y competencias generales muy dispares, lo que dificulta enormemente hipotetizar que el alumno conoce o desconoce todos esos conceptos y competencias.

Se ha planteado estudiar las diferencias en las redes aprendidas filtrando los dos tipos de preguntas anteriores, con el fin de determinar si es apropiado, o no, tenerlas en cuenta en el aprendizaje.

4.2. Aprendizaje estructural de redes bayesianas

En primer lugar, se referencian los algoritmos utilizados en el aprendizaje estructural, así como la implementación utilizada. Además, se describe un algoritmo diseñado en este trabajo que permite fusionar varias redes bayesianas en una sola. Veremos que por las características de los datos de que disponemos, será posible, o bien trabajar con un único conjunto de datos

imputando valores ausentes, o, por el contrario, utilizar muchos conjuntos de datos fusionando las estructuras de las redes aprendidas.

Posteriormente, se describen las métricas utilizadas para evaluar las redes bayesianas aprendidas. Estas métricas nos permitirán tener un criterio a la hora de determinar qué algoritmos, parámetros, operaciones de preprocesado, etc. son más adecuadas para el objetivo de este trabajo.

Por último, se describen los experimentos realizados y los resultados obtenidos para cada uno de los conjuntos de datos derivados de las clasificaciones realizadas de forma manual (descritas en la sección 4.1.2). Los experimentos se realizarán de forma secuencial, aprovechando en cada uno las conclusiones obtenidas de los anteriores.

4.2.1. Algoritmos

Para realizar el aprendizaje estructural de redes bayesianas, se utilizarán los algoritmos Hill Climbing (HC) y Max-Min Hill Climbing (MMHC), con la métrica BIC (Bayesian Information Criterion), descritos en la sección 2.2.2, utilizando el paquete de software “bnlearn” [Scu09] para R. En ambos algoritmos, se utilizan los valores por defecto de los parámetros, es decir, en HC no se realizan reinicios aleatorios de la búsqueda, y en MMHC se utiliza un umbral alfa de 0.05 para los tests de independencia condicional. Además, en ninguno de los casos se restringe la estructura de la red con información adicional a los datos.

En cuanto al tratamiento de los valores ausentes, de las diferentes técnicas descritas en la sección 2.2.3, se utilizarán las siguientes: eliminación de instancias con valores ausentes, imputación unidimensional con la mediana e imputación multidimensional kNN (k-Nearest Neighbors) con $k = 5$. Se utilizan los paquetes de R “Hmisc” [TAKP13] y “VIM” [Har13] para los dos tipos de imputación, respectivamente.

En el anexo C se puede obtener más información sobre las alternativas de software evaluadas, así como descripciones del código utilizado tanto para el aprendizaje estructural como para el tratamiento de valores ausentes.

Algoritmo de fusión de estructuras de redes bayesianas

En algunos experimentos, dado el alto porcentaje de valores ausentes de los datos agregados de todos los exámenes, no es recomendable (ni en algunos casos viable) el tratamiento de los mismos. Aún siendo factible la utilización de técnicas de imputación más avanzadas que las propuestas en este trabajo (e.g. imputación múltiple), se recomienda revisar antes el conjunto de datos y tratar de reducir el nº de valores ausentes [JY11, recomendación 9]. En nuestro caso, los valores ausentes se han creado de forma artificial (al mezclar varios conjuntos de datos más pequeños: los de cada examen), por lo que parece adecuado buscar opciones alternativas para no tener que tratar con un volumen tan alto de valores ausentes.

La estrategia planteada consiste en el aprendizaje estructural de redes bayesianas para cada examen por separado y la posterior fusión de las estructuras de las redes obtenidas en una sola, siguiendo el algoritmo 1, diseñado teniendo en cuenta la fuerza de los enlaces de las redes bayesianas (medida como la variación en el valor de la métrica utilizada si el enlace fuera eliminado) y el número de datos utilizados para el aprendizaje (no serán igual de relevantes los enlaces de redes aprendidas con 50 datos que con 200). El algoritmo consta de dos fases: una primera donde se añaden los enlaces de todas las redes de entrada evitando

enlaces bidireccionales, y una segunda donde se eliminan los ciclos que hayan resultado del paso anterior, mediante el uso de las componentes fuertemente conectadas.

El algoritmo descrito ha sido implementado en el lenguaje de programación Python [VRF03], utilizando la librería `python-graph`³, que dispone de clases para tratar grafos no dirigidos y calcular sus componentes fuertemente conectados.

Algoritmo 1 Fusión de la estructura de varias redes bayesianas en una única que contiene los enlaces más relevantes manteniendo la propiedad de que el grafo sea un GDA. Los pesos de los enlaces se calculan ponderando su fuerza con el número de datos utilizados en su aprendizaje automático.

```
Función: FusionarEstructuraRBs
Entrada: rbs[n] (array de redes bayesianas)
Salida: rb      (red bayesiana)

# 1º Fusionar todos los enlaces en una nueva red, evitando enlaces bidireccionales
# entre dos nodos. La red resultante será conectada siempre que al menos uno de
# los nodos de cada red de entrada aparezca en al menos otra distinta.

rb = nueva RedBayesiana()
Para cada red en rbs
    Para cada enlace en rbs.enlaces()
        Si (enlace ya tratado) entonces continuar
        # Sumamos el peso del enlace en todos los exámenes en que aparece
        peso_normal = suma(rbs, enlace.peso)
        Si (enlace.invertido() existe en rbs) entonces
            peso_invertido = suma(rbs, enlace.invertido().peso)
            Si (peso_total < peso_total_invertido) entonces
                rb.añadir_enlace(enlace, peso_normal + peso_invertido)
            Si no
                rb.añadir_enlace(enlace_invertido, peso_normal + peso_invertido)
        Si no
            rb.añadir_enlace(enlace, peso_normal)

# 2º Transformar el grafo en un GDA, eliminando los enlaces de menor peso que forman
# parte de un ciclo.

Mientras existan(rb.componentes_fuertemente_conectados(), longitud > 1) hacer
    enlace_a_borrar = Nada
    Para cada cfc en rb.componentes_fuertemente_conectados() hacer
        Si (cfc ya tratado) entonces continuar
        Si (cfc.enlace_minimo_peso() < enlace_a_borrar.peso()) entonces
            enlace_a_borrar = cfc.enlace_minimo_peso()
    # El grafo seguirá siendo conectado, ya que el enlace formaba parte de un ciclo.
    rb.borrar_enlace(enlace_a_borrar)

Devolver rb
```

4.2.2. Métricas de evaluación de redes bayesianas

Se plantea la utilización de tres métricas para evaluar las redes bayesianas aprendidas en los experimentos realizados. Estas métricas prestan atención a la estructura de la red, a su complejidad (visual y probabilística) y a lo distintas que son con respecto a los grafos de dependencias construidos manualmente (sección 3.5).

³`python-graph`: A library for working with graphs in Python, <http://code.google.com/p/python-graph/>

Tamaño medio del Markov Blanket El “markov blanket” de un nodo X , $MB(X)$, es el conjunto formado por sus nodos padre, sus nodos hijo y los otros padres de sus hijos. Tiene la propiedad de que un nodo $Y \notin MB(X)$ es condicionalmente independiente de X dado $MB(X)$, es decir, que el nodo X queda aislado del resto de la red dado su “markov blanket”. Cuanto más pequeño sea, en media, menor será el número de nodos que influyen directamente en otros nodos.

Factor de salto medio Se llama así al número de hijos que tienen, en media, los nodos de una red bayesiana. Cuanto menor sea este número, más sencilla será la red y más fáciles de interpretar serán las relaciones de dependencia entre los nodos.

Distancia Estructural de Hamming (SHD) Mide el número de operaciones de adición, eliminación y reorientación de enlaces, que hay que hacer en un grafo dirigido acíclico para transformarlo en otro. Esta métrica se describe en el artículo del algoritmo MMHC [TBA06] y se utiliza para evaluar los resultados obtenidos. Resulta interesante que esta métrica no trabaja directamente con el grafo de una red bayesiana, sino que lo transforma antes en un grafo que corresponde a su clase de equivalencia de Markov, de forma que mantiene todos los enlaces pero sólo son dirigidos los estrictamente necesarios para representar dicha clase. Se utilizará esta métrica para comparar las redes aprendidas con los grafos de dependencias construidos manualmente.

4.2.3. Experimentos con datos de competencias generales

En los datos de competencias generales los atributos corresponden a competencias generales de la asignatura de Teoría de Autómatas I, tal y como se describieron en la sección 4.1.2. A continuación, se describen los diferentes experimentos realizados sobre estos datos utilizando diversos algoritmos, tipos de datos, técnicas de preprocesado y de imputación. Además, las redes aprendidas se comparan con el grafo de la figura 6, que representa las relaciones propedéuticas observadas por un experto entre las diferentes competencias generales.

Análisis de valores ausentes

Los datos agregados (todos los exámenes) de competencias generales tienen un bajo porcentaje de valores ausentes, siendo la competencia “funcionamiento de autómatas finitos y de pila” la única que sólo aparece en 8 de los 9 exámenes estudiados; el resto aparecen en todos los exámenes. Sin embargo, cuando se realiza el filtrado de posibles fuentes de ruido descrito en la sección 4.1.4, el porcentaje de valores ausentes pasa del 5 % (sin filtrado) al 20 % (con filtrado), un valor considerablemente más alto, pero todavía aceptable. En los siguientes experimentos, se utilizarán diversas técnicas de imputación, así como la eliminación de las instancias con valores ausentes.

Experimento 1: Tipos de datos y filtrado de ruido

Se comparan diferentes tipos de datos (continuos, binarios con umbral 50 %, binarios con umbral 75 % y binarios con umbral 100 %) con y sin filtrado de posibles fuentes de ruido. En todos los casos, se utiliza el algoritmo MMHC y se eliminan las instancias con valores

ausentes, de forma que evitamos introducir más variables en el experimento, con el fin de determinar qué tipos de datos son más adecuados y si el filtrado de ruido es positivo para el aprendizaje de redes bayesianas con datos de competencias generales.

Para todas las métricas (Figura 8), se obtienen valores iguales o más pequeños con filtrado que sin él. Por lo tanto, podemos concluir que, con estos datos, la presencia de preguntas demasiado fáciles o demasiado generales, dan lugar a más dependencias entre nodos en la red aprendida.

En cuanto a los datos continuos, estos dan lugar a redes casi el doble de complejas que si se utilizan datos binarios. Además, sus valores para la métrica SHD son mayores, con lo que las redes son menos parecidas al grafo construido manualmente. Por otro lado, dentro de los posibles datos binarios, los umbrales 50 y 100 dan lugar a redes más sencillas que el umbral 75. A la hora de elegir entre el umbral 50 y el 100, hay que pensar qué nos dice cada tipo de datos: el 50 nos indica que al menos la mitad de las preguntas con una determinada categoría han sido respondidas correctamente, mientras que el 100 nos indica que todas las preguntas con una determinada categoría han sido respondidas correctamente. Esta segunda condición es mucho más fuerte y hace que sea más complicado (y a la vez más fiable) conocer cada categoría. Sin embargo, dado que una misma categoría aparecerá como mucho dos o tres veces en cada examen, parece razonable que sea necesario acertar todas las preguntas de dicha categoría para considerarla correcta (frente a la opción de que sólo sea necesario acertar la mitad).

Experimento 2: Algoritmos de aprendizaje e imputación de valores ausentes

Se comparan dos algoritmos de aprendizaje: Hill Climbing (HC) y Max-Min Hill Climbing (MMHC), utilizando diferentes técnicas de imputación de valores y comparando los resultados con la opción de eliminar las filas con datos ausentes. En todos los casos, se utilizan datos discretos con umbral 100% y el filtrado de posibles fuentes de ruido descrito en la sección 4.1.4, ya que como hemos visto en el primer experimento, los resultados obtenidos resultan más adecuados para el objetivo del trabajo.

En los resultados para las métricas utilizadas (Figura 8), se puede observar que todas las métricas dan resultados igual o más complejos cuando se utiliza algún tipo de imputación que cuando no se utiliza ninguna. Además, el algoritmo MMHC produce redes más sencillas que el algoritmo HC cuando no se utiliza imputación de valores ausentes kNN, en este caso, ambos algoritmos dan lugar a redes de idéntica complejidad.

Resultados

En las figuras 9 y 10 se muestra la red aprendida con una menor distancia SHD al grafo de competencias manual y la red aprendida con imputación kNN (idéntica con HC o MMHC), respectivamente. La primera tiene la característica de ser la menos compleja según todas las métricas utilizadas, mientras que la segunda resulta de interés porque es generada por ambos algoritmos, HC y MMHC, lo cual no ocurre en ningún otro caso.

Si comparamos las redes entre ellas se puede observar que son muy similares, aunque difieren en algunos enlaces, por ejemplo, la competencia “lenguaje generado por expresión regular” tiene sólo un hijo en la primera, mientras que en la segunda tiene tres.

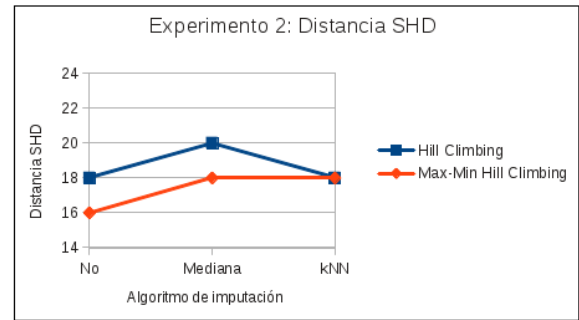
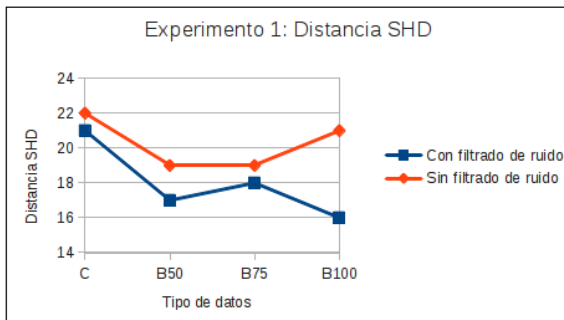
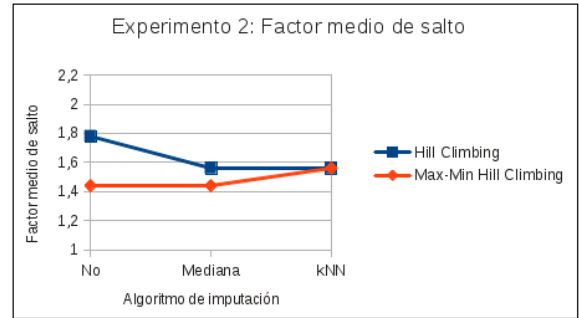
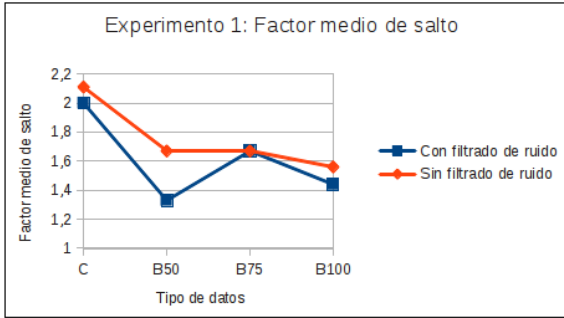
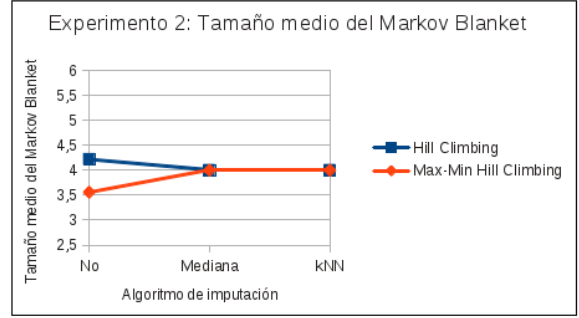
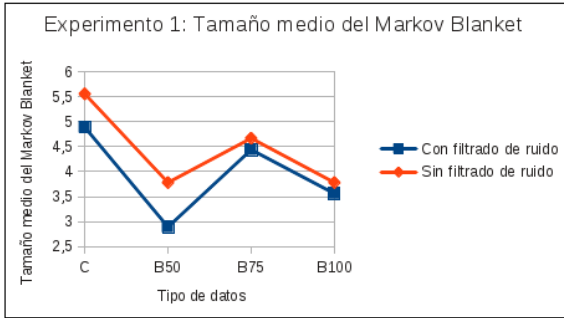


Figura 8: Métricas obtenidas para los experimentos 1 y 2 con datos de competencias generales. A la izquierda se muestran los valores obtenidos en el primer experimento y a la derecha los del segundo.

Si las comparamos con el grafo construido manualmente (sección 3.5), se observa que las similitudes son muy escasas (dos enlaces idénticos en el caso de la red con menor SHD). Además, resulta muy complicado realizar una interpretación causal de las mismas.

La imputación de valores con kNN hace que ambos algoritmos aprendan las mismas redes, probablemente porque las dependencias en los datos son más evidentes, debido a que esta imputación hace que haya más filas idénticas, aumentando las probabilidades de dependencias entre esos datos repetidos.

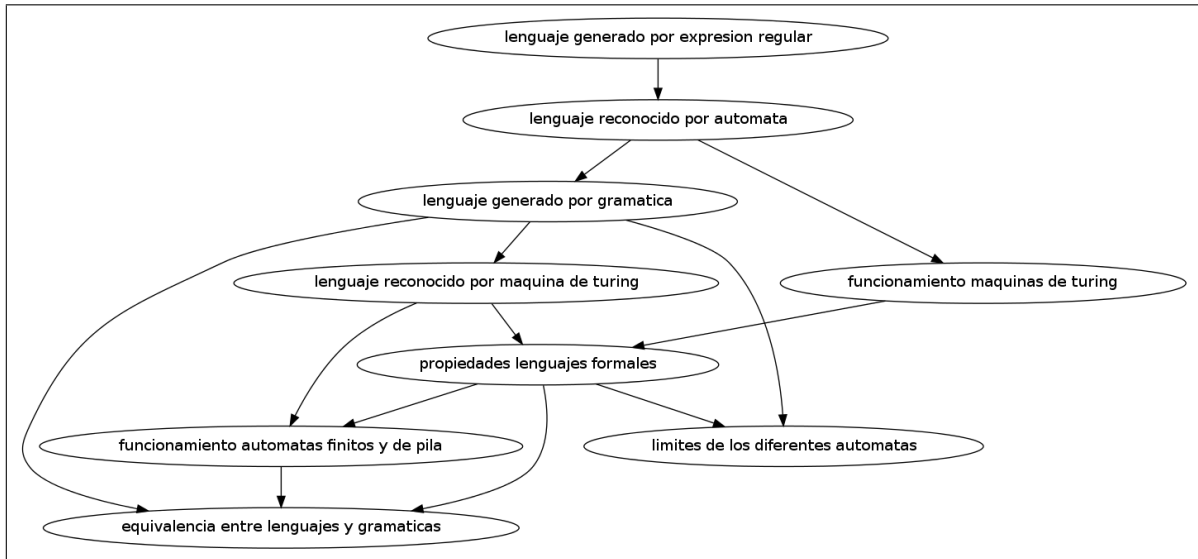


Figura 9: Red bayesiana aprendida (experimentos 1 y 2) con el algoritmo Max-Min Hill Climbing, eliminando datos con valores ausentes, utilizando datos con valores binarios obtenidos con un umbral del 100 % y filtrando posibles fuentes de ruido. El valor SHD de esta red es de 16. Las relaciones indican dependencias entre las diferentes competencias generales de la asignatura de Teoría de Autómatas I.

4.2.4. Experimentos con datos de competencias específicas

En esta sección se describen los experimentos realizados con los datos de competencias específicas de la asignatura de Teoría de Autómatas I, tal y como se describieron en la sección 4.1.2.

Análisis de valores ausentes

Los datos agregados de competencias específicas de todos los exámenes, dan lugar a un conjunto de datos con un 55 % de valores ausentes (20272/36790) y sin ningún caso completo, lo que imposibilita el uso de algunas técnicas de imputación de valores ausentes, ya que si elimináramos los casos incompletos nos quedaríamos con un conjunto vacío y, en cuanto a los algoritmos de imputación, resulta imposible utilizar kNN también a causa de la ausencia de casos completos.

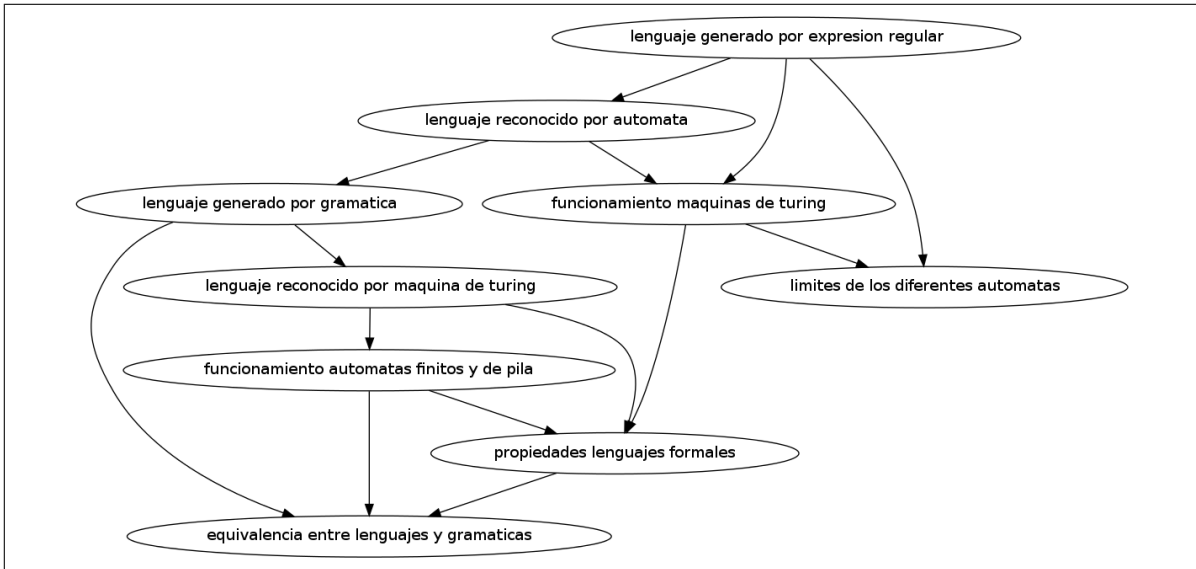


Figura 10: Red bayesiana aprendida (experimento 2) con el algoritmo Max-Min Hill Climbing, imputando valores ausentes con la técnica kNN, utilizando datos con valores binarios obtenidos con un umbral del 100 % y filtrando posibles fuentes de ruido. El valor SHD de esta red es de 18. Las relaciones indican dependencias entre las diferentes competencias generales de la asignatura de Teoría de Autómatas I.

Experimento 3: Fusión de redes bayesianas

A raíz del análisis de valores ausentes anterior, se ha decidido utilizar el algoritmo de fusión de estructuras de redes bayesianas descrito en la sección 4.2.1, de forma que se realizará un aprendizaje estructural por cada examen bajo estudio (en total 9) y, posteriormente, se fusionarán en una única estructura de red bayesiana.

En cuanto a la configuración de este experimento, tras analizar los resultados de los anteriores, se considera adecuado utilizar el filtrado de ruido y el algoritmo MMHC, ya que da mejores resultados en ausencia de valores ausentes. Además, se utilizan datos binarios con un umbral del 100 %, ya que, como se comentó en la sección 3.4.2, cada pregunta tiene asignada una única competencia específica, haciendo que las categorías que aparezcan en cada examen se repitan 2 o 3 veces a lo sumo, por lo que parece adecuado requerir que se respondan correctamente a todas ellas para dar la categoría por conocida.

Resultados

Los datos de las redes aprendidas para cada examen se encuentran en la tabla 3. Hay que observar que cuando las redes aprendidas tienen nodos sin ninguna relación (como es el caso en este experimento), estos no se tienen en cuenta en la red final, ya que el algoritmo de fusión sólo trabaja con enlaces, por lo que si un nodo no aparece relacionado en al menos una red, no se añadirá a la red fusionada. Esto hace que, de las 27 competencias específicas originales, la estructura de red resultante de este experimento tenga sólo 24 (en total, desaparecen dos, ya que la tercera corresponde a la competencia “pregunta-variada”, que se elimina por el

filtrado de ruido), como puede observarse en la misma tabla.

En la figura 11 se muestra la red resultante de fusionar las redes aprendidas para cada examen estudiado en una única red que relaciona las competencias específicas de la asignatura de Teoría de Autómatas I, de forma que las relaciones entre los nodos indican correlaciones en uno o más de los exámenes estudiados. Si observamos los enlaces en detalle, veremos que su dirección no es fácilmente interpretable de forma causal (entendiendo una relación propedéutica), aunque en algunos casos sí resulta apropiada (e.g. “conocer forma normal de chomsky” tiene como padres “conocer autómatas de pila” y “conocer gramática asociada a lenguaje”). Cuando a la dirección del enlace no se le encuentra sentido, en muchas ocasiones resulta posible interpretarla como una relación sin dirección.

A la hora de utilizar la red resultante, es interesante poder añadir/confirmar en el grafo de dependencias creado manualmente, enlaces de la red fusionada que tengan mayor fuerza. A continuación, se listan las cinco relaciones con mayor peso (indicando el peso del enlace entre paréntesis):

1. determinar equivalencia entre gramáticas \rightarrow establecer relación automatas finitos con lenguajes regulares (3189)
2. determinar si un lenguaje es independiente del contexto \rightarrow determinar si un lenguaje es regular (3042)
3. derivar producciones con gramáticas \rightarrow determinar función máquina de turing (2848)
4. determinar equivalencia autómata con gramática \rightarrow determinar equivalencia entre gramáticas (2204)
5. determinar función máquina de turing \rightarrow conocer tipos y mecanismo de control de autómatas finitos (2139)

Aunque ninguna de estas relaciones aparece en el grafo creado manualmente, muchos de los nodos relacionados están cercanos en dicho grafo. Además, se podría plantear la adición de nuevos enlaces a raíz de los enlaces obtenidos.

Red	Nodos	Nodos no relacionados	Enlaces	Factor de salto medio	Tamaño medio Markov Blanket
2004-F-1	12	1	12	1	2,17
2004-F-2	12	3	8	0,67	1,5
2004-S-1	15	5	7	0,47	0,93
2005-F-1	14	2	10	0,71	1,43
2005-F-2	9	2	6	0,67	1,56
2005-S-1	11	2	5	0,45	0,91
2006-F-1	12	8	2	0,17	0,33
2006-F-2	11	2	7	0,64	1,27
2006-S-1	14	4	9	0,64	1,29
Red fusionada	24	0	47	1,96	6,5

Tabla 3: Datos de las redes generadas en el experimento 3, incluyendo los de la estructura de red resultante de aplicar el algoritmo de fusión. Los nodos no relacionados son aquellos que no tienen ningún enlace en la red bayesiana aprendida.

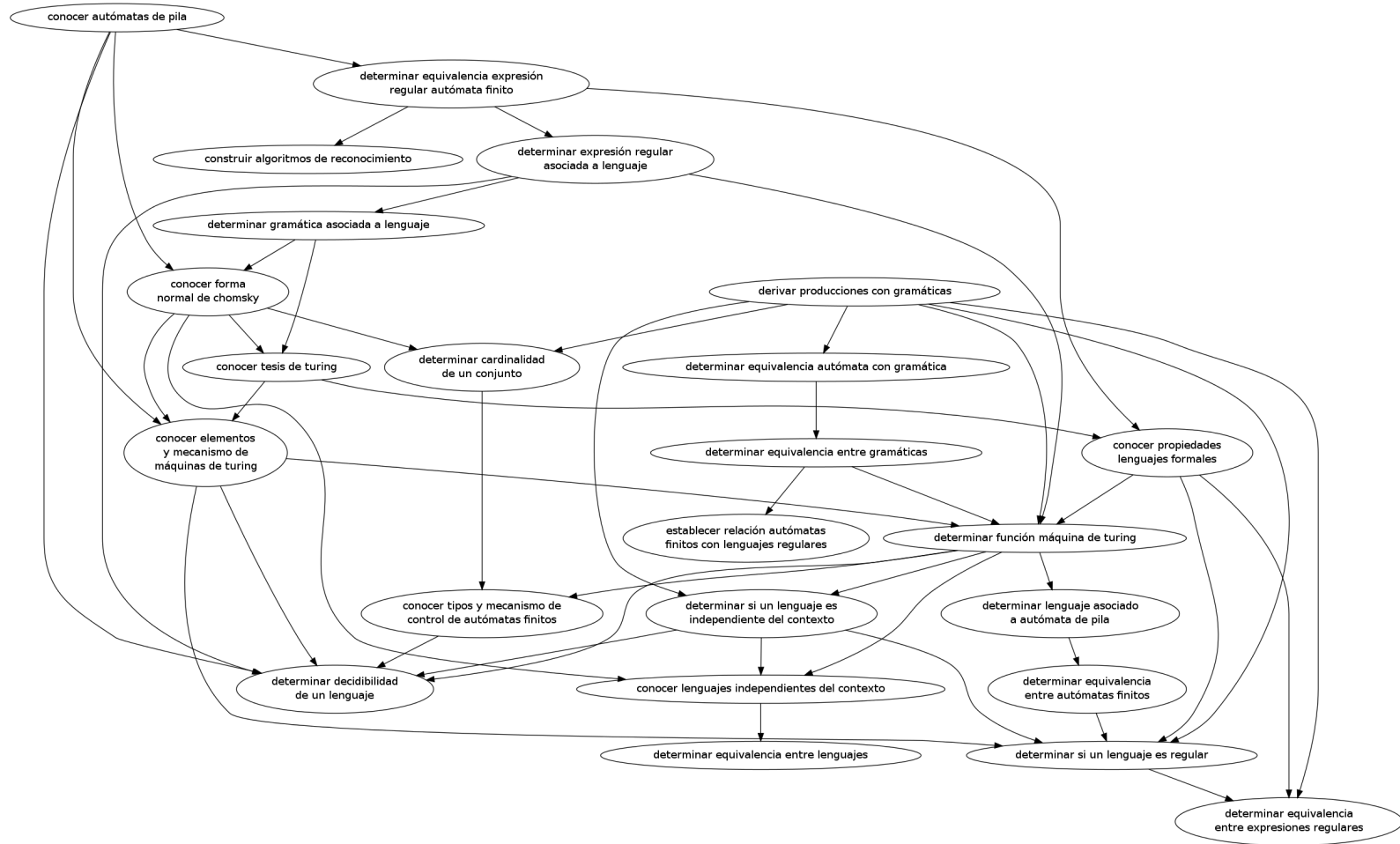


Figura 11: Estructura de red bayesiana resultante de aplicar el algoritmo de fusión de redes bayesianas (experimento 3) a las nueve redes aprendidas de forma independiente, correspondientes a los datos de competencias específicas de los exámenes bajo estudio.

4.2.5. Experimentos con datos de conceptos por tema

Los datos de conceptos tras el preprocesado descrito en la sección 4.1.3 tienen 47 atributos, uno por cada uno de los conceptos de la Tabla 2. Sin embargo, 17 de estos conceptos aparecen en menos de la mitad de los exámenes, lo cual da lugar a una gran cantidad de valores ausentes.

Por otra parte, una red aprendida con estos datos tendrá relaciones entre conceptos que pueden estar muy alejados dentro de la asignatura o que tienen distinto significado según el contexto. Sería como intentar crear un único mapa conceptual en el que aparecieran todos los conceptos de la asignatura. Esta tarea es difícil incluso para un humano, razón por la cual los mapas conceptuales suelen tener un ámbito más acotado o una menor granularidad.

Debido a estas razones, se ha decidido crear conjuntos de datos diferentes para cada tema principal de la asignatura (i.e. “autómatas finitos y lenguajes regulares”, “autómatas de pila y lenguajes independientes del contexto”, y, “máquinas de turing y lenguajes estructurados por frases”). De esta forma, acotamos el contexto de las redes aprendidas y facilitamos la interpretación de las relaciones obtenidas.

Preprocesado de los datos

La generación de diferentes conjuntos por tema supone modificar la etapa de preprocesado para utilizar sólo aquellas preguntas que han sido clasificadas como pertenecientes al tema en cuestión (recordemos que se clasificó manualmente cada pregunta con uno o más *temas* en los que se encuadraba, como se describió en la sección 3.4.2).

Además, para utilizar sólo los conceptos que más suelen estar relacionados con las preguntas de cada tema, se eliminarán aquellos que se encuentren en el primer cuartil si los ordenamos de forma creciente por número de apariciones en las preguntas de los exámenes. De esta forma, evitamos que conceptos que aparecen muy pocas veces relacionados con un determinado tema, aparezcan en la red bayesiana aprendida para ese tema.

Análisis de valores ausentes

Si agregamos los datos de todos los exámenes tras el preprocesado anterior, obtenemos tres conjuntos de datos de entrada del aprendizaje estructural de redes bayesianas. Estos datos tienen valores ausentes, ya que todos los conceptos no aparecen en todos los exámenes. En concreto, los datos agregados de conceptos para los tres temas tienen en torno al 20-30 % de valores ausentes (utilizando filtrado de posibles fuentes de ruido, como en anteriores experimentos). Este porcentaje se encuentra entre los obtenidos para datos de competencias generales y los de competencias específicas, por lo que se podrían aplicar ambas técnicas: imputación de valores ausentes y fusión de estructuras de redes bayesianas. Se ha decidido aplicar la primera, ya que

Experimento 4: Imputación de valores ausentes

En la figura 12 se muestra la estructura de la red bayesiana aprendida para los conceptos del tema 3 (Máquinas de Turing) utilizando el algoritmo MMHC, tipo de datos binario con umbral del 100 %, imputación de valores ausentes kNN y, como hemos dicho en el apartado anterior, filtrado de posibles fuentes de ruido.

Resultados

En la Tabla 4 se muestran diversos valores para las redes aprendidas en los temas 1, 2 y 3 con cada uno de los algoritmos.

Red	Nodos	Nodos no relacionados	Enlaces	Factor de salto medio	Tamaño medio Markov Blanket
Tema 1	29	0	78	2,69	9,10
Tema 2	25	0	61	2,44	8,40
Tema 3	29	0	69	2,38	8,14

Tabla 4: Propiedades de las redes bayesianas aprendidas con datos de conceptos de cada tema de la asignatura, agregados para todos los exámenes utilizando imputación de valores ausentes kNN. Se observa que no hay ningún nodo que quede aislado y que, en media, los conceptos tienen un alto número de relaciones con otros conceptos.

4.3. Construcción de mapas conceptuales a partir de redes bayesianas

En esta sección se considera cómo transformar las redes bayesianas aprendidas de forma automática, en mapas conceptuales. Se revisa la forma que tienen los mapas conceptuales creados manualmente en las primeras etapas del trabajo y se describen los pasos propuestos para realizar dicha transformación.

4.3.1. Mapas conceptuales creados manualmente

Durante la fase de clasificación manual de las preguntas, se crearon, también manualmente, diversos mapas conceptuales para familiarizarse con su construcción. Los mapas conceptuales suelen tener una estructura jerárquica (aunque hay otras, esta suele ser la más común) y una pregunta de enfoque, que especifica la cuestión que se quiere abordar con el mapa. Esta pregunta es crítica y dará lugar a diferentes mapas conceptuales dentro de un mismo dominio.

Una vez tenemos definido el dominio y la pregunta de enfoque, se podría realizar una lista de los conceptos más relevantes ordenados de más general a más concreto. Después, se construiría el mapa, que podría refinarse iterativamente las veces que hiciera falta. Durante esta construcción, es importante buscar enlaces entre diferentes zonas del mapa, lo que se conocen como *enlaces cruzados*.

En la figura 13 se muestra un mapa conceptual construido manualmente en torno a los autómatas finitos (tema 1 de la asignatura). El enlace entre “Aceptar” y “Símbolos” sería un ejemplo de enlace cruzado.

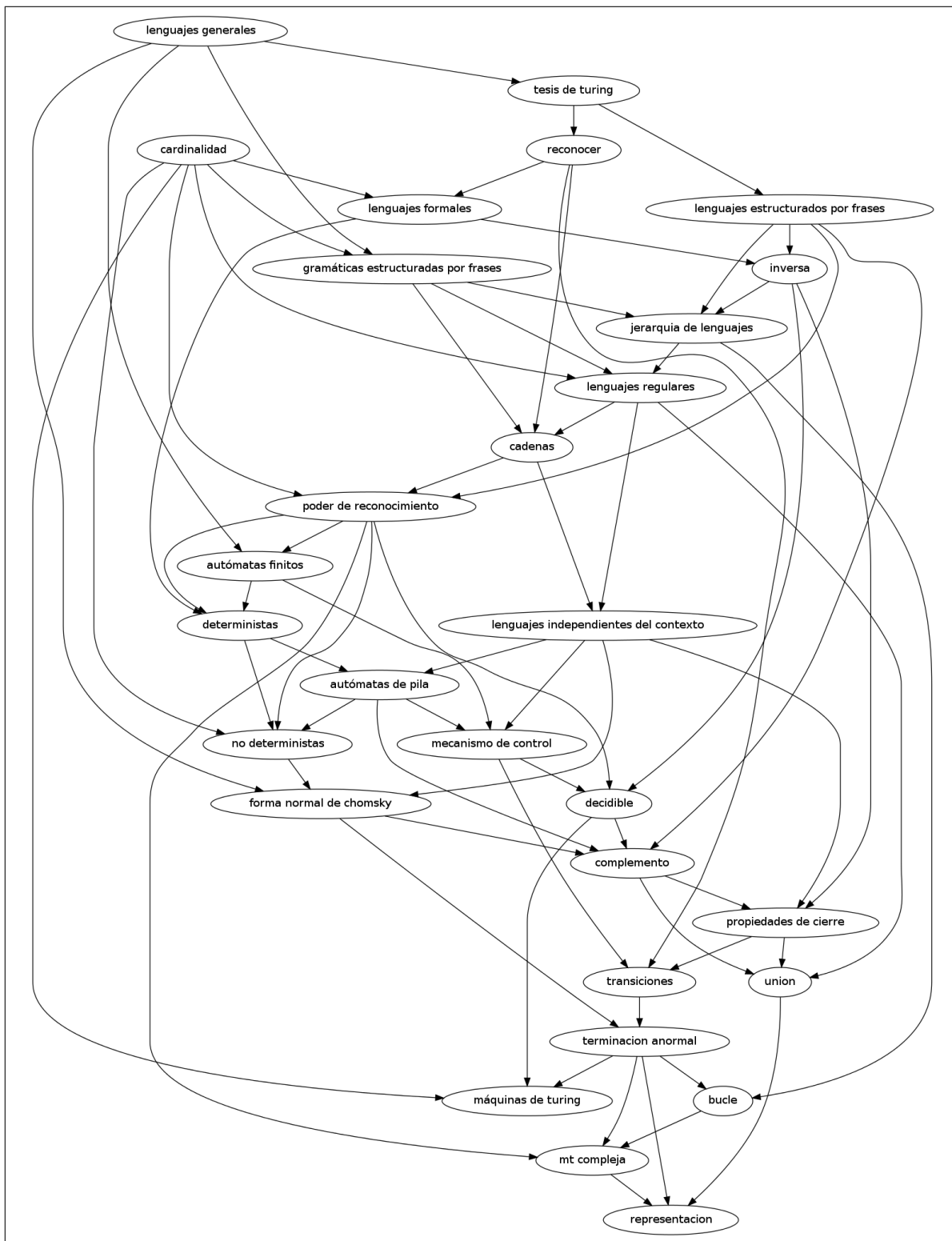


Figura 12: Red bayesiana aprendida con el algoritmo MMHC, utilizando datos de conceptos binarios con umbral 100% del tema 3, con imputación de valores ausentes kNN.

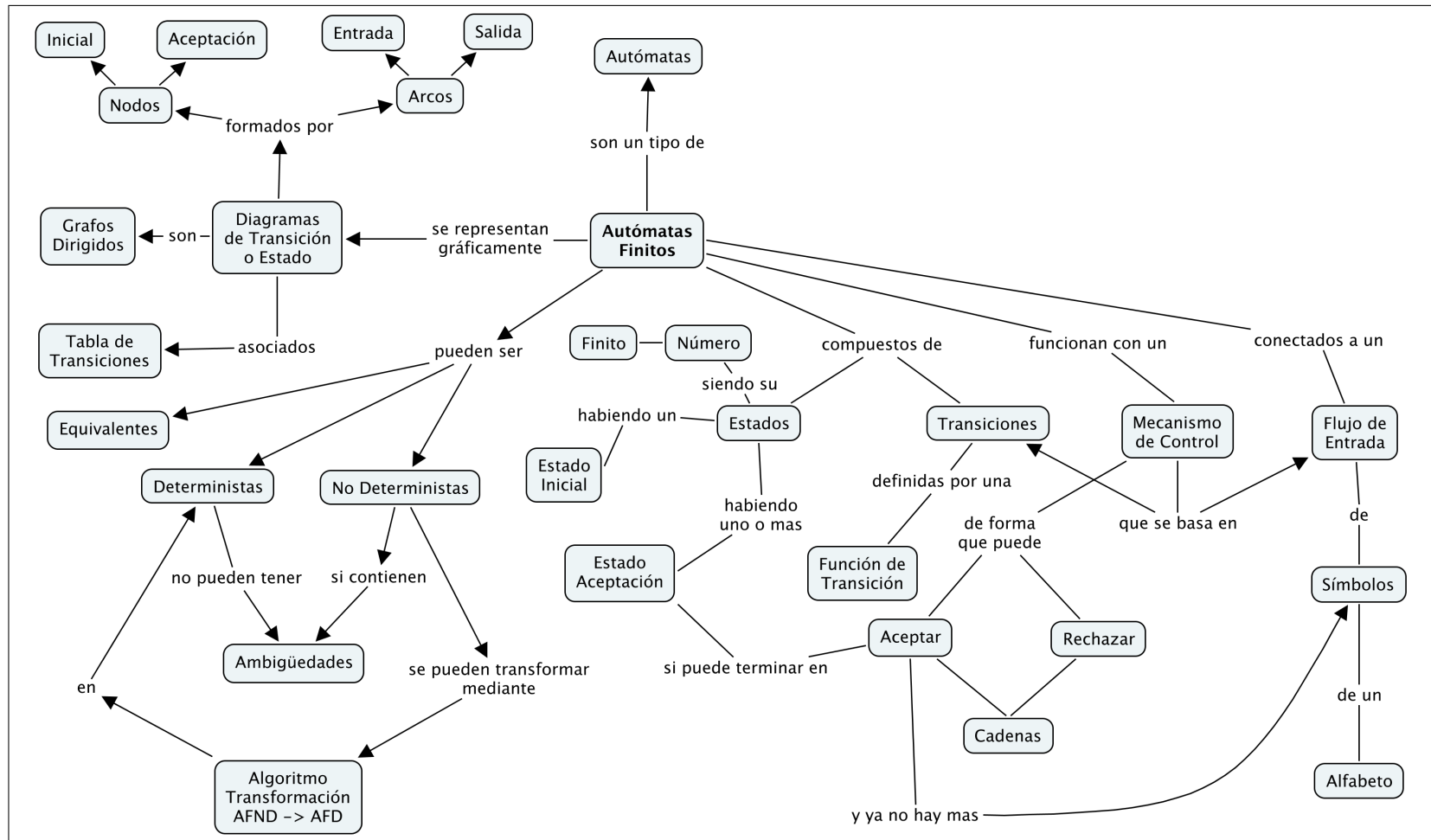


Figura 13: Mapa conceptual creado manualmente en torno a los Autómatas Finitos. La pregunta de enfoque fue “¿Qué son los autómatas finitos?”.

4.3.2. Paso 1: Algoritmo de simplificación de redes bayesianas

El primer paso en la transformación de una red bayesiana en un mapa conceptual consiste en la reducción del número de enlaces en la red, ya que los mapas conceptuales suelen tener un número menor a los presentes en las redes bayesianas aprendidas de forma automática en las secciones anteriores. Esto se debe a que los enlaces de un mapa conceptual han de formar una frase, por lo que generalmente resulta complicado que los nodos tengan más de uno o dos hijos.

Para simplificar una estructura de red bayesiana, se ha diseñado el algoritmo 2, que elimina los enlaces cuya fuerza, medida con la misma métrica utilizada para aprender dicha red (i.e. BIC), sea más pequeña. Desde un punto de vista probabilista, los enlaces que se eliminan son aquellos que contribuyen en menor medida a modelar los datos. Se describen a continuación los conceptos más relevantes del algoritmo.

Factor de salto medio Corresponde a la ratio arcos/nodos y se utiliza como criterio para decidir si seguimos eliminando enlaces del mapa conceptual o no. Un valor de 1 indica que cada nodo tiene, de media, un hijo en el grafo. Este valor umbral se ha elegido teniendo en cuenta los resultados obtenidos en las redes bayesianas aprendidas en los experimentos de la sección 4.2 y se encuentra por debajo de los valores observados más pequeños. El objetivo es simplificar el mapa conceptual dejando un número bajo de enlaces, de forma que sea más fácilmente inteligible.

Enlaces puente de un grafo Son aquellos enlaces que, si se eliminan, hacen que un grafo conectado y no dirigido, pase a tener más de una componente conectada, es decir, deje de ser conectado. Por lo tanto, el algoritmo sólo eliminará aquellos enlaces que no sean puentes, ya que el objetivo es tener un único mapa conceptual (i.e. un grafo conectado).

Algoritmo 2 Simplificación de una red bayesiana mediante la eliminación de los arcos más débiles, pero manteniendo la conectividad del grafo.

```
Función: SimplificarRB
Entrada: rb (red bayesiana)
Salida: rbs (red bayesiana simplificada)

# Se ordenan los enlaces de menor a mayor peso (i.e. relevancia)
enlaces = enlaces_ordenados(rb.enlaces(), métrica_bic)

rbs = nueva RedBayesiana(rb.enlaces())

Para cada enlace en enlaces
  Si rbs.ratio_arcos_nodos() > 1,5 entonces
    Si no rbs.es_puente(enlace) entonces
      rbs.eliminar_enlace(enlace)
    Fin Si
  Si no
    Salir del bucle
  Fin Si
Fin Para

Devolver rbs
```

4.3.3. Paso 2: Transformación de red bayesiana en mapa conceptual

Los conceptos de un mapa conceptual no son más que regularidades o patrones percibidos en eventos u objetos, o registros de eventos y objetos, designados por una etiqueta [NC08]. Las redes bayesianas, por otra parte, representan gráficamente relaciones de independencia entre diferentes variables. En nuestras redes, los nodos corresponden a conceptos o competencias, las cuales pueden considerarse también conceptos desde el punto de vista anterior.

De esta forma, la red bayesiana resultante de la simplificación de enlaces del paso 1, se puede transformar en un mapa conceptual realizando una asociación uno a uno de las variables de la red a conceptos en el mapa, y de las relaciones de la red a enlaces en el mapa.

El resultado es un mapa conceptual que tiene la misma estructura de grafo dirigido acíclico que tenía la red bayesiana, lo cual resulta adecuado, ya que un ciclo en un mapa conceptual supondría una frase que nunca se podría terminar de leer.

4.3.4. Paso 3: Asignación manual de frases de enlace

Una de las diferencias entre la estructura de una red bayesiana y un mapa conceptual es que las primeras carecen de frases de enlace, algo que resulta importante en los segundos. Sin embargo, no tenemos forma de crear estas frases de enlace de forma automática; como mucho, podríamos suponer que cada enlace lleva implícita la frase “está asociado con”, ya que eso es todo lo que sabemos: que algunos conceptos están correlacionados entre sí.

Por esta razón, se plantea un último paso en el que un experto realice la asignación de frases de enlace en los mapas conceptuales resultantes del paso anterior. Para ello, se propone un catálogo (Tabla 5) de posibles frases de enlace que pueden resultar adecuadas para el tipo de conceptos utilizados, es decir, para competencias de alumnos en una materia.

Frases de enlace (de A a B)		
A	es una subcompetencia relevante de	B
	es una subcompetencia imprescindible de	
	tiene similares subcompetencias que	
	es análoga de menor dificultad que	

Tabla 5: Catálogo de posibles frases de enlace entre dos conceptos A y B de un mapa conceptual/competencial generados de forma semi-automática en este trabajo.

4.3.5. Resultados

Los pasos propuestos se pueden aplicar tanto a las redes bayesianas aprendidas a partir de los datos agregados de todos los exámenes, como a las obtenidas como resultado del algoritmo de fusión de estructuras de redes bayesianas, ya que éstas últimas también tienen su correspondiente peso en los enlaces.

Se ha realizado una implementación de los pasos 1 y 2, propuestos en las secciones anteriores, utilizando el lenguaje de programación Python [VRF03], junto con la librería `python-graph`⁴, que permite calcular los enlaces puente de un grafo no dirigido, por lo que es nece-

⁴`python-graph`: A library for working with graphs in Python, <http://code.google.com/p/python-graph/>

sario transformar el grafo dirigido que representa el mapa conceptual, en un grafo no dirigido antes de poder calcular sus enlaces puente. Dicha implementación se ha aplicado a las redes bayesianas generadas en los cuatro experimentos descritos en la sección 4.2, dando como resultados los correspondientes mapas conceptuales (ver anexo D).

Comentarios

Tomando como ejemplo el mapa conceptual generado para el tema 3 de la asignatura (Figura 14), el cual se basa en la red bayesiana aprendida en el experimento 4 (Figura 12), podemos observar que los conceptos que aparecen están relacionados en su mayor parte con el tema en cuestión, aunque es cierto que también aparecen conceptos de otros temas.

Por otra parte, se ha reducido considerablemente el número de relaciones hasta llegar a una ratio de enlaces por nodo de 1,5 aproximadamente. Con respecto a los mapas conceptuales construidos a mano, hay algunos enlaces que se corresponden, pero la mayoría no lo hacen, por lo que la utilidad de los construidos semi-automáticamente puede ser complementar la creación de los manuales, por ejemplo, para encontrar enlaces cruzados que no se hubieran tenido en cuenta.

Nótese que no se ha realizado la asignación de frases de enlace, ya que hace falta un alto conocimiento de la materia, de sus competencias y de los problemas que los alumnos suelen enfrentar, para intentar determinar qué frases serían más adecuadas; por tanto, quedaría como trabajo futuro realizar dicha asignación.

Evaluación desde un punto de vista topológico

Si evaluamos los mapas conceptuales desde el punto de vista topológico, según los criterios planteados en [VLC08], podemos analizar a qué nivel corresponderían los mapas generados en este trabajo:

- El nivel 0 corresponde a mapas con al menos 4 conceptos conectados, por lo que los mapas creados cumplen este criterio.
- El nivel 1 corresponde a mapas con más conceptos que etiquetas de conceptos largas (i.e. con 12 caracteres o más) y con la mitad o más de frases de enlace ausentes, por lo que, aunque supongamos que nuestros enlaces tienen todos la etiqueta implícita “está relacionado con”, no se cumple la otra condición, ya que hay más nodos con etiquetas largas (normalmente formadas por más de una palabra) que con cortas. Este criterio, aunque un tanto arbitrario, nos indica que deberíamos haber separado conceptos como, por ejemplo, “autómatas finitos” en “autómatas” y “finitos”. Sin embargo, los resultados hubieran sido muy probablemente menos inteligibles, al aparecer más enlaces y nodos; además, nuestro objetivo era encontrar asociaciones entre conceptos y competencias de la materia, por lo que necesitábamos que estos aparecieran como nodos con entidad propia en el mapa, independientemente de su número de caracteres.

Por lo tanto, de acuerdo a esta clasificación topológica de mapas conceptuales, los mapas generados en este trabajo se pueden encuadrar en el nivel 0 de la misma.

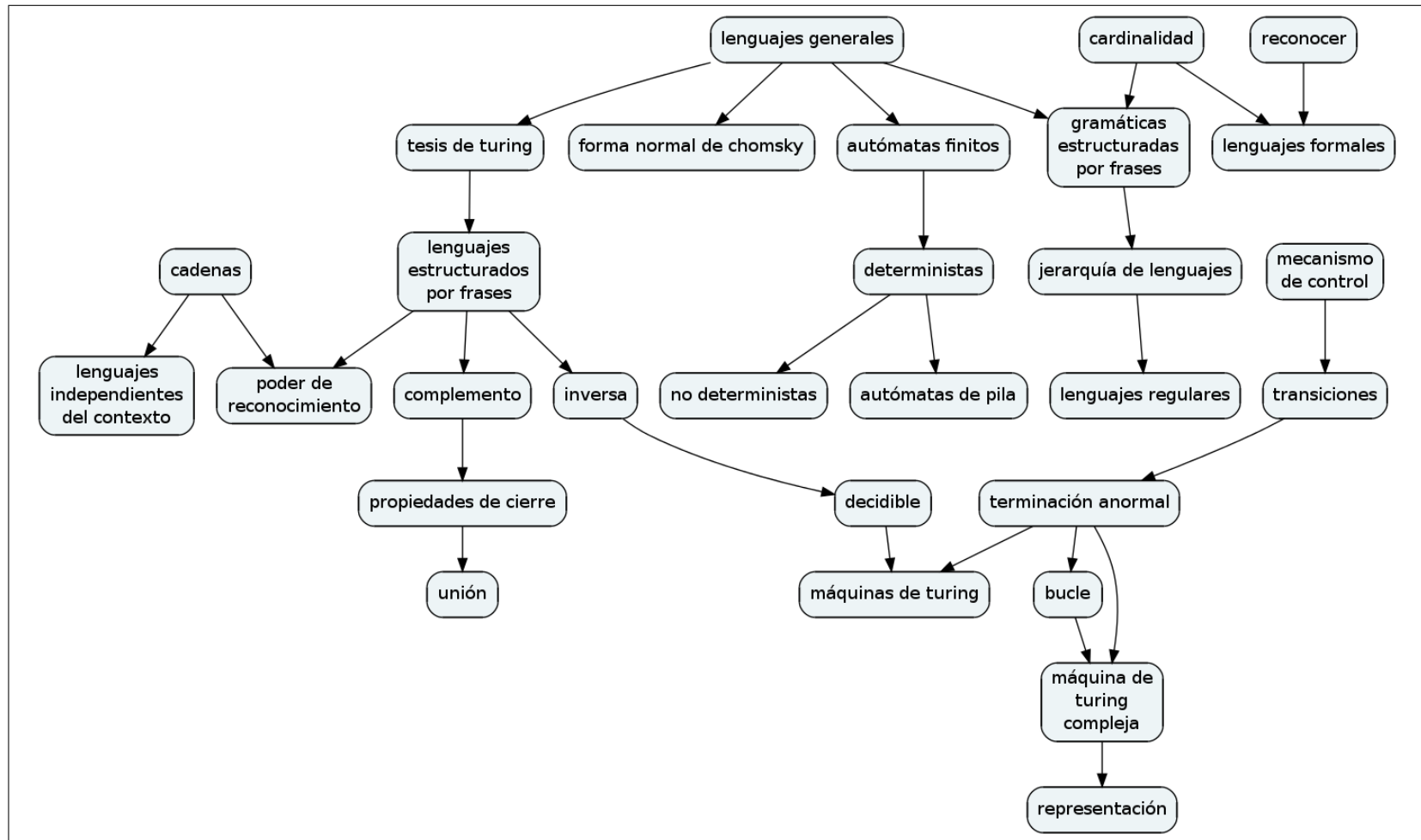


Figura 14: Mapa conceptual correspondiente al tema 3 de la asignatura Teoría de Autómatas I creado de forma semi-automática. El número de enlaces se ha reducido de 69 (en la red bayesiana original) a 29, una reducción del 68 %.

5. Sistema recomendador genérico basado en mapas conceptuales generados semi-automáticamente por métodos bayesianos

En esta sección se describe el segundo de los objetivos de este trabajo: el diseño de un sistema recomendador que utilice mapas conceptuales para recomendar al alumno el estudio de ítems de aprendizaje (e.g. conceptos o competencias) para reforzar su conocimiento, permitiéndole además visualizar mapas de su competencia en la materia. El contexto de trabajo es el de una asignatura puramente online dentro del entorno formal que supone una asignatura de una carrera oficial impartida por una universidad.

En primer lugar, se muestra un análisis del sistema planteado, comenzando por la arquitectura del sistema, mostrando los bloques más importantes, y continuando con una descripción detallada de las funcionalidades del mismo.

A continuación, se define el diseño detallado de los módulos que componen el sistema, clasificándolo dentro de los tipos de sistema recomendador existentes en el estado de la cuestión (sección 2.3).

En último lugar, se describe el prototipo realizado utilizando las preguntas, datos de respuestas y mapas conceptuales generados de forma semi-automática, de la asignatura de “Teoría de Autómatas I” utilizados y generados a lo largo de este trabajo. Se realiza una evaluación informal y se comentan algunas alternativas para la evaluación del sistema en un entorno real.

5.1. Análisis

5.1.1. Arquitectura

La arquitectura del sistema (Figura 15) se basa en un conjunto de entradas y salidas proporcionadas por profesores y alumnos, de forma que los primeros proporcionan datos al sistema y los segundos lo utilizan para mejorar su aprendizaje.

Las preguntas son un elemento central del sistema, siendo proporcionadas por los profesores, junto con clasificaciones basadas en los nodos de sendos mapas de conocimiento. Se propone el uso de las clasificaciones de la sección 3.4: competencias generales y específicas, y conceptos. Cada pregunta puede tener asociadas, además, otras propiedades, como temas relacionados, la dificultad o el tipo de pregunta (e.g. memorística).

Los mapas de conocimiento son también introducidos en el sistema por los profesores. El sistema, de forma opcional, puede sugerir relaciones en dichos mapas mediante un proceso similar al descrito en la sección 4 (generación semi-automática de mapas conceptuales), para lo que los profesores deberán introducir datos de respuestas de exámenes.

Toda la interacción de los profesores con el sistema se realiza a través de una zona de administración a la que no tienen acceso los alumnos, los cuales utilizan otra zona del sistema orientada al aprendizaje.

Entradas del sistema

- *Mapas de conocimiento*: corresponden a mapas conceptuales introducidos por los profesores, bien a través de una interfaz gráfica proporcionada por el sistema o bien mediante

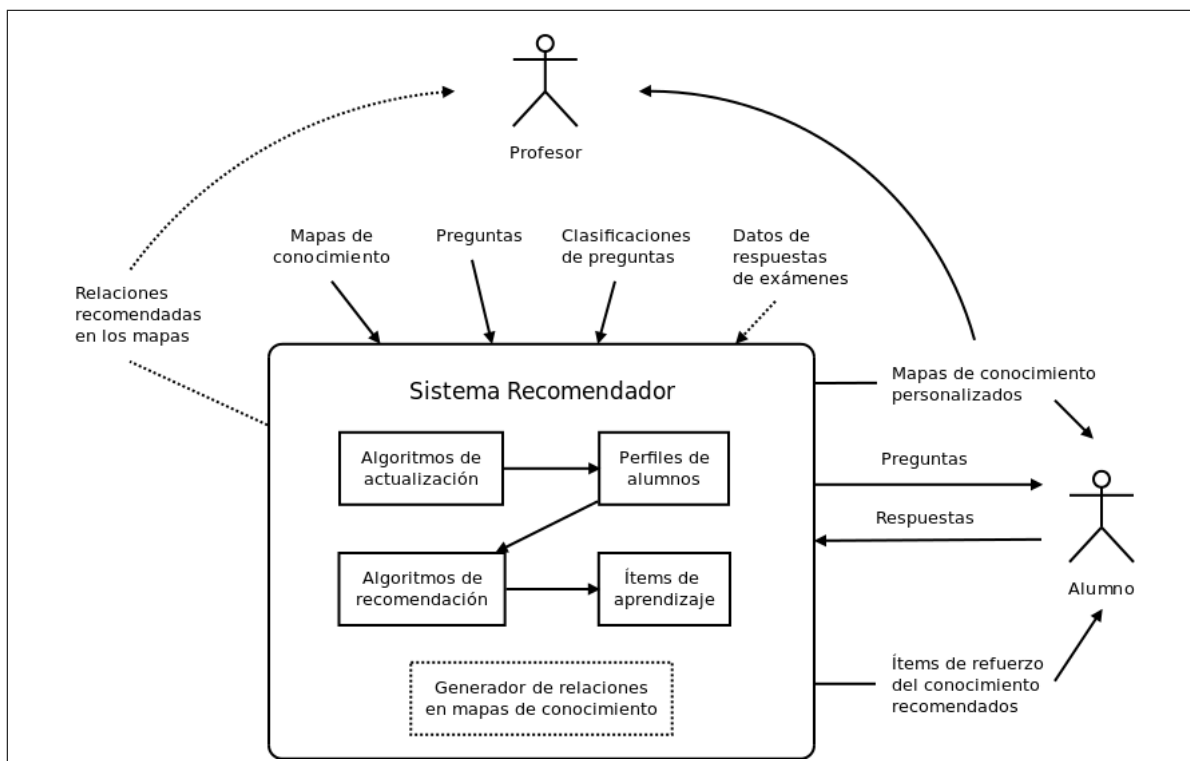


Figura 15: Arquitectura del sistema recomendador planteado, el cual recibe una serie de entradas, tanto de profesores como de alumnos, y genera unas salidas en forma de ítems de refuerzo y mapas conceptuales personalizados. Además, se muestra con líneas de puntos, la capacidad del sistema de recomendar relaciones en los mapas introducidos por los profesores si se le proporcionan datos de resultados de exámenes.

un fichero que los describa. Cada mapa utiliza conceptos pertenecientes a una clasificación determinada, por ejemplo, competencias generales, específicas o conceptos de la materia.

- *Preguntas*: son la base del sistema y vienen acompañadas de sus posibles respuestas, ya que sólo se trabaja con preguntas tipo test. En el caso de conocerse, los profesores pueden introducir su dificultad para indicar si la pregunta es más o menos difícil. Si no se conoce, el sistema supone un valor neutro.
- *Clasificaciones de preguntas*: corresponden a uno o más conceptos de los mapas de conocimiento. Los profesores han de proporcionar una clasificación para cada mapa de conocimiento, utilizando uno o más valores presentes en el mapa correspondiente.
- *Datos de respuestas de exámenes*: el sistema podrá recomendar enlaces relevantes dentro cada mapa de conocimiento utilizando datos de respuestas de preguntas de exámenes, las cuales han de estar presentes en el sistema con su correspondiente clasificación.
- *Respuestas*: El sistema realiza preguntas al alumno, introduciendo éste las respuestas que cree correctas o bien indicando al sistema que no conoce la respuesta. El sistema

utilizará estas respuestas para crear un perfil del conocimiento del alumno, lo que le permitirá realizar recomendaciones orientadas a reforzar su conocimiento.

Salidas del sistema

- *Ítems de refuerzo del conocimiento recomendados:* el sistema recomendará al alumno preguntas para practicar y conceptos para repasar, con el fin de mejorar y reforzar su conocimiento de la materia bajo estudio. Para ello, se basa en las respuestas que el alumno da a las preguntas que el sistema le plantea, así como en los mapas de conocimiento de la materia y en las clasificaciones de las preguntas.
- *Mapas de conocimiento personalizados:* dados los mapas de conocimiento de la materia introducidos por los profesores, el sistema los podrá mostrar al alumno indicando visualmente en qué conceptos necesita mejorar su conocimiento y cuáles tiene más dominados. Estos mapas también podrían ser visualizados por los profesores para conocer el estado de conocimiento de sus alumnos y poder así ajustar la programación del curso.

Módulos del sistema

- *Perfiles de alumnos:* el sistema contendrá el perfil de cada alumno dado de alta, entendido como su conocimiento en cada uno de los mapas introducidos por los profesores, o, lo que es lo mismo, en cada una de las clasificaciones de preguntas utilizadas.
- *Algoritmos de actualización:* el sistema actualizará el perfil de los alumnos utilizando las respuesta dadas por estos a diferentes preguntas propuestas por el sistema.
- *Ítems de aprendizaje:* corresponden a elementos que el sistema puede recomendar a los alumnos. Pueden ser elementos de los mapas de conocimiento que el alumno debería repasar, o preguntas para que el alumno pueda practicar en aquellas áreas de conocimiento que menos conoce.
- *Algoritmos de recomendación:* definen cómo recomendará el sistema los ítems de aprendizaje en función del perfil del alumno.
- *Generador de relaciones en mapas de conocimiento:* se utiliza como parte de la introducción de mapas de conocimiento en el sistema por parte de los profesores, de forma que éste puede recomendar la adición de enlaces en los mapas utilizando datos de respuestas de exámenes (o de cualquier otra fuente que crea de utilidad).

5.1.2. Funcionalidades

El sistema recomendador propondrá preguntas al alumno, el cual las responderá generando una serie de respuestas que el sistema procesará con el objetivo de generar el perfil de conocimiento del alumno en relación a las diferentes clasificaciones, para lo que se valdrá de los datos de entrada del sistema (i.e. mapas de conocimiento, preguntas y clasificaciones). El sistema recomendará el repaso de ítems de conceptos de los mapas (pueden ser conceptos de la materia, competencias o lo que el profesor decida introducir) y la práctica de nuevas preguntas en función del nivel de conocimiento del alumno.

Gestión de mapas de conocimiento

Los profesores podrán añadir, editar y eliminar, mapas de conocimiento del sistema. Estos mapas corresponden a clasificaciones de las preguntas en forma de competencias, conceptos de la materia u otras clasificaciones que crea oportunas. Los conceptos que aparezcan en los mapas se utilizarán posteriormente para clasificar las preguntas.

El sistema proporciona a los profesores un sistema de generación automática de relaciones en dichos mapas mediante los mecanismos descritos en la sección 4 de este trabajo, para lo que necesitará un conjunto de datos de respuestas de exámenes, incluyendo la clasificación de las preguntas. Como resultado del proceso, el sistema propondrá nuevos enlaces en el mapa en cuestión, pudiendo el profesor eliminarlos o conservarlos a su voluntad.

Gestión de preguntas y sus propiedades

Las preguntas han de ser introducidas en el sistema por los profesores, acompañando cada una con los siguientes campos opcionales: dificultad, temas relacionados y tipo de pregunta. Cada pregunta puede tener asociado uno o más mapas de conocimiento, en los que la pregunta es clasificada por los profesores asignando uno o más de los conceptos presentes en dichos mapas.

Visualización y actualización del perfil del alumno

Los alumnos podrán visualizar su perfil de conocimiento de forma gráfica mediante mapas de conocimiento que indicarán de forma visual qué conceptos se dominan más o menos. Esto permitirá, tanto a alumnos como a profesores, poder echar un vistazo rápido a las áreas de la asignatura para saber qué conceptos de la materia debe repasar o que competencias debe practicar, por ejemplo.

La actualización del perfil de los alumnos se realizará mediante tandas de preguntas, a modo de examen, que el alumno podrá realizar periódicamente para mantener el sistema actualizado. Es aconsejable realizar alguna de estas tandas antes de solicitar al sistema recomendaciones de temas, sugerencias de preguntas prácticas o visualizaciones gráficas del conocimiento.

Recomendación de ítems de aprendizaje para reforzar el conocimiento

El sistema recomendará al alumno, basándose en su perfil, ítems de aprendizaje (preguntas y conceptos de los mapas de conocimiento) con el fin de acelerar el aprendizaje y encamilarlo hacia las partes de la materia donde el alumno tiene más carencias de conocimiento.

Las preguntas para practicar que se proponen al alumno no cuentan para la actualización del perfil, ya que se plantean en un contexto de aprendizaje y no de evaluación.

5.2. Diseño

En esta sección, se describe el diseño de los aspectos más relevantes del sistema recomendador, comenzando por el tipo de sistema recomendador que se plantea, teniendo en cuenta los tipos más conocidos. Posteriormente, veremos cómo se representa el perfil de conocimiento de un alumno para, después, poder recomendar los ítems de aprendizaje más adecuados. Por

último, veremos cómo se realizará la visualización del conocimiento mediante los mapas de conocimiento.

5.2.1. Tipo de sistema recomendador

Como se describió en la revisión del estado de la cuestión en torno a los sistemas recomendadores (sección 2.3), existen tres tipos de sistemas recomendadores principales: los que realizan un filtrado colaborativo, los que realizan un filtrado basado en el contenido y los métodos híbridos. El tipo de recomendador que se va a utilizar está basado en el contenido, es decir, las recomendaciones se basan en el perfil del usuario, que se actualizará automáticamente mediante las respuestas que dé a las preguntas propuestas, y en las descripciones de los ítems, que corresponderán con las clasificaciones asociadas a cada pregunta.

La elección de este tipo de recomendador se debe a las siguientes razones:

- El filtrado colaborativo depende de la existencia de información de otros usuarios para recomendar nuevos ítems, mientras que el basado en el contenido no la necesita.
- El filtrado colaborativo encaja mejor cuando los usuarios muestran sus preferencias sobre los ítems (e.g. me gusta o no me gusta), mientras que en nuestro objetivo no muestran una preferencia sino que demuestran su conocimiento del ítem.
- El filtrado basado en el contenido nos permite dar más importancia al conocimiento del alumno mediante el uso de perfiles que luego se utilizan para realizar recomendaciones.

5.2.2. Descripción de los ítems

El sistema trabajará con dos tipos de ítems: clasificaciones (i.e. mapas de conocimiento) y preguntas. Las primeras no tienen información que las describa, sino que aparecen asociadas a las segundas (las preguntas). Son éstas las que se describen mediante las distintas clasificaciones realizadas, de forma que cada pregunta tendrá asociada la siguiente tupla de información: <clasificaciones en cada mapa de conocimiento, propiedades generales (e.g. dificultad)>

5.2.3. Perfil del alumno

El perfil del alumno se representará con varios vectores de pesos asociados a cada una de las diferentes clasificaciones (i.e. mapas de conocimiento). En concreto, si suponemos que se utilizan los tres mapas de conocimiento utilizados a lo largo de este trabajo, tendríamos:

- Un vector de pesos de conceptos de la materia, donde cada elemento corresponderá con el conocimiento del alumno en ese concepto, por ejemplo, “máquinas de turing” o “lenguajes regulares”. Estos conceptos corresponden con los que se asociaron manualmente a cada pregunta, tal y como se describe en la sección 3.4.1. En caso de que quisiéramos tener un mapa de conocimiento distinto para cada tema (como se planteó en la sección 4.2.5), se tendría un vector por cada tema/mapa.
- Un vector de pesos de competencias generales, donde cada elemento corresponde con el conocimiento del alumno en cada una de las nueve competencias generales definidas. Este vector permitiría proponerle preguntas de competencias generales que no domine.

- Un vector de pesos de competencias específicas, donde cada elemento corresponde con el conocimiento del alumno en cada competencia específica. Este vector permitiría proponerle preguntas de competencias específicas que no domine, igual que en el caso anterior.

Actualización del perfil

Cada uno de los vectores que forman el perfil de un alumno, tendrá la siguiente forma:

$$w = \langle w_1, w_2, \dots, w_n \rangle, \text{ con } w_i \in [-1, 1]$$

siendo w_i el peso para el concepto i de la clasificación a la que corresponda el vector y n el número de conceptos del mapa de conocimiento asociado a dicha clasificación. La actualización de los vectores que representan el perfil del alumno se basa en las respuestas que éste va dando a las preguntas que el propio sistema recomendador le formula. Estas tienen la siguiente forma:

$$r = \langle r_1, r_2, \dots, r_m \rangle, \text{ con } r_i \in \{\text{acierto}, \text{fallo}\}$$

siendo m el nº de preguntas presentadas al alumno durante una tanda de preguntas de evaluación. Una vez el sistema recibe las respuestas, se utiliza el algoritmo 3 para actualizar los vectores de pesos de cada vector del perfil mediante el uso de las clasificaciones asociadas a las preguntas y los mapas de conocimiento asociados a cada clasificación.

Conceptos relacionados en los mapas de conocimiento

A la hora de actualizar los vectores de pesos, no sólo se tienen en cuenta los conceptos directamente relacionadas con las preguntas respondidas por el alumno, sino que se aprovecha el mapa de conocimiento asociado a cada clasificación para determinar qué otros conceptos pueden influir en el conocimiento de los primeros.

Ajuste de los pesos usando la dificultad

El ajuste de los pesos de los vectores del perfil tiene dos pares de componentes: uno para los conceptos directamente asociados a las preguntas, $(K_1, K_2 \cdot d_i)$, y otro para aquellos relacionados con los anteriores, $(K_3, K_4 \cdot d_i)$, es decir, que tienen influencia en la comprensión de los primeros, pero no están directamente asociados a la pregunta.

Los valores de las constantes K_1 , K_2 , K_3 y K_4 definen qué efecto tiene cada respuesta fallida o acertada en el nivel de conocimiento de las clasificaciones asociadas a la pregunta. Se plantea el uso de los siguientes valores $K_1 = 0, 1$ y $K_2 = 0, 2$ para los conceptos directamente asociados a la pregunta y $K_3 = 0, 025$ y $K_4 = 0, 05$ para los conceptos indirectamente relacionados. De esta forma, se da más importancia a aquellos conceptos asociados directamente a las preguntas y se tiene en cuenta la dificultad de las mismas a la hora de actualizar el perfil. Las constantes K_j han sido elegidas de forma que sea necesario responder correctamente unas cuatro preguntas asociadas directamente a un concepto, sin fallar ninguna directa o indirectamente relacionada con éste, para que el sistema considere que el alumno domina dicho concepto.

Sea p_a el porcentaje de alumnos que respondieron correctamente a la pregunta en el examen en que apareció y f_c un factor de corrección relativo al nivel de conocimiento de todos los

Algoritmo 3 Actualización del vector de pesos de una clasificación cualquiera del perfil de un alumno. Se puede observar que se utilizan diferentes constantes para los conceptos directamente relacionados con las preguntas respondidas por el alumno (K_1 y K_2), que para aquellos relacionados con los primeros en el mapa de conocimiento (K_3 y K_4). De esta forma, se le dará más importancia a los primeros que a los segundos.

```

Procedimiento: ActualizarVectorPesos
Entradas: w (vector de pesos), c (clasificación asociada al vector de pesos),
          r (vector de respuestas), p (ítems de las preguntas planteadas),
          mc (mapa de conocimiento asociado al vector de pesos)

Para cada i = 1..p.tamaño()
  Para cada concepto en p[i].conceptos_asociados(c)
    Si r[i] es correcta
      w[concepto] += K1 + (K2 * p[i].dificultad())
    Sino
      w[concepto] -= K1 + (K2 * p[i].dificultad())
    Fin Si

    conceptos_relacionados = mc.nodos_vecinos(concepto)
    Para cada cr en conceptos_relacionados
      Si r[i] es correcta
        w[cr] += K3 + (K4 * p[i].dificultad())
      Sino
        w[cr] -= K3 + (K4 * p[i].dificultad())
      Fin Si
    Fin Para

    Si w[concepto] < -1.0
      w[concepto] = -1.0
    Sino Si w[concepto] > 1.0
      w[concepto] = 1.0
    Fin Si
  Fin Para
Fin Para

```

alumnos que realizaron dicho examen; la dificultad de la pregunta i , d_i , se calcularía mediante la siguiente fórmula:

$$d_i = (1 - p_a) \cdot f_c$$

En caso de que no se dispusiera de los valores p_a y f_c , lo adecuado sería usar $d_i = 0,5$ que corresponde a una dificultad media.

Sea n_e la nota media de los alumnos en el examen e al que pertenece la pregunta i , el factor de corrección f_c para dicha pregunta, se calcularía de la siguiente forma:

$$f_c = \frac{n_e}{5}$$

Se puede observar que este valor hace que la dificultad crezca cuanto mejor sea la nota media (i.e. más cercana al 10), y decrezca conforme sea peor (i.e. más cercana al 0).

5.2.4. Recomendación de ítems de aprendizaje

El sistema recomendará ítems de los dos tipos descritos en la sección 5.2.2, de forma que se complemente el repaso de conceptos por parte del alumno con la posterior realización de

ejercicios mediante las preguntas que le propondrá el sistema.

Ítems de conceptos

El sistema recomendará un número determinado de conceptos de los mapas de conocimiento, elegidos del perfil del alumno, de forma que se muestren en primer lugar los que tengan valores más negativos, es decir, aquellos de los que el alumno tenga un peor conocimiento. La recomendación de conceptos es un caso especial en un sistema recomendador, ya que no se tienen en cuenta las descripciones de los ítems (pues los conceptos no tienen tales).

Ítems de preguntas

El sistema podrá recomendar preguntas para reforzar el conocimiento del alumno utilizando para ello los datos de cualquiera de las clasificaciones, por ejemplo, competencias generales o específicas. Para ello, supondremos que cada pregunta tiene asociado un vector de ceros y unos para cada clasificación, donde el cero significa que la pregunta no tiene asociado ese concepto y el uno que sí lo tiene. Si w es el vector de pesos (tal y como se define en la sección 5.2.3), y sea v el vector asociado a la pregunta

$$v = \langle v_1, v_2, \dots, v_n \rangle, \text{ con } v_i \in \{0, 1\}$$

la distancia con cada pregunta se calcularía de la siguiente forma:

$$d_p = \sqrt{(w_1 - v_1)^2 + \dots + (w_n - v_n)^2}$$

que corresponde a la distancia euclídea entre el estado de conocimiento actual y una pregunta en concreto. Se sugerirá al alumno la práctica de aquellas preguntas que se encuentren a mayor distancia de su estado de conocimiento (i.e. aquellas que tengan el potencial de mejorar o reforzar lo máximo posible dicho conocimiento).

5.2.5. Visualización de mapas de conocimiento

El perfil del alumno, es decir, los vectores correspondientes a cada clasificación o mapa de conocimientos, se pueden utilizar para mostrar, tanto al alumno como al profesor, una versión personalizada de dichos mapas de conocimiento, de forma que sea posible ver de un vistazo qué conceptos tiene más o menos dominados el alumno, así como determinar cuáles de ellos puede ser necesario practicar o reforzar.

Se plantea la visualización de los nodos coloreados en función del peso de cada concepto en el perfil del alumno, utilizando los siguientes rangos:

$$\text{color}(i) = \begin{cases} w(i) > 0,75 & \text{verde} \\ w(i) > 0,25 \wedge w(i) \leq 0,75 & \text{verde claro} \\ w(i) > -0,25 \wedge w(i) \leq 0,25 & \text{blanco} \\ w(i) \leq -0,25 \wedge w(i) > -0,75 & \text{rojo claro} \\ w(i) \leq -0,75 & \text{rojo} \end{cases}$$

siendo $w(i)$ el peso del concepto i dentro de un determinado mapa de conocimientos. En la figura X se muestra un ejemplo ficticio de cómo podría resultar esta visualización en uno de los mapas de competencias generales generados en este trabajo.

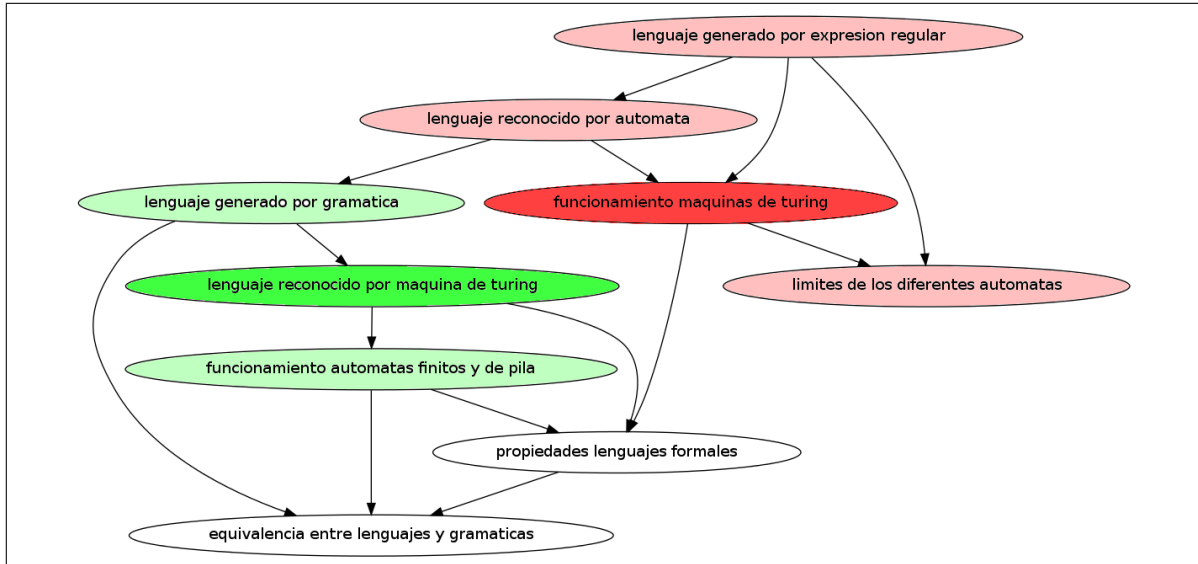


Figura 16: Ejemplo de visualización del mapa de competencias generales de un alumno utilizando información del perfil. En color rojo aparecen las competencias peor conocidas (cuanto más rojo peor) y en verde las mejor conocidas (cuanto más verde mejor). En blanco se muestran aquellas cuyo conocimiento se encuentra en valores medios.

5.3. Prototipo

Se ha desarrollado un prototipo web utilizando la plataforma Google App Engine [Zah09] con Python, lo que nos ha permitido reutilizar el código desarrollado para los experimentos de generación semi-automática de mapas conceptuales de la sección 4.2. El prototipo es accesible desde Internet⁵ e implementa algunas de las funcionalidades del sistema recomendador propuesto.

Como continuación a las fases anteriores del trabajo, se utilizarán las preguntas y mapas de conocimiento (generados de forma semi-automática) de la asignatura de “Teoría de Autómatas I” impartida en la UNED. Este prototipo se centra sólomente en la zona de los alumnos, ya que se ha considerado más relevante y, además, se dispone de los datos generados en las secciones anteriores de este trabajo (e.g. preguntas, clasificaciones y mapas conceptuales), de forma que no es imprescindible implementar la zona de administración.

En la figura 17 se muestra una captura de la pantalla de inicio del prototipo y en la figura 18 se muestra la pantalla en que se realiza la recomendación de conceptos y preguntas.

⁵<http://recomendador-teoria-automatas.appspot.com/>

Sistema Recomendador - Teoría de Autómatas I

Si no conoces la respuesta a una pregunta, deja marcada la opción "No sé la respuesta". Esto permitirá al sistema aprender mejor tus conocimientos actuales.

Pregunta 1

Indique cuál de las siguientes afirmaciones es verdadera:

- Nunca se puede afirmar con seguridad que un autómata finito no determinista acepta una cadena
- Un autómata finito no determinista sólo puede utilizarse para aceptar lenguajes finitos
- Un autómata finito no determinista acepta una cadena cuando es posible que su análisis deje a la máquina en un estado de aceptación
- No sé la respuesta

Pregunta 2

El número de estados mínimo para una máquina de Turing no determinista es

- Uno
- Dos
- Ninguno de los anteriores
- No sé la respuesta

Pregunta 3

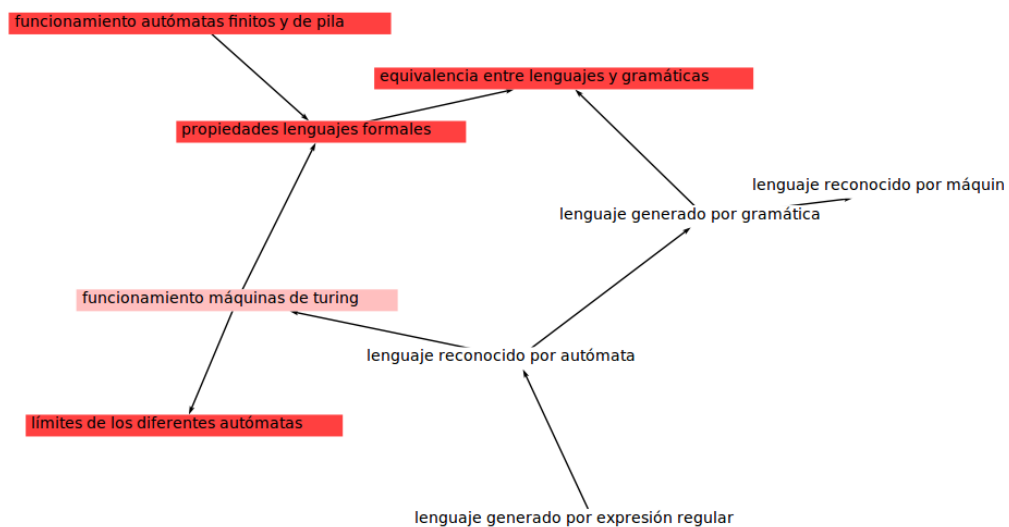
Siendo R una expresión regular, indique cuál de las siguientes relaciones es falsa:

Figura 17: Captura de la pantalla de inicio del prototipo de sistema recomendador. En ella se plantean al alumno una serie de preguntas que servirán para actualizar su perfil y realizar las recomendaciones y la visualización de su mapa de competencias.



Sistema Recomendador - Teoría de Autómatas I

Mapa de competencias



Conceptos de repaso recomendados

Teniendo en cuenta las respuestas que has dado a las preguntas anteriores, te recomendamos que revises los siguientes conceptos:

- complemento (-0.750)
- propiedades de cierre (-0.600)
- lenguajes regulares (-0.600)
- autómatas finitos (-0.450)
- cadenas (-0.400)

Figura 18: Captura de la pantalla de recomendaciones del prototipo de sistema recomendador. Se muestra el mapa de competencias generales correspondiente al perfil del alumno y los conceptos de repaso recomendados. Aunque no aparecen en la imagen, debajo de estos se plantean las preguntas de práctica para el alumno.

5.3.1. Funcionalidad

Se implementan los módulos principales del sistema recomendador, excepto porque: no tiene memoria, es decir, no es capaz de guardar el perfil del alumno y actualizarlo en sucesivas preguntas; y, finalmente, no utiliza la dificultad de las preguntas sino que supone un peso similar para todas ellas.

Datos de preguntas y mapas utilizados

Los mapas de conocimiento utilizados son los mapas conceptuales generados de forma semi-automática en la sección 4.3 (como mapa de conceptos de la materia se ha utilizado el correspondiente al tema 2). Las preguntas también corresponden a las utilizadas a lo largo de este trabajo (sección 3.3), así como sus correspondientes clasificaciones manuales (sección 3.4).

Datos utilizados para cada tipo de ítem recomendado

Los ítems de conceptos recomendados corresponden a conceptos de la materia (los cinco con menor peso en el vector). Aunque también podrían ser competencias generales o específicas, se ha considerado más adecuado recomendar el repaso de conceptos de la materia. Por otro lado, los ítems de preguntas recomendadas corresponden a competencias generales y específicas, en concreto, se sugieren cinco preguntas para practicar, las cuales se obtienen utilizando el método descrito en la sección 5.2.4.

Preguntas propuestas al alumno

Se ha elegido plantear cinco preguntas en cada tanda de ejercicios prácticos, buscando el equilibrio entre información recibida y tiempo requerido para responderlas. Las preguntas se obtienen utilizando los vectores de competencias generales y específicas del perfil del alumno, alternándose uno u otro para cada una de las cinco preguntas planteadas.

Se ha supuesto que la dificultad de las preguntas es siempre 0,5 por lo que los pesos se actualizarán tal y como se describió en la sección 5.2.3, utilizando los valores $K_1 = 0,1$ $K_2 = 0,2$ $K_3 = 0,025$ $K_4 = 0,05$ recomendados en la misma.

Preguntas en blanco

Es importante evitar que el alumno seleccione una pregunta al azar, ya que esto puede dar información errónea al sistema en caso de acertar. Para ello, se ha añadido un texto antes de las preguntas que indica al alumno este hecho. Además, se ha hecho explícita la opción de no conocer la respuesta mediante una cuarta respuesta posible: “No sé la respuesta”, que además está seleccionada por defecto. De esta forma, evitamos que el alumno sienta la necesidad de elegir una opción (antes que dejar en blanco la pregunta), y hacemos “normal” (ya que no estamos en un examen) que el alumno pueda no conocer la respuesta.

5.3.2. Evaluación

Se ha comprobado que el prototipo realiza las actualizaciones de forma correcta y que los ítems de conceptos y preguntas que recomienda tienen sentido desde el punto de vista lógico, es decir, que efectivamente recomienda ítems de preguntas y conceptos relacionados con las áreas de menor conocimiento del alumno.

Comprobación de las recomendaciones

Se muestra a continuación un ejemplo de funcionamiento del prototipo desarrollado. En la Tabla 6 tenemos toda la información de las preguntas que se muestran al alumno. Esta información representa la descripción de los ítems de preguntas.

Examen	Pregunta	Conceptos	Temas	Competencias generales	Competencias específicas
2006-F-2	9	autómatas finitos, no deterministas, cardinalidad, estados	1	propiedades lenguajes formales, funcionamiento autómatas finitos y de pila	conocer tipos y mecanismo de control de autómatas finitos
2004-F-1	8	máquinas de turing, deterministas, estados	3	funcionamiento máquinas de turing	conocer elementos y mecanismo de máquinas de turing
2006-S-1	5	expresiones regulares, estrella de kleene, equivalentes, cadenas	1	lenguaje generado por expresión regular	determinar equivalencia entre expresiones regulares
2006-F-1	3	autómatas de pila, deterministas, no deterministas, poder de reconocimiento	2	funcionamiento autómatas finitos y de pila	conocer autómatas de pila
2006-S-1	1	gramáticas regulares, derivación, cadenas	1	lenguaje generado por gramática	derivar producciones con gramáticas

Tabla 6: Información de las preguntas que se muestran al alumno en el ejemplo descrito.

Supongamos que el alumno responde correctamente a las preguntas 2006-F-2-09 y 2006-F-1-03, que están relacionadas con el funcionamiento de los autómatas finitos y de pila, pero falla las otras tres. En la tabla 7 se muestra el perfil del alumno una vez añadida la información de todas las respuestas.

Los ítems de conceptos de la materia recomendados, cuyo repaso se sugiere, son los seis que aparecen al final de la lista de conceptos de la tabla 7, los cuales resultan adecuados y encajan con las preguntas falladas. Se observa que algunos de estos conceptos pueden ser confusos si los aislamos del contexto; por esta razón, se planteó la visualización en forma de mapa coloreado de la sección 5.2.5.

Por otro lado, los ítems de preguntas recomendadas han sido los siguientes (junto con la distancia calculada según el mecanismo descrito en 5.2.4):

- Pregunta 14, examen 2006-S-1 ($d_p = 1,759$), utilizando el vector de competencias generales. Está clasificada como: “lenguaje generado por expresión regular” y “lenguaje reconocido por autómata”.
- Pregunta 3, examen 2004-F-1 ($d_p = 1,273$), utilizando el vector de competencias específicas. Está clasificada como “conocer elementos y mecanismo de máquinas de turing”.
- Pregunta 1, examen 2006-F-2 ($d_p = 1,759$), utilizando el vector de competencias generales. Está clasificada como: “lenguaje reconocido por autómata” y “lenguaje generado por gramática”.
- Pregunta 15, examen 2006-F-1 ($d_p = 1,273$), utilizando el vector de competencias específicas. Está clasificada como “conocer elementos y mecanismo de máquinas de turing”.
- Pregunta 15, examen 2004-F-1 ($d_p = 1,759$), utilizando el vector de competencias generales. Está clasificada como: “lenguaje reconocido por autómata” y “lenguaje generado por expresión regular”.

Se puede observar que se recomiendan preguntas que tienen clasificaciones similares a las de menor valor en el perfil (Tabla 7). Además, se prefieren aquellas que tienen más de una competencia en la que alumno tenga bajo conocimiento, lo cual tiene sentido ya que su práctica tiene el potencial de aumentar el conocimiento en un mayor número de competencias.

Evaluación en un entorno real

Además de la evaluación informal realizada en este trabajo, se podría plantear el uso de este sistema recomendador con alumnos reales, de forma que se pudieran obtener resultados subjetivos (e.g. grado de satisfacción y utilidad de la herramienta) y objetivos (e.g. correlación entre alumnos con un perfil alto y alumnos aprobados). Esto permitiría ajustar algunos parámetros del sistema y valorar su capacidad para trabajar con otras asignaturas, lo cual posiblemente diera lugar a mejoras en el sistema que lo hicieran más general. Por supuesto, antes sería necesario completar la implementación del prototipo con todas las funcionalidades propuestas.

Competencias generales	<p>funcionamiento autómatas finitos y de pila = 0,45</p> <p>propiedades lenguajes formales = 0,25</p> <p>lenguaje reconocido por máquina de turing = -0,05</p> <p>límites de los diferentes autómatas = -0,05</p> <p>lenguaje reconocido por autómatas = -0,15</p> <p>funcionamiento máquinas de turing = -0,15</p> <p>lenguaje generado por expresión regular = -0,20</p> <p>lenguaje generado por gramática = -0,20</p>
Competencias específicas	<p>conocer autómatas de pila = 0,20</p> <p>conocer tipos y mecanismo de control de autómatas finitos = 0,20</p> <p>conocer forma normal de chomsky = 0,05</p> <p>determinar equivalencia autómatas con gramática = -0,05</p> <p>determinar función máquina de turing = -0,05</p> <p>determinar cardinalidad de un conjunto = -0,05</p> <p>determinar si un lenguaje es regular = -0,10</p> <p>conocer elementos y mecanismo de máquinas de turing = -0,20</p> <p>determinar equivalencia entre expresiones regulares = -0,20</p> <p>derivar producciones con gramáticas = -0,20</p>
Conceptos	<p>no deterministas = 0,35</p> <p>poder de reconocimiento = 0,25</p> <p>cardinalidad = 0,25</p> <p>autómatas finitos = 0,20</p> <p>autómatas de pila = 0,20</p> <p>deterministas = 0,10</p> <p>pila = 0,10</p> <p>lenguajes regulares = 0,05</p> <p>lenguaje estructurado por frases = 0,05</p> <p>transiciones = -0,05</p> <p>decidible = -0,05</p> <p>equivalentes = -0,15</p> <p>expresiones regulares = -0,20</p> <p>estrella de kleene = -0,20</p> <p>máquinas de turing = -0,20</p> <p>gramáticas regulares = -0,30</p> <p>derivación = -0,35</p> <p>cadenas = -0,45</p>

Tabla 7: Perfil del alumno tras responder a las preguntas de la Tabla 6. Sólo se muestran los conceptos y competencias con valores distintos de cero.

6. Metodología general

Se describe a continuación una metodología general para aplicar las técnicas descritas en este trabajo en el contexto de otra asignatura diferente, así como algunas recomendaciones para obtener unos buenos resultados. Recordemos que el objetivo es doble: por un lado, crear mapas conceptuales que representen relaciones de dependencia entre clases (de una clasificación) y, por otro, utilizarlos en un sistema recomendador dentro de un entorno educativo.

Clasificación manual de preguntas

La primera fase consiste en que un experto en el dominio de la materia bajo estudio, realice una clasificación de una serie de preguntas tipo test, correspondientes a exámenes de dicha materia. Las clasificaciones posibles son muchas, aunque algunas tienen especial sentido en el contexto en que nos encontramos, por ejemplo, las competencias que se espera que adquiera el alumno de una asignatura son una opción muy recomendable.

Es importante tener en cuenta que las clasificaciones que se realicen tendrán aplicación directa en los dos objetivos de esta metodología (i.e. mapas conceptuales y sistemas recomendadores), por lo que se han de definir los objetivos previamente a realizar las clasificaciones.

En general, las mejores clasificaciones serán aquellas que den lugar a suficientes clases como para que aparezcan en todos los exámenes (reduciendo así valores ausentes), pero sean lo suficientemente específicas como para dar lugar a conocimiento de interés en el dominio bajo estudio. Para ello, los exámenes deberían estar equilibrados en cuanto a las competencias que evalúan, cosa que no suele suceder, pues en general existe una tendencia a evaluar unas sobre otras.

Generación semi-automática de mapas conceptuales

En caso de disponer de datos de resultados de exámenes tipo test, se puede utilizar el algoritmo de generación semi-automática de mapas conceptuales, descrito en este trabajo, con cada una de las clasificaciones utilizadas para clasificar las preguntas en la etapa anterior. El resultado será un mapa conceptual por clasificación, que podrá servir para orientar al profesor en las dependencias existentes en la materia, una vez éste haya analizado las relaciones del mapa y asignado frases de enlace adecuadas.

A la vista de los resultados de los experimentos realizados en la sección 4.2, se recomienda configurar el algoritmo para que utilice datos discretos binarios con un umbral del 100 %, el algoritmo de aprendizaje automático MMHC, el filtrado de fuentes de ruido, y, en caso de realizar imputación de valores ausentes, el algoritmo kNN. A la hora de elegir cuándo realizar la fusión de redes o la imputación de valores ausentes, se recomienda realizar ambas (para obtener más información), salvo que haya un alto porcentaje de valores ausentes, en torno al 40 %, donde la imputación no está recomendada.

Los mapas conceptuales generados en esta fase, pueden servir como entrada para para el sistema recomendador de la siguiente fase, aunque también se pueden utilizar de forma independiente.

Introducción de datos en el sistema recomendador

Se recomienda introducir las preguntas y mapas de conocimiento que hayan sido utilizados en las etapas anteriores, y, además, cuantas preguntas se crea oportuno. Las preguntas deberán tener asignadas la correspondiente clasificación por cada mapa de conocimiento.

Por supuesto, sería posible introducir un mapa de conocimiento que no hubiera sido generado con el algoritmo descrito en este trabajo, siempre que las preguntas estén clasificadas con los conceptos del mismo.

7. Conclusiones

7.1. Principales aportaciones

Se ha presentado un proceso novedoso para generar de forma semi-automática mapas conceptuales mediante aprendizaje automático de redes bayesianas a partir de resultados de exámenes tipo test, prestando especial atención al problema de agregar resultados de diferentes exámenes. Los mapas conceptuales generados tienen una doble utilidad: por un lado pueden permitir a un experto encontrar nuevos enlaces cruzados en sus propios mapas conceptuales creados manualmente, y, por otro, permiten derivar sistemas recomendadores que aprovechen las correlaciones representadas en los mapas conceptuales generados para mejorar el modelo del conocimiento de los alumnos.

Además, se ha presentado un diseño de sistema recomendador genérico que aprovecha las clasificaciones realizadas para generar los mapas conceptuales, así como los propios mapas conceptuales generados de forma semi-automática, para actualizar el perfil de los alumnos y poder recomendar el estudio de los ítems de conceptos y la realización de los ítems de preguntas más adecuados para el conocimiento actual del alumno. Las pruebas realizadas indican que los ítems recomendados pueden resultar adecuados en el proceso de aprendizaje de un alumno.

Este sistema recomendador podría ser una herramienta muy útil para los profesores, ya que les permitiría construir un sistema introspectivo ad-hoc para la materia que imparten de forma sencilla, sin más que proporcionar al sistema cuestionarios de examen clasificados y crear los mapas conceptuales asociados a estas clasificaciones.

7.2. Problemas encontrados

Los resultados obtenidos en nuestros experimentos de generación semi-automática de mapas conceptuales a partir de redes bayesianas, no han permitido determinar conclusivamente la hipótesis del trabajo, es decir, que los resultados de cuestionarios de exámenes llevan implícita una gran cantidad de información sobre el dominio de conocimiento de una materia y su aprendizaje. Esto se podría deber a:

- Las características específicas del dominio utilizado para la experimentación. En Teoría de Autómatas, las competencias están muy relacionadas entre sí, por lo que aparecerán muchas correlaciones, pero sin que unas se muestren más relevantes que otras, aparte de las que un experto podría haber determinado a priori.
- Los alumnos de la UNED son por lo general alumnos de edad madura y perfil muy diverso, de modo que sus competencias, y el modo en que se relacionan, dependen de la idiosincrasia particular de cada uno (experiencia laboral, formación académica previa, contexto cultural de origen...).
- La existencia de más fuentes de ruido en los datos que las tratadas en este trabajo, como el hecho que las preguntas pueden tener respuestas trampa, distinta calidad en el material docente de distintas partes de la materia, tipos de preguntas recurrentes en exámenes pasados que permiten utilizar heurísticas no relacionadas con el dominio o que las preguntas de un mismo nivel de dificultad pueden estar más correlacionadas

entre sí. Además, a pesar de que las respuestas erróneas descuentan puntuación en el examen, los alumnos en ocasiones eligen una al azar para aumentar sus posibilidades de aprobado.

- Las pocas reglas existentes a la hora de generar mapas conceptuales hacen de su creación automática un problema complejo. De hecho, incluso cuando se crean de forma manual, los resultados suelen ser distintos entre personas.

7.3. Trabajo futuro

Se describen a continuación algunas líneas de investigación que han quedado abiertas en el trabajo.

Generación semi-automática de mapas conceptuales

- Queda abierta la posibilidad de realizar más experimentos utilizando diferentes algoritmos de aprendizaje estructural de redes bayesianas, así como de otras técnicas de imputación, tipos de atributos, etc. con el fin de determinar si pueden ser más adecuados que los utilizados en este trabajo a la hora de demostrar la hipótesis del mismo.
- Se podría aplicar la metodología propuesta en este trabajo a datos de resultados de cuestionarios de exámenes de otras materias en las que presumiblemente pudieran reportar mejores resultados. Por ejemplo, si las preguntas hubieran sido diseñadas para evaluar las competencias de forma equitativa e independiente unas de otras.
- El algoritmo de fusión de la estructura de varias redes bayesianas en una sola, se podría completar para que el resultado fuera una red bayesiana que incluyera también los parámetros, además de la estructura. Esto permitiría utilizar el resultado como red bayesiana, además de como un grafo dirigido acíclico (que es como se ha utilizado en este trabajo).
- Considerar en la metodología propuesta la existencia de diferentes perfiles de alumnos, con el fin de obtener mapas conceptuales adaptados a la idiosincrasia de cada estudiante.
- Se podrían valorar las posibilidades de integrar los resultados de otras investigaciones relacionados con habilidades cognitivas, ya que en este trabajo nos hemos limitado al conocimiento de conceptos y competencias del dominio, pero las respuestas de los alumnos también están influenciadas por factores más globales, como la memoria de trabajo, el razonamiento inductivo o la velocidad de procesamiento de la información [GL08].

Sistema recomendador

- Explotar adicionalmente las relaciones propedéuticas que proporcionan los mapas conceptuales para sugerir nuevos caminos de aprendizaje, que no necesariamente coincidan con el programa de la asignatura. Por ejemplo, se podría utilizar lo que se conoce como *Ruta de Instrucción Correctiva* (en inglés, “Remedial Instruction Path” [LLL09]), que consiste en recorrer el mapa ascendiendo desde un concepto en cuestión hasta su nodo raíz. Durante dicha ascensión, cuando un nodo tuviera varios padres, sería necesario

elegir por cuál continuar. Una opción sería elegir aquél en el que el alumno tiene un menor conocimiento. La secuencia lógica de aprendizaje que se propondría al alumno sería el camino por el que se ha realizado la ascensión.

- Completar el prototipo de sistema recomendador realizado, con las funcionalidades no implementadas, de forma que se pudiera poner a disposición de profesores para que pudieran evaluar su uso en entornos reales. Sería interesante ajustar los valores de las constantes K_j en función de los resultados obtenidos.
- Una vez madurado el método de generación de mapas conceptuales, se podría integrar en el sistema recomendador genérico un módulo que diera soporte a dicha generación semi-automática de mapas conceptuales a partir de respuestas a cuestionarios de exámenes.
- Si bien los resultados obtenidos en el dominio de la Teoría de Autómatas no son concluyentes, algunos resultados parciales, como los mapas de conceptos y competencias diseñados manualmente, las clasificaciones de las preguntas de los exámenes, el prototipo de sistema recomendador, etc. podrían servir de base para el desarrollo de un sistema recomendador introspectivo para la materia "Teoría de Autómatas", inédito (hasta donde sabemos) en la literatura de sistemas recomendadores.

Referencias

- [AN11] A. Aluri and M. Natu. Determining causal relationships amongst data using bayesian networks. *Proceedings of 18th International Conference on High Performance Computing (HiPC 2011), Bangalore, India*, December 2011.
- [BC08] S. Bai and S. Chen. Automatically constructing concept maps based on fuzzy rules for adapting learning systems. *Expert systems with Applications*, 35(1):41–49, 2008.
- [Bro93] J. G. Brookshear. *Teoría de la Computación*. Addison-Wesley Iberoamericana, 1993.
- [CDS10] F. Colace and M. De Santo. Ontology for e-learning: a bayesian approach. *Education, IEEE Transactions on*, 53(2):223–233, 2010.
- [CDSV04] F. Colace, M. De Santo, and M. Vento. An automatic algorithm for building ontologies from data. In *Information and Communication Technologies: From Theory to Applications*, pages 117–118. IEEE, 2004.
- [CFU03] D.T. Croasdell, L.A. Freeman, and A. Urbaczewski. Concept maps for teaching and assessment. *Communications of the Association for Information Systems*, 12:396–405, 2003.
- [CH92] G. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.
- [CJFM09] L. Candillier, K. Jack, F. Fessant, and F. Meyer. State-of-the-art recommender systems. *Collaborative and Social Information Retrieval and Access-Techniques for Improved User Modeling*, pages 1–22, 2009.
- [CLW01] P. H. Chung, T. W. Lin, and Y. L. Wu. Conceptual graph for assisted learning on the web. *The 5th global chinese conference on computers in education (GCCCE) and the 10th international conference on computer-assisted instruction (ICCAI)*, 2001.
- [Dí08] F. J. Díez. *Introducción a los Modelos Gráficos Probabilistas*. Dpto. Inteligencia Artificial, UNED, 2008.
- [Eat08] J. W. Eaton. *GNU Octave Manual Version 3*. Network Theory Ltd., Bristol UK, 2008.
- [FGG97] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3):131–163, 1997.
- [GG08] V. Graudina and J. Grundspenkis. Concept map generation from OWL ontologies. In *Proc. of the Third Int. Conference on Concept Mapping*. 2008.
- [GL08] S. Graf and T. Lin. The relationship between learning styles and cognitive traits—getting additional information for improving student modelling. *Computers in Human Behavior*, 24(2):122–137, 2008.

- [Har13] FE Jr. Harrell. Hmisc: Harrell miscellaneous, 2013.
- [Hec08] D. Heckerman. *A tutorial on learning with Bayesian networks*. Springer, 2008.
- [Hop01] J. E. Hopcroft. *Introduction to Automata Theory, Languages, and Computation*. Pearson Education, 2 edition, 2001.
- [IG96] R. Ihaka and R. Gentleman. R: A language for data analysis and graphics. *Journal of computational and graphical statistics*, 5(3):299–314, 1996.
- [JY11] D. R. Johnson and R. Young. Toward best practices in analyzing datasets with missing data: Comparisons and recommendations. *Journal of Marriage and Family*, 73(5):926–945, 2011.
- [KCB10] J. Kowata, D. Cury, and M. C. Boeres. A review of semi-automatic approaches to build concept maps. In *Proc. 4th International Conference on Concept Mapping*, pages 40–48. 2010.
- [KN11] K. Korb and A.E. Nicholson. *Bayesian artificial intelligence*, volume 1. cRc Press, 2nd edition, 2011.
- [KN12] T. Koski and J. Noble. A review of bayesian networks and structure learning. 40(1):51–103, 2012.
- [Lar02] Harold J. Larson. Teoría de conjuntos. In *Introducción a la Teoría de Probabilidades e Inferencia Estadística*. Limusa, 2002.
- [LH08] J-H Lin and P. J. Huag. Exploiting missing clinical data in bayesian network modeling for predicting medical problems. *Journal of biomedical informatics*, 41(1):1–14, 2008.
- [LLL09] C-H Lee, G-G Lee, and Y. Leu. Application of automatically constructed concept map of learning to conceptual diagnosis of e-learning. *Expert Systems with Applications*, 36(2):1675–1684, 2009.
- [LR02] R. Little and D. B. Rubin. *Statistical analysis with missing data*. Wiley, 2002.
- [Mar03] D. Margaritis. Learning bayesian network model structure from data, 2003.
- [MDV⁺11] N. Manouselis, H Drachsler, R. Vuorikari, H. Hummel, and R. Koper. Recommender systems in technology enhanced learning. In *Recommender systems handbook*, pages 387–415. Springer, 2011.
- [NC08] J. D. Novak and A. J. Cañas. The theory underlying concept maps and how to construct and use them. *Technical Report IHMC CMaptTools 2006-01 Rev 2008-01, Florida Institute for Human and Machine Cognition*, 2008.
- [NDS07] A. Nägel, M. Dejori, and M. Stetter. Bayesian substructure learning-approximate learning of very large network structures. In *Machine Learning: ECML 2007*, pages 238–249. Springer, 2007.

- [NG84] J. D. Novak and D. B. Gowin. *Learning how to learn*. Cambridge University Press, 1984.
- [Par08] A. P. Parkes. *A Concise Introduction to Languages and Machines*. Undergraduate Topics in Computer Science. Springer, 2008.
- [Pea88] J. Pearl. *Probabilistic reasoning in intelligent system*. Morgan Kaufmann, San Mateo, CA, 1988.
- [Pea00] J. Pearl. *Causality, models, reasoning, and inference*. Cambridge University Press, 2000.
- [Pig01] T. D. Pigott. A review of methods for missing data. *Educational research and evaluation*, 7(4):353–383, 2001.
- [RN10] S. Russell and P. Norvig. Chapter 14. probabilistic reasoning. In *Artificial Intelligence: a modern approach*. 3rd edition, 2010.
- [SB82] D. Sleeman and J. S. Brown. *Intelligent tutoring systems*. 1982.
- [Sch78] G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.
- [Scu09] M. Scutari. Learning bayesian networks with the bnlearn r package. *arXiv preprint arXiv:0908.3817*, 2009.
- [SGS93] P. Spirtes, C. Glymour, and R. Scheines. Causation, prediction and search. *Lecture Notes in Statistics*, 81, 1993.
- [SJ07] A. Sears and J.A. Jacko. *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*. CRC Press, 2007.
- [TAKP13] M. Templ, A. Alfons, A. Kowarik, and B. Pranter. VIM: visualization and imputation of missing values, 2013.
- [TBA06] I. Tsamardinos, L. Brown, and F. Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006.
- [TKF11] M. Templ, A. Kowarik, and P. Filzmoser. Iterative stepwise regression imputation using standard and robust methods. *Computational Statistics & Data Analysis*, 55(10):2793–2806, 2011.
- [VB12] S. Van Buuren. *Flexible Imputation of Missing Data*. Chapman & Hall/CRC, 2012.
- [VLC08] A. Valerio, D. Leake, and J.C. Cañas. Automatic classification of concept maps based on a topological taxonomy and its application to studying features of human-built maps. In *Proc. of the Third Int. Conference on Concept Mapping*. 2008.

- [VRF03] G. Van Rossum and L. Fred. *Python language reference manual*. Network Theory Ltd., 2003.
- [Wik13] Wikipedia. *Pearson product-moment correlation coefficient* — *Wikipedia, The Free Encyclopedia*. 2013. [Online; accessed 21-April-2013].
- [Zah09] A. Zahariev. Google app engine. *Helsinki University of Technology*, 2009.

Anexos

A. Datos estadísticos de las preguntas de los exámenes

En las tablas 8, 9 y 10, se muestran los resultados obtenidos tras un análisis estadístico básico de los exámenes de “Teoría de Autómatas I” a nivel de pregunta.

Se puede observar que el examen 2004F2 tiene una pregunta acertada por todos los alumnos (o que se les dió por acertada), con lo que no tendrá sentido utilizarla en nuestro estudio. Por otro lado, un 7% de las preguntas se encuentran en los extremos, es decir, son muy difíciles o muy fáciles, suponiendo los intervalos $[0, 0,1]$ y $[0,9, 1]$, respectivamente.

B. Correlaciones entre las preguntas de los exámenes

Suponiendo que cada pregunta es una variable aleatoria que podrá tomar el valor de acierto o fallo, según el alumno haya respondido correctamente o no, podremos calcular el coeficiente de correlación de Pearson [Wik13] entre cada par de variables aleatorias, de la siguiente forma:

$$\rho(X, Y) = \frac{Cov(X, Y)}{\sigma_X \sigma_Y}$$

siendo $Cov(X, Y)$ la covarianza entre X e Y, y $\sigma_X \sigma_Y$ el producto de las desviaciones típicas de X e Y. El valor ρ es una medida de la dependencia lineal entre las variables X e Y.

Para calcular dichas correlaciones, se ha utilizado la función `corrcoef`⁶ de Octave que, además de la matriz de correlaciones R, devuelve una matriz P de p-valores que permiten verificar la hipótesis de no correlación. Se han calculado las correlaciones de los nueve exámenes bajo estudio, con lo que se han obtenido las correspondientes nueve matrices de correlación.

Las órdenes ejecutadas en Octave para obtener los pares de preguntas correlacionados son las siguientes:

```
↳ x
x =
0 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0
0 1 1 0 1 0 1 1 0 1 0 1 1 1 0 0
...}
↳ [r,p]=corrcoef(x);
↳ [i,j]=find(p < 0.05);
```

siendo (i, j) las coordenadas de los elementos de p que tienen un nivel significancia superior al 95%. Estos elementos indican que el valor de correlación obtenido es significativo, lo que no quiere decir que los elementos de R correspondientes a estos elementos de P, indiquen una correlación mayor o menor.

Los valores de los elementos de R están entre -1 y 1 , indicando una correlación positiva cuando son mayores que cero y negativa cuando son menores. En la Tabla 11, se muestran los pares de preguntas correlacionados para valores de R significativos y superiores a $0,3$ o

⁶CORRCOEFF: Más información en <http://octave.sourceforge.net/nan/function/corrcoef.html>

inferiores a $-0,3$ (podría elegirse uno más cercano a cero y obtener más preguntas, o menos cercano y obtener menos), junto con el valor de R correspondiente.

C. Software para redes bayesianas

Comparativa de los paquetes de software evaluados

Se ha realizado una comparativa de diversas herramientas software, con el fin de determinar cuál se adecuaba mejor al objetivo del trabajo y a los algoritmos y técnicas planteados. En la tabla 12 se muestran los puntos más relevantes de cada uno de los programas valorados.

Tras evaluar los paquetes de software de la tabla anterior, se ha decidido utilizar R con los módulos comentados (i.e. bnlearn, VIM y Hmisc), ya que nos permite aprender redes para descubrir conocimiento, es posible automatizar su ejecución (mediante el lenguaje R), implementa los algoritmos y métricas elegidos, y, al existir una gran cantidad de paquetes complementarios, nos da mucha flexibilidad para resolver otros problemas, como el de la imputación de valores ausentes. El artículo [Scu09] sirve de tutorial para el uso de bnlearn en el aprendizaje de redes bayesianas.

Tratamiento de valores ausentes con R

Para realizar la imputación de valores ausentes se han elegido algoritmos que permitan trabajar con datos nominales, ya que nuestros datos son de este tipo (acierto o fallo, sí o no), pero también con datos continuos, con los que también se ha realizado algún experimento. A continuación, se muestra cómo cargar un conjunto de datos y eliminar las filas con valores ausentes:

```
data <- read.csv(filename)
data <- data[complete.cases(data),]
```

Para realizar la imputación unidimensional mediante la mediana, se hace uso del paquete Hmisc [Har13] y, concretamente, de su función “impute” (que utiliza por defecto la mediana), como se muestra a continuación:

```
library('Hmisc')
data <- read.csv(filename)
for column in names(data) {
  data[[column]] <- impute(data[[column]])
}
```

Por último, se muestra cómo realizar la imputación multidimensional mediante el método kNN (k-Nearest Neighbors), disponible en el paquete VIM [TAKP13]:

```
library('VIM')
data <- read.csv(filename)
imputed_data <- kNN(data)
data <- imputed_data[, 1:length(names(data))]
```

Aprendizaje estructural de redes bayesianas con R

Una vez se dispone de los datos sin valores ausentes, se puede proceder a realizar el aprendizaje automático de redes bayesianas mediante los algoritmos elegidos en este trabajo. Para ello, se utiliza el paquete `bnlearn` [Scu09] de la siguiente forma:

```
library('bnlearn')
bn1 <- hc(data)
bn2 <- mmhc(data)
```

Se muestra a continuación un ejemplo de contenido de una red aprendida usando el algoritmo Max-Min Hill Climbing (`mmhc`):

```
Bayesian network learned via Hybrid methods
model:
[conocer.forma.normal.de.chomsky][conocer.propiedades.lenguajes.formales]
[conocer.tipos.y.mecanismo.de.control.de.automatas.finitos]
[determinar.decidibilidad.de.un.lenguaje]
[determinar.lenguaje.asociado.a.automata.de.pila]
[determinar.si.un.lenguaje.es.estructurado.por.frases]
[determinar.si.un.lenguaje.es.independiente.del.contexto]
[determinar.tipo.de.lenguaje]
[conocer.tesis.de.turing|conocer.forma.normal.de.chomsky]
[determinar.equivalencia.entre.expresiones.regulares|
conocer.propiedades.lenguajes.formales]
[determinar.funcion.maquina.de.turing|
determinar.decidibilidad.de.un.lenguaje]
[determinar.si.un.lenguaje.es.regular|
conocer.propiedades.lenguajes.formales]
[conocer.elementos.y.mecanismo.de.maquinas.de.turing|
conocer.tesis.de.turing]
[determinar.cardinalidad.de.un.conjunto|
determinar.si.un.lenguaje.es.regular]
[determinar.equivalencia.automata.con.gramatica|
determinar.si.un.lenguaje.es.regular]
nodes: 15
arcs: 7
  undirected arcs: 0
  directed arcs: 7
average markov blanket size: 0.93
average neighbourhood size: 0.93
average branching factor: 0.47
learning algorithm: Max-Min Hill-Climbing
constraint-based method: Max-Min Parent Children
conditional independence test: Mutual Information (discrete)
score-based method: Hill-Climbing
score: Bayesian Information Criterion
alpha threshold: 0.05
penalization coefficient: 2.138333
tests used in the learning procedure: 146
```

2004F1						2004S1					
#	A	B	C	Blanco	P(acertar)	#	A	B	C	Blanco	P(acertar)
1	5	108	1	6	0.9000	1	1	4	61	6	0.8472
2	97	0	6	17	0.8083	2	30	11	8	23	0.4167
3	16	1	91	12	0.7583	3	16	21	26	9	0.2917
4	8	62	25	25	0.5167	4	25	16	21	10	0.3472
5	4	5	106	5	0.8833	5	11	54	4	3	0.7500
6	32	54	18	16	0.2667	6	11	1	21	39	0.1528
7	15	1	99	5	0.8250	7	18	6	9	39	0.2500
8	13	88	5	14	0.7333	8	9	49	8	6	0.6806
9	17	13	46	44	0.3833	9	12	10	30	20	0.4167
10	39	25	19	37	0.3250	10	34	16	5	17	0.4722
11	26	11	50	33	0.4167	11	5	0	51	16	0.7083
12	70	15	8	27	0.5833	12	15	30	1	26	0.4167
13	84	5	5	26	0.7000	13	24	6	25	17	0.3472
14	5	75	13	27	0.6250	14	8	2	59	3	0.8194
15	3	1	111	5	0.9250	15	5	52	11	4	0.7222
16	8	71	18	23	0.5917	16	5	9	45	13	0.6250
2004F2											
#	A	B	C	Blanco	P(acertar)						
1	138	4	30	18	0.7263						
2	137	2	21	30	0.7211						
3	11	3	168	8	0.8842						
4	3	24	144	19	0.7579						
5	174	3	12	1	0.9158						
6	17	5	156	12	0.8211						
7	20	107	20	43	0.5632						
8	117	53	6	14	0.6158						
9	113	16	11	50	0.5947						
10	0	190	0	0	1.0000						
11	18	16	81	75	0.4263						
12	11	122	30	27	0.6421						
13	25	13	95	57	0.5000						
14	8	60	81	41	0.4263						
15	109	15	0	66	0.5737						
16	3	1	168	18	0.8842						

Tabla 8: Estadísticas de las preguntas de los exámenes estudiados de 2004.

2005F1						2005S1					
#	A	B	C	Blanco	P(acertar)	#	A	B	C	Blanco	P(acertar)
1	111	0	3	3	0.9487	1	117	13	15	10	0.7548
2	28	4	66	19	0.5641	2	88	20	13	34	0.5677
3	13	5	73	26	0.6239	3	30	6	89	30	0.5742
4	80	7	9	21	0.6838	4	19	85	10	41	0.5484
5	10	84	7	16	0.7179	5	119	6	17	13	0.7677
6	40	9	14	54	0.3419	6	74	14	40	27	0.4774
7	15	68	20	14	0.5812	7	32	39	20	64	0.1290
8	18	30	18	51	0.2564	8	50	63	8	34	0.3226
9	17	6	77	17	0.6581	9	36	13	56	50	0.3613
10	27	21	29	40	0.2308	10	39	46	21	49	0.2968
11	63	13	10	31	0.5385	11	50	36	30	39	0.2323
12	25	20	36	36	0.3077	12	55	39	18	43	0.1161
13	84	11	7	15	0.7179	13	27	96	9	23	0.6194
14	8	43	48	18	0.4103	14	12	101	32	10	0.6516
15	22	65	13	17	0.5556	15	9	36	50	60	0.2323
16	1	104	5	7	0.8889	16	7	126	10	12	0.8129
2005F2											
#	A	B	C	Blanco	P(acertar)						
1	11	13	204	9	0.8608						
2	66	44	90	37	0.3797						
3	6	219	12	0	0.9241						
4	41	42	72	82	0.3038						
5	26	36	129	46	0.5443						
6	68	12	81	76	0.3418						
7	21	5	163	48	0.6878						
8	85	16	66	70	0.2785						
9	34	15	79	109	0.4599						
10	161	15	17	44	0.6793						
11	4	27	154	52	0.6498						
12	113	37	28	59	0.4768						
13	178	38	6	15	0.7511						
14	55	22	41	119	0.1730						
15	33	74	108	22	0.3122						
16	14	33	145	45	0.6118						

Tabla 9: Estadísticas de las preguntas de los exámenes estudiados de 2005.

2006F1						2006S1					
#	A	B	C	Blanco	P(acertar)	#	A	B	C	Blanco	P(acertar)
1	3	3	83	23	0.7411	1	25	58	13	26	0.4754
2	53	31	12	16	0.4732	2	8	5	27	82	0.0656
3	0	5	88	19	0.7857	3	21	33	29	39	0.2705
4	4	15	39	54	0.3482	4	10	52	31	29	0.4262
5	57	9	28	18	0.5089	5	14	34	37	37	0.3033
6	3	29	66	14	0.5893	6	34	71	16	1	0.2787
7	18	26	9	59	0.2321	7	108	7	5	2	0.8852
8	7	10	36	59	0.3214	8	46	16	5	55	0.3770
9	11	7	86	8	0.7679	9	15	15	38	54	0.1230
10	107	0	2	3	0.9554	10	79	7	4	32	0.6475
11	5	7	86	14	0.7679	11	7	0	99	16	0.8115
12	104	4	2	2	0.9286	12	13	81	12	16	0.6639
13	4	93	7	8	0.8304	13	8	4	90	20	0.7377
14	16	58	6	32	0.5179	14	18	7	83	14	0.6803
15	80	9	4	19	0.7143	15	8	21	54	39	0.4426
16	5	99	2	6	0.8839	16	10	83	13	16	0.6803
2006F2											
#	A	B	C	Blanco	P(acertar)						
1	35	43	159	53	0.5483						
2	45	16	104	125	0.3586						
3	36	46	101	107	0.3483						
4	15	185	9	81	0.6379						
5	43	152	9	86	0.5241						
6	187	19	39	45	0.6448						
7	44	113	57	76	0.3897						
8	0	289	1	0	0.9966						
9	16	12	259	3	0.8931						
10	69	16	150	55	0.5172						
11	201	18	20	51	0.6931						
12	4	32	204	50	0.7034						
13	166	45	25	54	0.5724						
14	209	43	18	20	0.7207						
15	59	117	95	19	0.4034						
16	19	27	191	53	0.6586						

Tabla 10: Estadísticas de las preguntas de los exámenes estudiados de 2006.

2004F1			2004F2			2004S1		
1	5	0,31	1	2	0,38	1	14	0,30
1	15	0,33				2	4	0,33
2	3	0,37				3	4	0,30
3	15	0,36				3	15	0,33
5	7	0,31				4	9	0,45
5	14	0,31				5	14	0,40
10	14	0,35				5	15	0,57
14	16	0,30				8	15	0,31
						9	16	0,31
						10	12	0,39
			14	15	0,43			
2005F1			2005F2			2005S1		
2	5	0,41				11	12	0,33
2	9	0,35				13	14	0,32
3	15	0,44						
4	10	0,33						
5	9	0,35						
7	16	0,36						
2006F1			2006F2			2006S1		
10	16	0,33	4	6	0,31	1	8	0,34
11	13	0,31				8	10	0,33
11	15	0,40				10	11	0,39
11	16	0,33				11	12	0,37
13	16	0,36						

Tabla 11: Pares de preguntas de los exámenes bajo estudio, con un valor de correlación significativo y superior a 0,30.

Software	Algoritmos híbridos y de búsqueda heurística	Métricas búsqueda heurística	Descubrir conocimiento	Automatización	Imputación de valores ausentes
Elvira	K2, K2SN	BIC, BDe, K2	Sí	No	Rellenar con ceros, media, media condicionada a una clase, árbol de clasificación con ID3 o C4.5 o Dirichlet, MPE iterativo e incremental
OpenMarkov	Greedy Hill Climbing	K2, Bayesian, BDe, MDL, Entropy, AIC	Sí	Sí (Java)	No
Pebl	Greedy Hill Climbing, Simulated Annealing	Bayesian	Sí	Sí (Python)	Muestreo de Gibbs
R (bnlearn, VIM y Hmisc)	Greedy Hill Climbing, Tabu Search, Max-Min Hill Climbing, Two-Phase Restricted Maximization	BIC, BDe, AIC, K2, Entropy	Sí	Sí (R)	Estadísticos unidimensionales como la media, la mediana o la moda (Hmisc), kNN (VIM) y muchos otros
Weka	Genetic Search, Tabu, Greedy & LAGD Hill Climbing, K2	Bayesian, MDL, BDe, AIC, Entropy	No (sólo clasificación)	No	Media y mediana

Tabla 12: Características relevantes para este trabajo de diferentes paquetes de software de aprendizaje de redes bayesianas.

D. Mapas conceptuales generados de forma semi-automática

En las figuras 19 y 20, se muestran los mapas conceptuales generados a partir de dos redes bayesianas aprendidas de forma automática a partir de los datos de competencias generales de la asignatura de Teoría de Autómatas I (sección 4.2.3); en concreto, se utilizan las redes de las figuras 9 y 10, por ser las mayor relevancia del experimento.

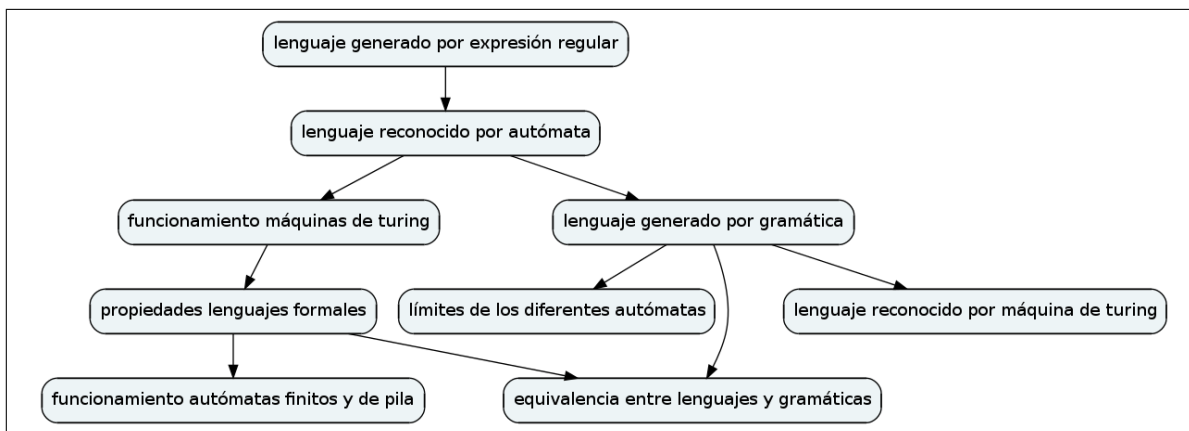


Figura 19: Mapa conceptual correspondiente a los datos de competencias generales de la asignatura de Teoría de Autómatas I, obtenida mediante el algoritmo de transformación de redes bayesianas a mapas conceptuales, sin imputación de valores ausentes, diseñado en este trabajo. Se utilizó como red original la de la figura9.

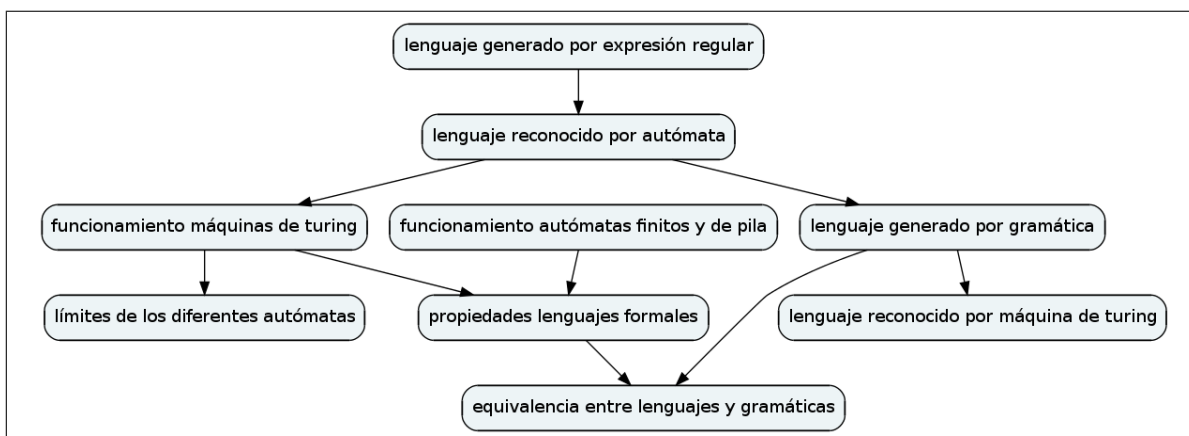


Figura 20: Mapa conceptual correspondiente a los datos de competencias generales de la asignatura de Teoría de Autómatas I, obtenida mediante el algoritmo de transformación de redes bayesianas a mapas conceptuales, con imputación de valores ausentes, diseñado en este trabajo. Se utilizó como red original la de la figura10.

En la figura 21, se muestra el mapa conceptual de competencias específicas aprendido a partir de los experimentos de la sección 4.2.4, donde se aplicó el algoritmo de fusión de estructuras de redes bayesianas diseñado en este trabajo.

Finalmente, en las figuras 22, 23 y 24, se muestran los mapas conceptuales aprendidos a partir de las redes bayesianas de conceptos de los temas 1, 2 y 3 de la asignatura de Teoría de Autómatas I (sección 4.2.5).

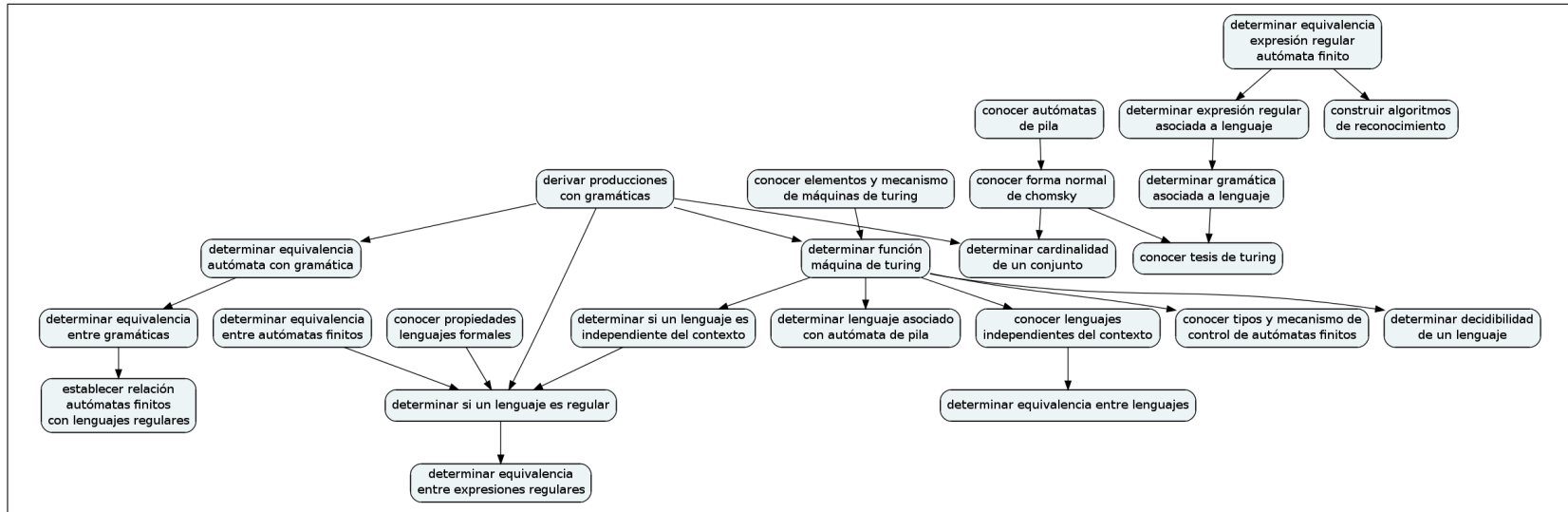


Figura 21: Mapa conceptual correspondiente a los datos de competencias específicas de la asignatura de Teoría de Autómatas I, obtenida mediante el algoritmo de transformación de redes bayesianas a mapas conceptuales, con imputación de valores ausentes, diseñado en este trabajo.

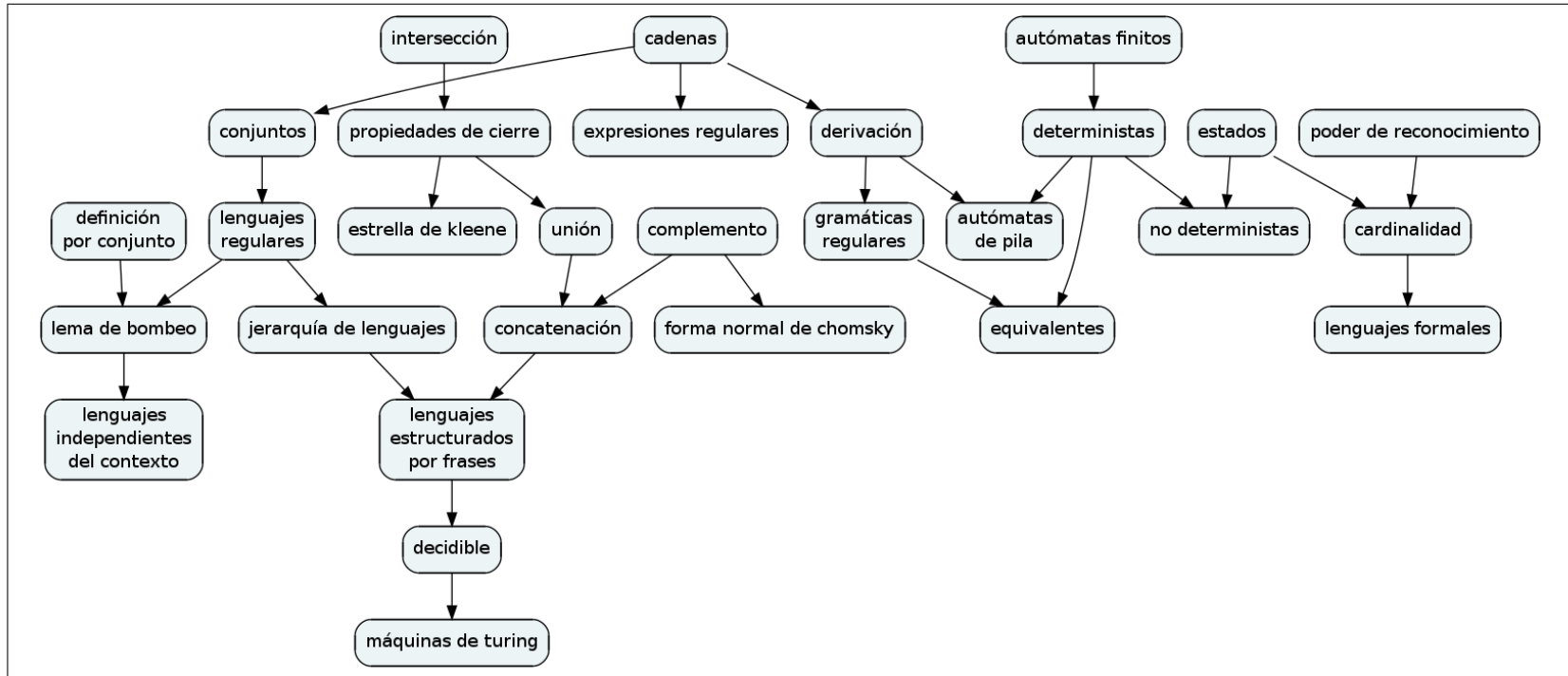


Figura 22: Mapa conceptual correspondiente a los datos de conceptos del tema 1 de la asignatura de Teoría de Autómatas I, obtenida mediante el algoritmo de transformación de redes bayesianas a mapas conceptuales, con imputación de valores ausentes, diseñado en este trabajo.

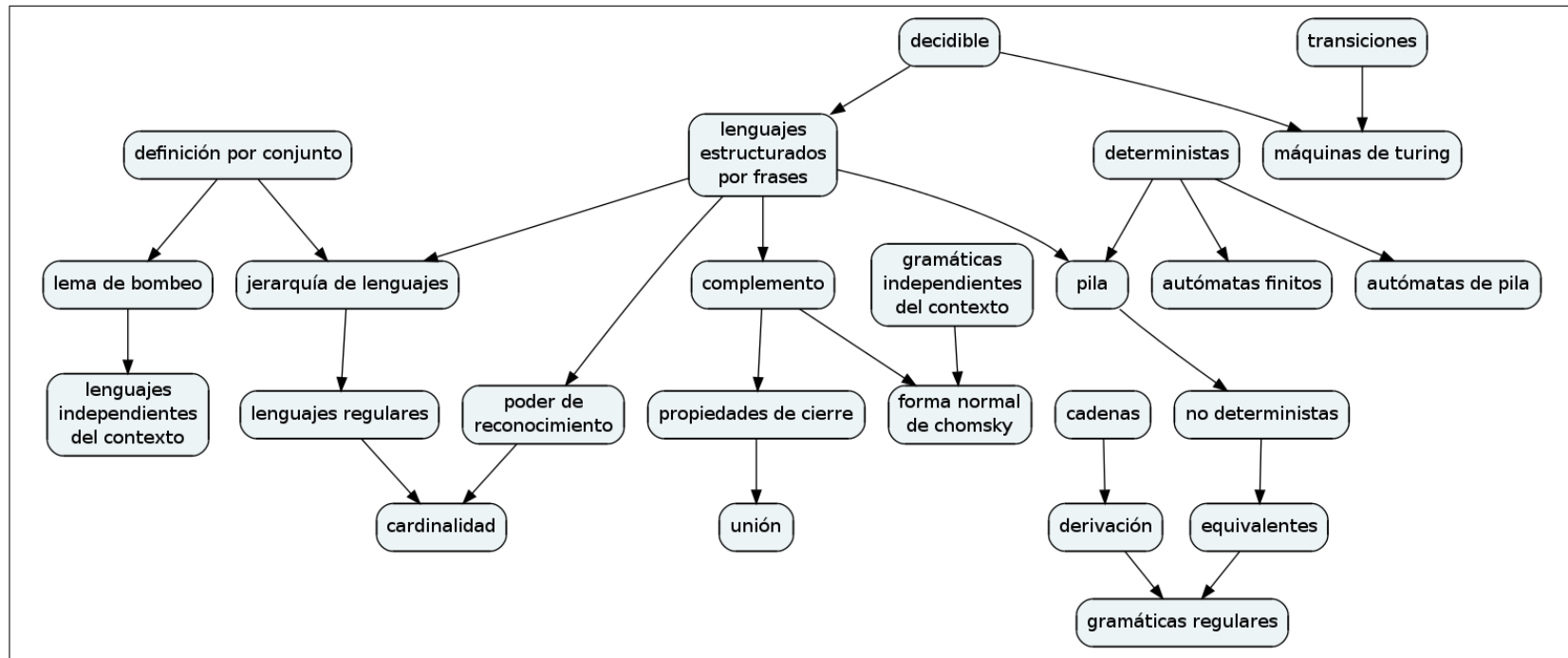


Figura 23: Mapa conceptual correspondiente a los datos de conceptos del tema 2 de la asignatura de Teoría de Autómatas I, obtenida mediante el algoritmo de transformación de redes bayesianas a mapas conceptuales, con imputación de valores ausentes, diseñado en este trabajo.

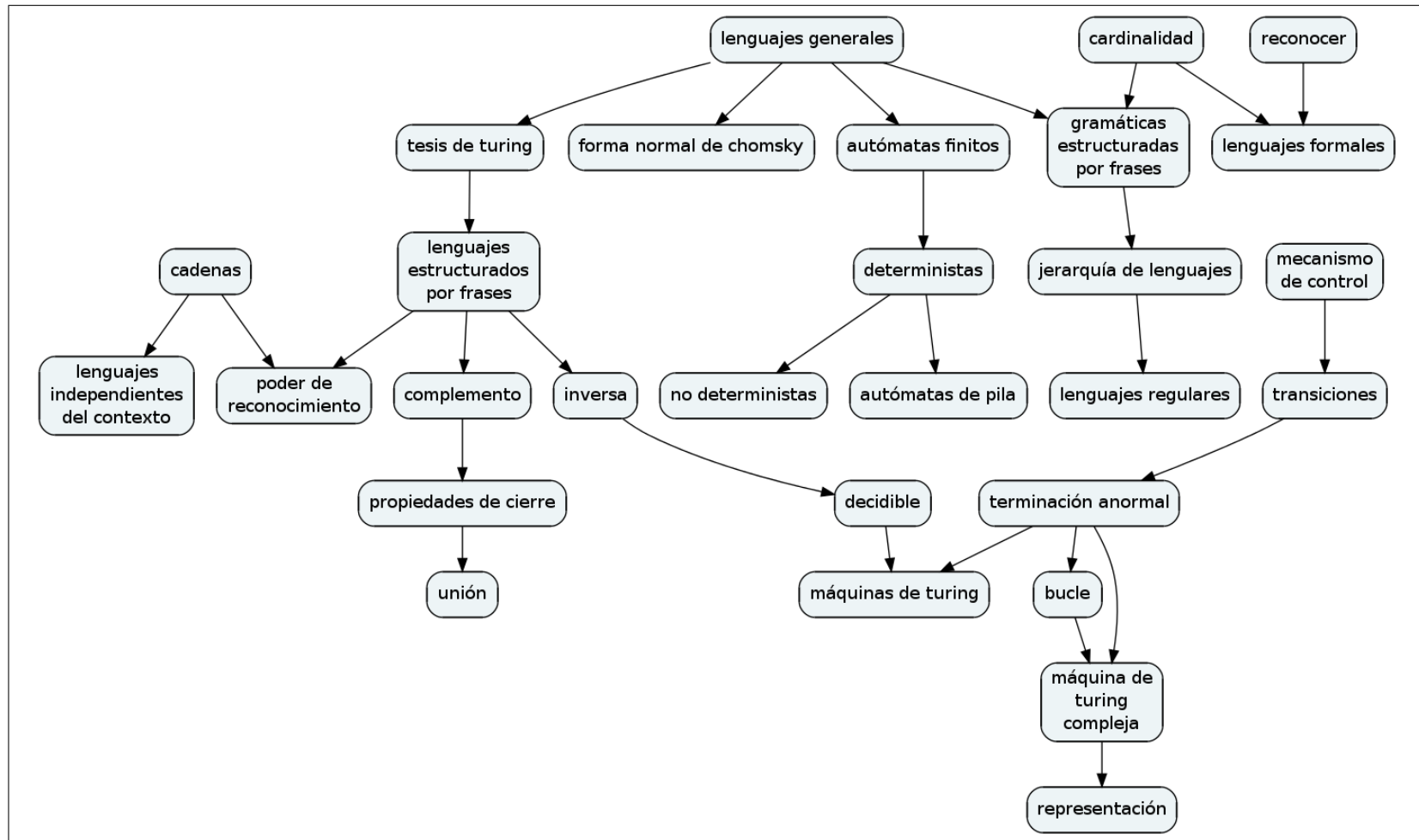


Figura 24: Mapa conceptual correspondiente a los datos de conceptos del tema 3 de la asignatura de Teoría de Autómatas I, obtenida mediante el algoritmo de transformación de redes bayesianas a mapas conceptuales, con fusión de redes, diseñado en este trabajo.