Universidad de Granada
Departamento de Ciencias de la Computación e Inteligencia Artificial

Tesis doctoral

Doctorado en Tecnologías de la Información y Comunicación

MODELOS GRÁFICOS PROBABILÍSTICOS APLICADOS A LA PREDICCIÓN DEL RENDIMIENTO EN EDUCACIÓN

Byron Wladimir Oviedo Bayas

Directores:
Prof. Dr. Serafín Moral Callejón

Granada, 2016

Editor: Universidad de Granada. Tesis Doctorales
Autor: Byron Wladimir Oviedo Bayas
ISBN: 978-84-9163-070-8
URI: http://hdl.handle.net/10481/44592

Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.

— Albert Einstein

A Dios todopoderoso, quien supo guiarme y darme las fuerzas necesarias para seguir adelante y no desmayar cuando se presentaban inconvenientes en el camino de esta investigación.

A mis padres Gonzalo y Fanny por ser los pilares fundamentales de todo lo que soy, por todo el apoyo, consejos, comprensión y amor brindado. Por ser la guía y el ejemplo a seguir como persona, por enseñarme valores, principios y perseverancia, para conseguir mis objetivos.

A mi bella esposa Lilo por ser mi compañera durante muchos años, y por haberme regalado dos bellos hijos, para criarlos, cuidarlos y amarlos.

Para mis hijos: Joan, Max y Gonzalo, mi razón de ser, mi más grande éxito logrado en la vida y uno de los motivos por lo que siempre vivo agradecido de Dios.

A mis hermanos Matilde, Ligia, Omar y Mayra (+), y a todos mis queridos sobrinos...

A ellos les dedico mi tesis, pues ellos son quienes me dieron su apoyo incondicional.

LOS AMO

Agradecimientos

Agradezco a varias personas e instituciones por toda la ayuda brindada, por su empuje, sus palabras y actos durante el desarrollo de este trabajo de investigación.

- A Dios Todopoderoso que me ha dado la vida y una bella familia.
- Al Dr. Serafín Moral Callejón, mi tutor, por el esfuerzo, por la dedicación, la motivación y orientaciones que me ha brindado como amigo y profesional que han permitido ganarse mi admiración.
- Al PhD. Amilkar Puris Cáceres, gran amigo, por su decisivo apoyo, por las sugerencias y comentarios sobre la elaboración de este trabajo.
- En las circunstancias más difíciles de esta etapa tuve la suerte de contar con amigos muy valiosos que solidariamente se acercaron a ratificarme su amistad y brindarme apoyo para salir de cualquier estado en que me encontraba. Hoy no tengo más que gestos y palabras de gratitud muy especiales a mis compañeros del CITIC: Virgilio, Jorge, Karel, Gonzalo, Andrés, Raúl, Francisco, por el aliento, motivación y sobretodo por crear ese ambiente de camaradería necesario para que las estancias en Granada sean más agradables y provechosas.
- A La Universidad Técnica Estatal de Quevedo - Facultad de Ciencias de la Ingeniería, por brindarme la oportunidad de formar parte de este programa doctoral, y apoyarme durante todo este tiempo de estudios.
- A la Asociación Universitaria Iberoamericana de Posgrado (AUIP), por la beca concedida y a su Director Regional para Andalucía Oriental por el apoyo brindado.

En general, a todas aquellas personas que de una u otra forma me han apoyado y criticado. A todos muchas gracias.

Índice general

Índice de tablas

Tabla

Índice de figuras

Figura

Resumen

La deserción estudiantil en las universidades es un problema que debe ser investigado y tratado con prioridad ya que se ha constituido como un indicador de eficiencia dentro de las instituciones de educación superior.

Lo expuesto anteriormente nos lleva a proponer métodos que nos ayude a identificar a tiempo los estudiantes con riesgo de deserción en la Facultad de Ciencias de la Ingeniería de la Universidad Técnica Estatal de Quevedo y de esa manera aplicar alguna metodología para evitar que se lleve a efecto la deserción.

Para realizar esta investigación se adoptó utilizar modelos gráficos probabilísticos como las redes bayesianas con métodos de clasificación y agrupamiento para poder analizar el comportamiento de los alumnos, tomando en consideración los datos socio-económicos y académicos de los estudiantes legalmente matriculados en el periodo 2012-2013 en la Facultad de Ciencias de la Ingeniería. Así como también otros experimentos con datos académicos de la UCI para dos escuelas en Portugal sobre los cursos de matemáticas y lengua portuguesa. El trabajo de clasificación se lo aplicó con los algoritmos que nos proporciona la herramienta Weka y el agrupamiento con Elvira.

Adicionalmente se plantea un método basado en clasificación supervisada para la construcción de una jerarquía de variables artificiales de todas las variables observadas aplicando un método con clúster jerárquico que mejor se acople a estos datos estudiantiles.

De igual manera se define una nueva forma de aprendizaje estructural para clasificadores bayesianos basados en métodos envolventes teniendo como problema principal encontrar la topología que mejor clasifique los datos (problema de optimización complejo), por lo que se propone aprender clasificadores bayesianos a través de la metaheurística conocida como optimización basada en mallas variables (VMO) para solucionar los problemas de optimización.

Más adelante introducimos un nuevo clasificador que llamaremos clasificador bayesiano simple, que será una red bayesiana genérica, pero aprendida con una técnica voraz. Una vez analizados los datos de clasificación obtenidos con la herramienta Elvira con varios algoritmos se los comparara con el propuesto SBND.

Se puede indicar que el problema de deserción estudiantil que se analiza es complejo y difícil y que ha sido necesario utilizar métodos que usan una combinación de factores (clasificadores

bayesianos) para poder obtener algunas mejoras sobre los clasificadores triviales.

Se propone realizar trabajos futuros relacionando mayor cantidad de variables que influyan en la deserción estudiantil como es el caso de la capacitación docente y la infraestructura tecnológica institucional.

Parte I

Preliminares

1 Introducción

La falta de seguridad, de confianza o de certeza sobre algo, es normalmente lo que sucede cuando tratamos de relacionarnos con el mundo; es decir, no tenemos un conocimiento exacto de lo que está sucediendo y por eso utilizamos el razonamiento aproximado. Se puede dejar más claro lo manifestado con ejemplos sencillos, como el conocer que una cerveza tiene un valor de un dólar, o saber si con la caída del precio del petróleo, en Ecuador el costo de sus derivados va a subir. Otro ejemplo, en el campo de la mecánica cuántica se indica que no se pueden conocer las trayectorias para las partículas ya que no se puede medir al mismo tiempo la posición y la velocidad. Continuamos en el día a día sin tener un conocimiento exacto de lo que está sucediendo en el mundo, pero el razonamiento nos ayuda a desenvolvernos sin problemas.

El razonamiento probabilístico es fundamental para poder tomar decisiones en el área profesional, en el día a día; como también en la investigación. Todos los días nos surgen problemas que necesitan ser razonados probabilísticamente, como es el caso de hacer una estimación de probabilidad de que el día de mañana llueva o no (Serrano et al.1998).

Las redes bayesianas, los árboles de decisión y las redes neuronales artificiales, han sido tres métodos usados en razonamiento automático durante estos últimos años en tareas como la clasificación de documentos o filtros de mensajes de correo electrónico.

(Larrañaga et al.2005) indica que en los últimos años los sistemas expertos probabilísticos han alcanzado un alto grado de desarrollo. Hasta los 80 se consideraba que la probabilidad requería mucha información y que los cálculos eran demasiado complejos para la resolución de problemas reales donde intervengan un gran número de variables. Sin embargo esto fue cambiando gracias a los trabajos de (Pearl1988).

La idea esencial es aprovechar las relaciones tanto de dependencia como de independencia que se encuentran entre las variables de un problema antes de especificar y calcular con los valores numéricos de las probabilidades. Estas relaciones se representan a través de modelos gráficos, habitualmente grafos acíclicos dirigidos (Pearl1988).

Las redes bayesianas se conocen también como redes causales o redes causales probabilísticas, redes de creencia, sistemas probabilísticos o sistemas expertos bayesianos. Las redes bayesianas son modelos estadísticos que representan la incertidumbre a través de las relaciones

de independencia condicional que se establecen entre las variables del problema (Edwards1998). Este tipo de redes codifica la incertidumbre asociada a cada variable por medio de probabilidades. (Neapolitan2004) afirma que una red bayesiana es un conjunto de variables, una estructura gráfica conectada a estas variables y un conjunto de distribuciones de probabilidad. Son una representación compacta de una distribución de probabilidad multivariante. De igual manera, (Castillo, Gutiérrez, y Hadi1997) manifiesta que una red bayesiana es un grafo dirigido acíclico en el que cada uno de sus nodos representa a una variable y que las dependencias probabilísticas se representan mediante arcos; la variable a la que apunta un arco es dependiente de la que está en el origen de éste. La topología o estructura de la red nos da información sobre las dependencias probabilísticas entre las variables pero también sobre las independencias condicionales de una variable (o conjunto de variables) dada otra u otras variables; dichas independencias simplifican la representación del conocimiento (menos parámetros) y el razonamiento (propagación de las probabilidades) (Puris2010). Dado un grafo, para especificar una probabilidad conjunta para todas las variables, es suficiente con dar una distribución condicional para cada nodo dados sus padres.

Una de las razones del éxito de las redes bayesianas es que se pueden aprender a partir de datos; éste es un proceso de aprendizaje que se divide en dos etapas: el aprendizaje estructural y el aprendizaje paramétrico (Neapolitan2004). La primera de ellas consiste en obtener la estructura de la red bayesiana, es decir, las relaciones de dependencia e independencia entre las variables involucradas. La segunda etapa tiene como finalidad obtener las probabilidades condicionales requeridas a partir de una estructura dada.

En este trabajo se propone el uso de modelos gráficos probabilísticos en el campo de la enseñanza para realización del diagnóstico de los estudiantes y poder predecir su comportamiento. La educación está sujeta a la supervisión constante de todos los procesos de enseñanza y aprendizaje que la componen. La monitorización del proceso educacional, es por tanto un método que analiza constantemente cómo avanzan las actividades de enseñanza y aprendizaje en relación con los objetivos propuestos; esto permite garantizar la dirección del proceso hacia una situación deseada, introducir acciones educativas adicionales y obtener la información necesaria y útil para tomar las decisiones que correspondan. Se debe recordar que monitorizar no es sinónimo de evaluar, aunque sin dudas son procesos que tienen muchos puntos en común.

En los datos educacionales es común que se use la técnica de clustering para descubrir la estructura en los datos de entrada buscando agrupamientos entre las variables y grupos de casos de forma que cada grupo sea homogéneo y distinto de los demás. Hay varios métodos de clustering entre los más conocidos tenemos: jerárquicos (los datos se agrupan de manera arborescente), no jerárquicos (genera particiones a un solo nivel), paramétricos (asume que las densidades condicionales de los grupos tienen cierta forma paramétrica conocida y se reduce a estimar los parámetros), no paramétricos (no asumen nada sobre el modo en el que se agrupan los objetos).

En muchas ocasiones es posible que el clustering esté asociado a situaciones en las que no se

hayan determinado algunas variables que eran importantes para la explicación del fenómeno bajo estudio, y es notado que existe un influencia en los datos que no es percibida fácilmente, entonces es posible postular la existencia de alguna o algunas variables ocultas como responsables de explicar esta producción anormal de los datos (Acosta et al.2014).

AutoClass (Cheeseman et al.1993) es una técnica de agrupamiento que usa razonamiento bayesiano dado un conjunto de datos de entrenamiento y sugiere un conjunto de clases posibles asociadas a los valores de una variable oculta. Los agrupamientos naturales obtenidos mediante esta técnica usando similitud resultan muy útiles a la hora de construir clasificadores cuando no están bien definidas las clases. AutoClass es un Naive Bayes con una variable clase oculta y que permite encontrar la hipótesis más probable, proporcionado los datos e información a priori. Normalmente se busca un balance entre lo bien que se ajustan los datos a las clases y la complejidad de las clases (casos extremos, una clase por dato o una sola clase para todos los datos).

En AutoClass los datos se pueden representar por valores discretos, enteros y reales; pero en nuestro caso solo usaremos variables discretas. El modelo es una mezcla finita de distribuciones de probabilidad, cada una con su conjunto de parámetros. Para cada dato se asigna una probabilidad de pertenencia a una clase (o un peso). Se asume que los datos son condicionalmente independientes dada la clase.

En Autoclass se supone que todas las variables participan en la definición de los grupos, pero en muchos casos, dado un conjunto de mediciones éstas se pueden particionar en conjuntos de tal manera que se puede organizar un agrupamiento distinto para cada conjunto, por esto en este trabajo se propone un nuevo método para la clasificación no supervisada basada en la construcción de una jerarquía de variables artificiales que estarán en la parte superior de las variables observadas $\mathbf{X} = \{X_1, \ldots, X_n\}$. La idea básica es similar a Autoclass, pero en lugar de usar todas las variables, primero hacemos una clasificación de las variables buscándolas por grupos con alto grado de dependencia de estas variables. El resultado final será una red bayesiana $\mathscr{B}$, que inicialmente contiene variables $\mathbf{X}$ y arcos de manera que se creará un árbol de variables no observadas en el que hay una variable observada en cada hoja.

Otro tema que se estudia en la tesis es el problema del aprendizaje de una red para clasificación supervisada. Este problema se puede resolver definiendo una métrica y buscando la red que optimice la métrica. La dificultad está en que este problema de optimización es NP-difícil (Aroztegui, Arraga, y Nesmachnow2003). Por lo que se hace necesario usar técnicas de optimización combinatoria. En este trabajo se propone utilizar una metaheurística poblacional conocida como VMO (Variable Mesh Optimization), (Puris et al.2012), para el aprendizaje de redes bayesianas y para evaluar su comportamiento. En este modelo, un conjunto de nodos que representan soluciones potenciales forman una malla inicial, la cual con la aplicación de distintos operadores se expande en la exploración del espacio de búsqueda.

La esencia del método VMO es crear una malla de puntos en un espacio m dimensional, donde se realiza el proceso de optimización de una función objetivo FO. Dicha malla se hace

más fina en aquellas zonas que parecen ser más promisorias. Es variable en el sentido de que la malla cambia su tamaño (cantidad de nodos) y configuración durante el proceso de búsqueda. Cada nodo se codifica por un vector de **m** números flotantes, que representan la solución al problema de optimización (Puris et al.2012).

Nosotros usaremos VMO aplicado al aprendizaje estructural de una red bayesiana como métrica + búsqueda. VMO ha presentado soluciones competitivas en problemas continuos con diferentes características: aproximación de funciones multimodales (Puris et al.2012), problemas de nichos (Molina et al.2013); pero, no se conoce de algún estudio previo sobre la aplicación de esta técnica en problemas de aprendizaje de redes bayesianas.

La idea es representar en cada nodo de una malla una red bayesiana a través de un conjunto de arcos. Las BNs serán creadas mediante operaciones entre conjuntos (unión y diferencia). Para este proceso se identifican tres tipos de redes bayesianas: la óptima local (BN con mejor valor de métrica en cada vecindad), la óptima global (BN con mejor score entre las óptimas locales), y las soluciones frontera (BN más o menos diferentes en la estructura). Finalmente se aplica un proceso de simplificación (clearing) para seleccionar las BN más representativa de la malla (métrica + estructura).

Para obtener la métrica global, cada BN es usada como un clasificador bayesiano para evaluar su comportamiento aplicando validación cruzada sobre el conjunto de datos de entrenamiento.

En esta investigación también se introduce un nuevo clasificador al que llamaremos 'clasificador bayesiano simple', que será una red bayesiana genérica, pero que es aprendida mediante una técnica voraz.

En aplicación a problemas de educación, contamos con una base de conocimientos de 773 estudiantes matriculados en el periodo 2012-2013 en la Facultad de Ciencias de la Ingeniería de la Universidad Técnica Estatal de Quevedo, de los cuales se les ha extraído sus datos socioeconómicos y de rendimiento académico para poder ser usados en clustering y en clasificación. Uno de los métodos que ha utilizado para el proceso de agrupamiento es el clúster jerárquico, el mismo que nos ayuda a evaluar las encuestas dirigidas a los estudiantes. De igual manera se han realizado experimentos con datos de la UCI referente a dos instituciones educativas en lo que tiene que ver con cursos de matemáticas y lengua portuguesa.

1.1. Estructura de la Tesis

El trabajo está estructurado de la siguiente manera:

En el primer capítulo se hace una breve descripción del trabajo y la estructura que llevará está investigación.

En el segundo capítulo se describen los conceptos generales, para luego analizar los modelos gráficos probabilísticos sobre todo las redes bayesianas. Adicionalmente se hace referencia a los clasificadores bayesianos, se analizan los diferentes clasificadores como Naive Bayes, TAN,

BAN, entre otros y un algoritmo utilizado en clasificadores no supervisados como EM. Más adelante se realiza un estudio de las diferentes técnicas de aprendizaje no supervisado, como técnica para construir redes bayesianas a partir de una base de datos como K2 que está basado en la optimización de una métrica y PC que se basa en pruebas de independencias entre variables.

En el capítulo tres se analiza un método para la clasificación no supervisada pero que se basa en la creación de una jerarquía similar a Autoclass y se aplica a los datos de deserción estudiantil y evaluación de encuestas.

Para el capítulo cuatro se aplica la metaheurística VMO (Variable Mesh Optimization) para el aprendizaje de redes bayesianas, representando cada nodo de la malla como una red bayesiana a través de un conjunto de arcos.

En el capítulo cinco se realiza la aplicación de técnicas de aprendizaje automático para resolver problemas educacionales, en particular el problema de deserción estudiantil. También se propone un nuevo clasificador bayesiano basado en un método rápido para calcular una frontera de Markov de la variable clase y una estructura de red asociada a dicha frontera. Para finalizar, en el sexto capítulo, se concluye de acuerdo con los resultados obtenidos en los diferentes algoritmos propuestos y sus aplicaciones. Adicionalmente se presentará la posibilidad de realizar futuros trabajos relacionados.

2 Modelos Gráficos Probabilísticos

Los modelos gráficos probabilísticos permiten el desarrollo de modelos probabilísticos que ayudan a la toma de decisiones, representando las dependencias entre variables por medio de grafos.

Se tienen varios tipos de modelos gráficos probabilísticos, como son los modelos ocultos de Markov, las redes bayesianas, diagramas de influencias (García et al.2006), los clasificadores bayesianos, y otros (Koller y Friedman2009). Estas técnicas son usadas por la Inteligencia Artificial para solucionar problemas complejos.

Este capítulo tiene como propósito introducir aspectos teóricos básicos relacionados con los modelos gráficos probabilísticos. Se empezará con el estudio de las redes bayesianas en la sección 2.1, En la segunda sección se hace una explicación detallada de aprendizaje de redes 2.2, donde se distinguen tres tipos: la estimación de una red, el aprendizaje supervisado y el aprendizaje no supervisado mediante técnicas de clúster. A continuación se estudian los clasificadores bayesianos de acuerdo a los modelos de clasificación supervisada 2.3 y no supervisada 2.4. Finalmente, las conclusiones parciales de este capítulo son dadas en la sección 2.5.

2.1. Redes Bayesianas

Si tenemos un conjunto de variables denotadas $\mathbf{X} = \{X_1, X_2, .., X_n\}$ donde cada variable X_i toma valores dentro de un conjunto finito Ω_i; entonces, usamos x_i para expresar uno de los valores de $X_i, x_i \in \Omega_i$. Ahora si tenemos un conjunto de índices denotado como I, el conjunto de variables $\{X_i\}_{i \in I}$ se denotará como $\mathbf{X}_I$ y notaremos Ω_I como el conjunto $\prod_{i \in I} \Omega_i$ a sus elementos se les llama configuraciones de $\mathbf{X}_I$ y serán representadas con $\mathbf{x}$ o $\mathbf{x}_I$.

`Definición de red bayesiana:` es un grafo acíclico dirigido que representa un conjunto de variables aleatorias y sus dependencias condicionales (véase Figura 2.1), donde hay un nodo por cada variable (conjunto de variables), y la topología del gráfico muestra las relaciones de independencia entre variables de acuerdo con el criterio d-separación (Pearl1988). Además, cada nodo X_i tiene una distribución de probabilidad condicional $P_i(X_i|\pi_i)$ para esa

variable, donde Π_i es el conjunto de padres de $\mathbf{X}_i$ en el grafo. De acuerdo con las independencias representables por el grafo, una red bayesiana determina una distribución única probabilidad conjunta:

$$P(\mathbf{X} = \mathbf{x}) = \prod_{i=1}^{n} P_i(x_i|\Pi_i) \qquad \forall \mathbf{x} \in \Omega \tag{2.1}$$

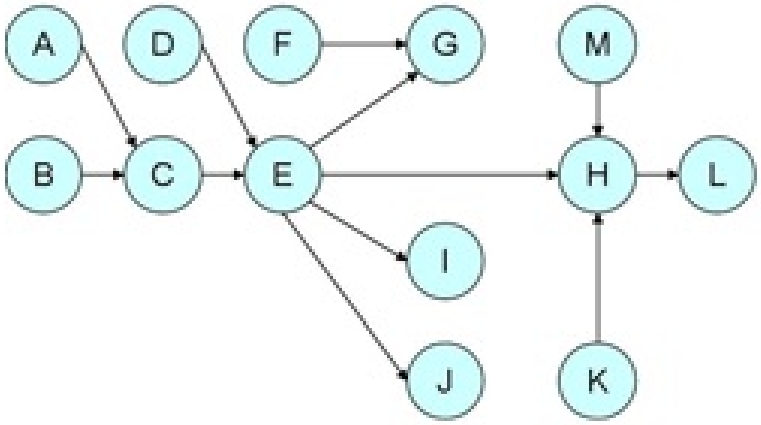

Figura 2.1 Ejemplo de red bayesiana

2.2. Aprendizaje de redes

El aprendizaje consiste en estimar una red bayesiana a partir de un conjunto de observaciones de las variables del problema $\mathscr{D}$. $\mathscr{D} = \{\mathbf{x}^1, \mathbf{x}^2, .., \mathbf{x}^N\}$; donde $\mathbf{x}^i$, es un conjunto de observaciones para todas las variables en $\mathbf{X}$. El aprendizaje automático se divide en aprendizaje supervisado y aprendizaje no supervisado. En el aprendizaje supervisado la atención se centra en la predicción de los valores de una variable especial llamada clase y denotada como C, mientras que en el aprendizaje no supervisado el objetivo es encontrar descripciones compactas de los datos de manera que se aproxime la distribución de conjuntos de todas las variables. En ambos aprendizajes el interés está en los métodos que generalizan bien a los datos nuevos. En este sentido, se diferencia entre los datos que se utilizan para entrenar un modelo y los datos que se utiliza para probar el funcionamiento del modelo estimado.

Para el aprendizaje de una red genérica hay algoritmos basados en las relaciones de independencia como es el caso de PC (Cooper y Herskovits1992) y algoritmos de métricas + búsqueda.

2.2.1. Algoritmo PC

Es uno de los algoritmos utilizados para el aprendizaje bayesiano. Este algoritmo se basa en pruebas de independencias entre variables $I(X_i, X_j|\mathbf{A})$, donde $\mathbf{A}$ es un subconjunto de variables. El algoritmo PC empieza con un grafo completo no dirigido para, posteriormente, ir reduciéndolo. Primero elimina las aristas que unen dos nodos que verifican una independencia condicional de orden cero, después las de orden uno, y así sucesivamente. El conjunto de nodos candidatos para formar el conjunto separador (el conjunto al que se acondiciona) es el de los nodos adyacentes a alguno de los nodos que se pretende separar. Posteriormente se procede a orientar las aristas de forma compatible con las independencias de la etapa anterior. Se puede demostrar que si las independencias del problema se representan exactamente mediante un grafo dirigido acíclico y los test no tienen errores, entonces el algoritmo obtendrá un grafo equivalente (representa las mismas relaciones). En este trabajo usamos la implementación hecha de PC en el programa Elvira (Elvira2002), con un nivel de significancia de 0.05.

2.2.2. Algoritmos métrica + búsqueda

Los algoritmos basados en funciones de métrica + búsqueda pretenden tener un grafo que mejor modelice a los datos de entrada, de acuerdo con un criterio específico o métrica. Además hay un procedimiento de exploración que busca en el espacio de todos los grafos que optimice la métrica considerada. Durante el proceso de exploración, la función de evaluación es aplicada para evaluar el ajuste de cada estructura candidata a los datos. Cada uno de estos algoritmos se caracterizan por la función de evaluación y el método de búsqueda utilizados.

2.2.2.1. Métricas

Cada uno de los algoritmos de métrica + búsqueda utilizan como función de evaluación una métrica como BIC (Schwarz1978), BDEu (Heckerman, Kadie, y Listgarten2007), K2 (Cooper y Herskovits1992). Por eficiencia la métrica usada debe ser descomponible, esto quiere decir que se puede expresar como una suma o un producto de funciones que solo dependen de cada nodo y de sus padres. Así bajo modificaciones locales de un grafo, solo hay que recalcular la parte que corresponde al nodo al que se le ha modificado el conjunto de padres.

Las métricas bayesianas miden la probabilidad de la estructura dado los datos. La métrica K2 para una red G y una base de datos $\mathscr{D}$ es la siguiente:

$$f(G:\mathscr{D}) = \sum_{i=1}^{n} \left[\sum_{k=1}^{r_i} \left[\log \frac{\Gamma(r_i)}{\Gamma(N_{ik}+r_i)} + \sum_{j=1}^{s_i} \log \frac{\Gamma(N_{ijk}+r_i)}{\Gamma(1)} \right] \right] \tag{2.2}$$

Donde: N_{ijk} es la frecuencia de las configuraciones encontradas en la base de datos $\mathscr{D}$ en las que la variable X_i toma el valor x_p $(X_i = x_k)$ cuando los padres toman la configuración número j; s_i es el número de configuraciones posibles del conjunto de padres, r_i es el número de valores que puede tomar la variable X_i, n es el número de variables y Γ es la función Gamma

$$N_{ik} = \sum_{j=1}^{r_i} N_{ijk} \tag{2.3}$$

La métrica BDEu depende de un solo parámetro, el tamaño muestral S y se define de la siguiente manera:

$$g_{BDeu}(G:\mathscr{D}) = \sum_{i=1}^{n}\left[\log\left(\frac{\Gamma(\frac{S}{q_i})}{\Gamma(N_{ij}+\frac{S}{q_i})}\right) + \sum_{k=1}^{r_i}\log\left(\frac{\Gamma(N_{ijk}+\frac{S}{q_i r_i})}{\Gamma(\frac{S}{q_i r_i})}\right)\right] \tag{2.4}$$

La métrica BIC se define de la siguiente manera:

$$f(G:\mathscr{D}) = \sum_{i=1}^{n}\sum_{j=1}^{r_i}\sum_{k=1}^{s_i} N_{ijk}\log\frac{N_{ijk}}{N_{ik}} - \frac{1}{2}C(G)\log N \tag{2.5}$$

Donde: N es el número de registros de la base de datos; $C(G)$ es una medida de complejidad de la red G, definida como:

$$C(G) = \sum_{i=1}^{n}(r_i - 1)s_i \tag{2.6}$$

La métrica Akaike se define de la siguiente manera:

$$f(G:\mathscr{D}) = \sum_{i=1}^{n}\sum_{j=1}^{r_i}\sum_{k=1}^{s_i} N_{ijk}\log\frac{N_{ijk}}{N_{ik}} - C(G) \tag{2.7}$$

Cuando nos refiramos a una métrica genérica sin especificar la métrica concreta que se usa, escribiremos $Score(G:\mathscr{D})$. Todas estas métricas son descomponibles, en el sentido de que se pueden expresar como:

$$Score(G:\mathscr{D}) = \sum_{i=1}^{n} Score(X_i, \pi_i, \mathscr{D}) \tag{2.8}$$

Donde Π_i son los padres de X_i en G

2.2.3. Algoritmo K2

Este algoritmo está basado en la búsqueda y optimización de una métrica bayesiana. K2 realiza una búsqueda voraz y muy eficaz para encontrar una red de buena calidad en un tiempo aceptable (Cooper y Herskovits1992). K2 es un algoritmo heurístico basado en la optimización de una medida. Esa medida se usa para explorar, mediante un algoritmo de ascensión de colinas, el espacio de búsqueda formado por todas las redes que contienen las variables de la base de datos. El algoritmo supone un orden entre las variables. Va añadiendo cercos de forma voraz empezando por una red vacía y considerando en cada caso el arco que produce un mejor aumento de la métrica, hasta que no sea posible obtener una mejora.

Para definir la red con este algoritmo debemos seguir tres pasos importantes:

1. Se tiene un grafo inicial sin arcos
2. Se elige un arco para añadir al grafo desde un nodo X_i a otro X_j; precediendo X_i a X_j en el orden dado
   - Calculamos la métrica de la nueva red con un arco nuevo en cada paso
   - Se escoge el arco con mayor métrica
3. Si el arco nuevo aumenta el valor de la métrica de la nueva red, se añade y se va al paso 2, caso contrario esa es la red de salida y finalizará

2.2.4. Algoritmo voraz de búsqueda

Se basa en una heurística para resolver un problema determinado y con esta poder elegir la opción óptima en cada paso esperanzado en obtener un solución general óptima.

Es uno de los esquemas más simples y al mismo tiempo de los más utilizados. Típicamente se emplea para resolver problemas de optimización donde existe una entrada de tamaño n que son los candidatos a formar parte de la solución, existe un subconjunto de esos n candidatos que satisface ciertas restricciones que viene a ser la solución factible; hay que obtener esta solución factible que maximice o minimice una cierta función objetivo y esta será la solución óptima.

(Heckerman1998) presenta un caso de algoritmo de métrica + búsqueda basado en búsqueda local utilizando operadores de inserción, eliminación e inversión de arcos.

Uno de los elementos claves de estos algoritmos es la función de seleccionar los candidatos que garanticen un resultado óptimo. Al contrario que con otros métodos algorítmicos, no siempre es posible encontrar la solución a un problema empleando un algoritmo voraz. Estos algoritmos tienden a ser bastante eficientes y pueden implementarse de forma relativamente sencilla. Su eficiencia se deriva de la forma en que trata los datos, llegando a alcanzar muchas veces una complejidad de orden lineal.

El espacio de búsqueda es el conjunto de todos los DAGs que contienen las variables. Las operaciones que se pueden realizar son las siguientes:

1. Si dos nodos no son adyacentes, añadir una arista entre ellos en cualquier dirección.
2. Si dos nodos son adyacentes, quitar la arista entre ellos.
3. Si dos nodos son adyacentes, invertir la arista entre ellos.

Todas estas operaciones tienen como restricción de que el grafo resultante sea un grafo sin ciclos. Al conjunto de DAGs que se pueden obtener a partir de G mediante la aplicación de una de las operaciones se la conoce como $\mathbf{Nb}(G)$. Si $G' \in \mathbf{Nb}(G)$, se determina que G' está en el vecindario de G. Este conjunto de operaciones es completo para el espacio de búsqueda. Es decir, para cualquier G o G' existe un conjunto de operaciones que transforma G en G'. No es necesaria la operación de arista inversa para que las operaciones sean completas, pero mejora la conectividad del espacio sin mucha complejidad, lo que conduce a la mejor búsqueda. Por otra parte, cuando se utiliza un algoritmo voraz de búsqueda, incluyendo reversiones de aristas parece conducir a menudo a una mejor máximo local (García2009).

El algoritmo empieza con un DAG sin aristas. A cada paso de la búsqueda, de todos los DAGs en el vecindario de nuestro DAG, el algoritmo voraz elige el que maximiza el $Score(G, \mathscr{D})$. Nos detenemos cuando ninguna operación aumenta la métrica. Nótese que en cada paso, si una arista a X_i es añadida o borrada, solo se necesita re-evaluar $Score(X_i, \Pi_i, \mathscr{D})$. Si una arista entre X_i y X_j es revertida, solo se necesita re-evaluar $Score(X_i, \Pi_i, \mathscr{D})$ y $Score(X_j, \Pi_j, \mathscr{D})$.

```
1: Problema: Encontrar un DAG G que se aproxime a la maximización del score_B(𝒟, G)
2: Ingreso: Una red bayesiana con esquema de aprendizaje estructural BL, conteniendo n
   variables, datos 𝒟
3: Salidas: Un conjunto de aristas E en un DAG Score(G, 𝒟)
4:     E = ∅; G = (V, E)
5:     hacer
6:       si (algún DAG en el vecindario de nuestro actual DAG incrementa Score(G, 𝒟))
7:         modifica E de acuerdo con el que más incrementa Score(G, 𝒟)
8:       mientras (alguna operación incremente Score(G, 𝒟) )
9:     fin
```

Algoritmo 2.1: Algoritmo voraz

2.2.4.1. Aprendizaje de árbol: Algoritmo de Chow-Liu

Chow y Liu proponen un método para aproximar la distribución conjunta de un conjunto de variables discretas usando productos de distribución que afectan no más de un par de variables. Este algoritmo también es conocido como algoritmo de árbol de expansión de máximo peso (MWST), uno de los usos de este algoritmo incluye la construcción de relaciones entre variables gráficas. Este algoritmo se usa para aprender árboles.

El objetivo es encontrar un árbol que maximice la probabilidad de los datos. Se manejan

varios supuestos en este algoritmo, como que no haya datos faltantes, que las variables son discretas y que los datos son distribuidos de forma independiente e idéntica.

(Chow y Liu1968) proporcionan un algoritmo sencillo para la construcción del árbol óptimo; en cada etapa del procedimiento el algoritmo lo que hace es conectar el par de máxima información mutua hasta que todas las variables estén conectadas (Chow y Liu1968).

2.2.4.2. Algoritmo de Chow-Liu

- Calcular el peso $I(X_i, X_j)$ de cada posible arista (X_i, X_j)

$$I(X_i, X_j) = \sum_{x_i, x_j} \hat{P}(x_i, x_j) \log \frac{\hat{P}(x_i, x_j)}{\hat{P}(x_i)\hat{P}(x_j)} \tag{2.9}$$

Donde

$$\hat{P}(x_i, x_j) = \frac{Numero(x_i, x_j)}{N} \tag{2.10}$$

Donde $Numero\ (x_i, x_j)$ es la frecuencia con la que en la muestra ocurre que $X_i = x_i$ y $X_j = x_j$. Análogamente, $\hat{P}(x_i)$ es la frecuencia con la que $X_i = x_i$.

- El algoritmo empieza con un árbol vacío T
- Mientras el grafo no sea conexo
  - Elegir el arco (X_i, X_j) de máximo peso $I(X_i, X_j)$ de entre aquellos arcos en los que no existe un camino de X_i a X_j en T
  - Añadir el arco (X_i, X_j) a T

Las ventajas de los árboles Chow- Liu incluyen:

- La existencia de un algoritmo simple y polinómico para encontrar el árbol óptimo
- La estructura de árbol resultante T tiene a menudo una simple interpretación

2.2.5. Restricciones

Las restricciones pueden ser usadas en conjunción con algoritmos para aprendizaje de redes bayesianas y se debe hacer referencia al estudio de tres tipos de restricciones en la estructura del grafo definido por el dominio **X** a saber: existencia, ausencia y orden.

Todos estos tipos deberán ser consideradas restricciones duras (opuesto a restricciones suaves) (Heckerman, Geiger, y Chickering1995). En este sentido se supone que para las redes bayesianas que representan el dominio del conocimiento este tipo de restricciones son verdaderas, y por lo tanto deberán necesariamente ser satisfechas por todas las redes bayesianas candidatas.

(Acid y De Campos2003), indica que la simple adaptación de los algoritmos de búsqueda puede resultar algo ineficiente por lo que es preferible adaptarlo a las características propias del algoritmo de aprendizaje que vayamos a usar.

Existen otros algoritmos de métrica + búsqueda más sofisticados que una búsqueda local simple que pueden ser fácilmente extendidos para tratar de forma eficiente las restricciones. Son muchos los algoritmos de aprendizaje de redes bayesianas que desarrollan una búsqueda más poderosa que la búsqueda local pero que usan los tres operadores básicos como: búsqueda de entornos variables-VNS (De Campos y Puerta2001), búsqueda tabú (Acid y De Campos2003)), o un subconjunto de estos como optimización por colonia de hormigas-ACO (De Campos et al.2002).

2.2.5.1. Restricciones de existencia

Debemos considerar dos tipos de restricciones de existencia a saber: existencia de arcos y existencia de enlace. Muchas veces encontramos que dentro de las reglas que se establecen, está la obligatoriedad de que se enlace la variable clase con los atributos, a esto se le denomina arcos forzados, ya que siempre estarán enlazados.

Un ejemplo del uso de este tipo de restricciones podría ser un algoritmo BAN (Cheng y Russell1999), que no es más que un clasificador naive bayes aumentado con una red, en el que se agrega una estructura general de dependencia entre atributos, sin otras limitaciones. Este corrige la estructura de naive Bayes (arcos desde la variable clase a todas los atributos) y búsquedas para las conexiones adicionales de pares de arcos de los atributos, (De Campos y Castellano2007).

2.2.5.2. Restricciones de ausencia

Se deben considerar dos tipos de restricciones de ausencia a saber: ausencia de arcos y ausencia de enlaces. Dentro de las reglas que se establecen se pueden determinar reglas que prohíban el enlace de la variable clase con los atributos, a esto se lo conoce como arcos prohibidos.

Un ejemplo de las restricciones de ausencia es un clasificador naive Bayes (Langley y Sage1994), que prohíbe arcos entre los atributos y también arcos desde los atributos a la variable clase, (De Campos y Castellano2007).

2.2.5.3. Restricciones de orden parcial

Las restricciones de orden pueden representar la precedencia temporal o funcional entre variables. Ejemplos de uso de restricciones de orden son todos los algoritmos de aprendizaje de redes bayesianas que requiere un orden total fijo de las variables como el bien conocido algoritmo K2 (Cooper y Herskovits1992).

2.3. Clasificación supervisada

Generalmente, en la clasificación supervisada se asume la existencia de dos tipos de variables: las variables predictoras, $\mathbf{X} = (X_1, ..., X_n)$, y la variable clase, C. Mediante los clasificadores supervisados se trata de aprender las relaciones entre las variables predictoras y la clase, de forma que se pueda asignar un valor de C a un nuevo caso, $\mathbf{x} = (x_1, ..., x_n)$, en el que el valor de la clase es desconocido. En este caso ya se tiene conocimiento de manera a priori de la clase. El inconveniente es encontrar una función que asigne a cada instancia su respectiva clase correspondiente (Friedman, Geiger, y Goldszmidt1997). Lo manifestado se da a partir de un conjunto de variables $\mathbf{V} = \mathbf{X} \cup C$; donde C es la clase y las variables en $\mathbf{X}$ son variables predictorias que serán utilizadas para predecir los valores de C (Corso2009).

`Definición de aprendizaje supervisado:`. Dado un conjunto de datos $\mathscr{D} = \{(\mathbf{x}^i, c^i), i = 1, ..., N\}$; donde $(\mathbf{x}^i, c^i)$ es una observación de los atributos y la clase en un caso concreto. Se debe aprender la relación entre la entrada $\mathbf{X}$ y la salida C de manera que, cuando se tenga una nueva entrada $\mathbf{x}^*$ se puede predecir el valor de C con c^*. La pareja $(\mathbf{x}^*, c^*)$ no está en $\mathscr{D}$, pero se asume que se generan por el mismo proceso desconocido que generó a $\mathscr{D}$. Para especificar explícitamente lo que significa con precisión se define una función de pérdida $L(c^{pred}, c^*)$ o, a la inversa, una función utilidad $U = -L$. Aquí la utilidad suele ser 0 o 1, $L(c^{pred}, c^*) = 1$; si $(c^{pred} \neq c^*)$, $L(c^{pred}, c^*) = 0$; $si(c^{pred} = c^*)$. En este tipo de aprendizaje lo importante es la distribución condicional $p(C|X, \mathscr{D})$. La salida también es llamada una "etiqueta", sobre todo cuando se habla de clasificación. El término 'clasificación' denota el hecho de que la etiqueta es discreta, es decir, que consiste en un número finito de valores (Murphy2012).

Un clasificador bayesiano recibe la probabilidad posterior de cada clase, c_i, usando regla de Bayes, como el producto de la probabilidad a priori de la clase por la probabilidad condicional de los atributos $(\mathbf{x}^i)$ dada la clase, dividido por la probabilidad de los atributos, (Sucar2008).

$$P(c_i|\mathbf{x}^i) = P(c_i)P(\mathbf{x}^i|c_i)/P(\mathbf{x}^i) \tag{2.11}$$

El desarrollar clasificadores eficaces y eficientes es un problema importante debido a la gran cantidad de datos que día a día son generados.

La clasificación es una tarea primordial del análisis de datos y reconocimiento de patrones que necesita la construcción de un clasificador; es decir, una función que asigna una clase a los casos descritos por un conjunto de atributos, (Friedman, Geiger, y Goldszmidt1997).

Cualquier red bayesiana puede ser usada para realizar una clasificación solo con distinguir la variable de interés del problema como la variable clase, para luego aplicarle algún algoritmo de propagación con las nuevas evidencias y una regla de decisión para elegir un valor de la clase en función de la distribución condicional.

Para determinar un buen clasificador se debe considerar que este proporcione clasificaciones correctas (exactitud), que sea rápido al hacer la clasificación (rapidez), que sea lo más comprensible para los humanos (claridad), y que el tiempo para obtener o ajustar el clasificador a partir de datos sea razonable, (tiempo de aprendizaje). Se puede determinar la tasa de error o la exactitud de un clasificador como la probabilidad con la que clasifica de manera adecuada un caso seleccionado al azar, (Kohavi1995b). También se puede determinar como el número de casos clasificados de manera adecuada entre el número total de elementos (caso de pérdida 0-1).

$$exactitud = \frac{\#de-casos-clasificados-adecuadamente}{\#total-de-casos} \tag{2.12}$$

$$costemedio = \frac{\#de-casos-mal-clasificados}{\#total-de-casos} \tag{2.13}$$

El problema que se encuentra en la clasificación (supervisada) es obtener el valor más probable de una variable (hipótesis) dado los valores de otras variables (evidencia, atributos).

2.3.1. Naive Bayes (NB)

Vamos a estudiar la clasificación supervisada y el modelo gráfico para clasificación supervisada más usado y sencillo es **Naive Bayes (NB)**. NB se basa en dos supuestos: primero, que cada atributo es condicionalmente independiente de los atributos dada la clase y segundo, que todos los atributos tienen influencia sobre la clase.

Esto no siempre se cumple, pero debido a su simplicidad NB ha demostrado ser competitivo, en términos de precisión de clasificación en muchos dominios que varios algoritmos más complejos, como las redes neuronales y árboles de decisión, (Webb y Pazzani1998). El algoritmo utiliza los datos de entrenamiento para estimar todos los valores de la probabilidad requeridos.

Si se tiene una variable a clasificar C, la probabilidad de que un ejemplo pertenezca a la clase i-ésima de C, se calcula aplicando el teorema de Bayes (Langley y Sage1994), tal como se muestra a continuación:

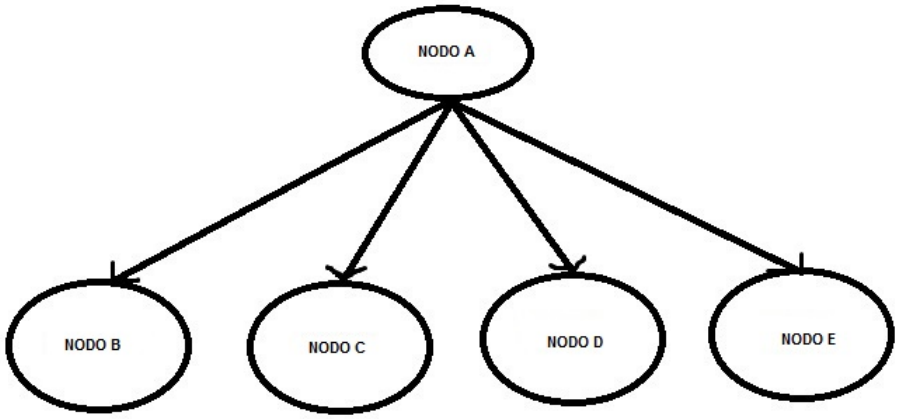

Figura 2.2 Clasificador Naive Bayes

$$P(C = c_i|X_1 = x_1, ..., X_n = x_n) \propto P(C = c_i) \times P(X_1 = x_1, ..., X_n = x_n|C = c_i) \tag{2.14}$$

Como se supone que dado C las variables predictoras sean condicionalmente independientes, obtenemos:

$$P(C = c_i|X_1 = x_1, ..., X_n = x_n) \propto P(C = c_i)\ \Pi_{r=1}^{n} P(X_r = x_r|C = c_i) \tag{2.15}$$

Este clasificador cuenta con unas ventajas significativas como son:

- Bajo tiempo de clasificación
- Bajo tiempo de aprendizaje
- Bajos requerimientos de memoria
- Sencillez
- Buenos resultados en muchos dominios

El buen rendimiento de este clasificador NB ha servido como impulso para desarrollar más investigaciones basados en modelos gráficos probabilísticos que debilite la suposición de independencia. Entre otros, podemos mencionar a la red bayesiana con estructura de árbol (TAN), (Friedman, Geiger, y Goldszmidt1997). que se descibe a continuación.

2.3.2. Clasificador Bayesiano Simple Aumentado con un Árbol (TAN)

Se puede manifestar que este modelo es una red bayesiana que tiene una estructura de árbol entre los atributos. TAN se recalcula con una adaptación del algoritmo presentado por Chow-Liu en

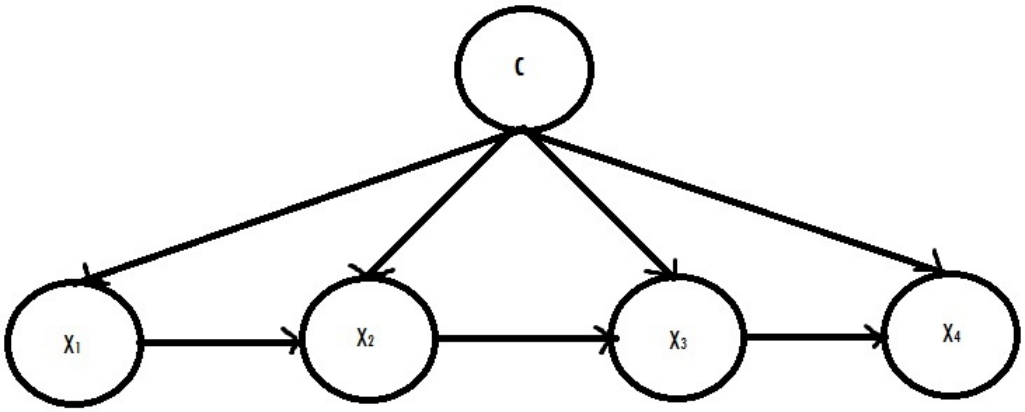

Figura 2.3 Clasificador TAN

1968, donde consideran la cantidad de información recíproca ajustada a la variable clase, en lugar de la que se usa en el algoritmo de Chow-Liu. (Friedman et al.2000). En TAN se agrega una estructura de árbol de dependencias entre atributos, por lo que en un principio se tienen pocas conexiones y no aumenta de forma significativa la complejidad de la estructura. Lo anterior se ilustra en la Figura 2.3. Se puede indicar que en este modelo, C es la varible a clasificar la misma que tiene un conjunto de padres vacíos, mientras que el conjunto de variables padres de las variables predictoras, X_i, contiene a la variable a clasificar, y como máximo otra más (Chávez2008).

La cantidad de información recíproca entre las variables X_i, X_j (que deben ser discretas) condicionada a la variable C se define como:

$$I(X_i, X_j|C) = \Sigma_i \Sigma_j \Sigma_r \hat{P}(x_i, x_j, c_r) log \frac{\hat{P}(x_i, x_j|c_r)}{\hat{P}(x_i|c_r)\hat{P}(x_j|c_r)} \tag{2.16}$$

(Friedman et al.1997), propone un algoritmo que optimiza la verosimilitud de la estructura dados los datos. Esta representación simplemente es el algoritmo Chow-Liu, pero en vez de usar $I(X_i, X_j)$, usa $I(X_i, X_j|C)$

- Calcular la cantidad de información recíproca para cada par de variables (X_i, X_j) condicionadas por la variable C
- Construir un grafo no dirigido completo con n nodos, uno por variable predictora, en la que el peso de la arista está dada por la información recíproca entre las dos variables unidas por la arista condicionada a la variable clase
- Aplicar el algoritmo de Kruskall para el árbol generador minimal partiendo de los $n(n-1)/2$ pesos obtenidos en el paso anterior.
- Añadir la variable C y una arista desde esta variable a cada uno de los atributos predictores.

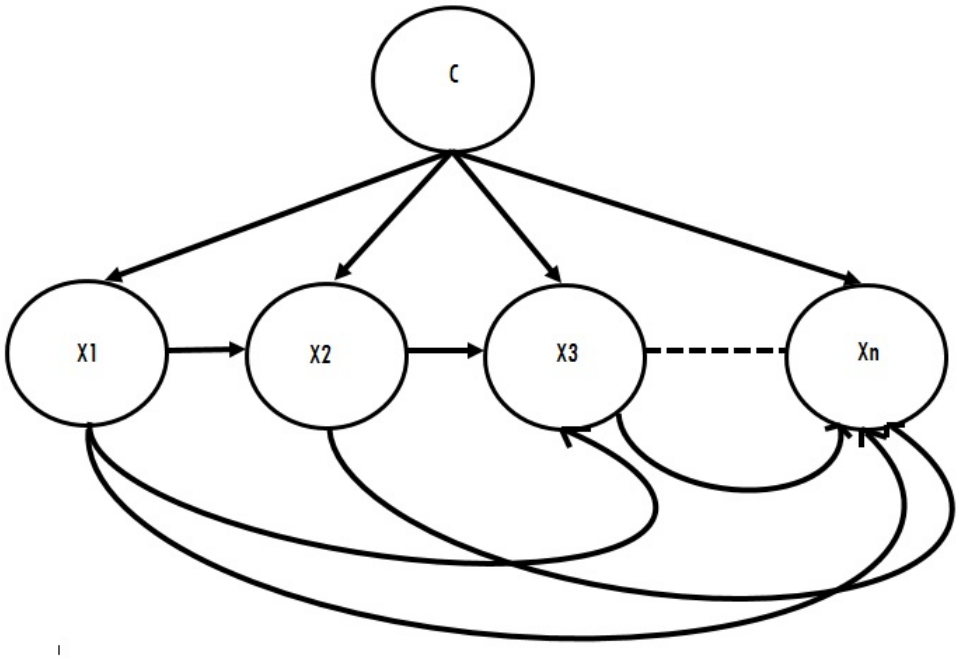

Figura 2.4 Clasificador bayesiano k-dependiente

Para construir el árbol expandido de máximo peso se lo hace de la siguiente manera:

1. Seleccionamos las dos aristas de mayor peso y asignamos al árbol que se va a construir
2. Determinamos la siguiente arista de mayor peso, y la incluimos en el árbol; si forma un ciclo se descarta y se examina la siguiente arista de mayor peso
3. Repetimos el paso 2 hasta que se hayan seleccionado $n-1$ aristas

Este clasificador es un caso particular de redes bayesianas, por lo que para realizar la clasificación en base a estos modelos se utilizan técnicas de propagación de probabilidades en redes bayesianas.

2.3.3. Clasificadores Bayesianos k-dependientes (KDB)

KDB es una generalización de TAN que permite que cada atributo tenga un máximo de k atributos como padres en lugar de 1 como TAN. Según (Sahami1996), en función de la disponibilidad de la capacidad computacional podemos elegir un grado, k, más alto o bajo de dependencias. Podemos visualizar el resultado en la figura 2.4.

El pseudocódigo puede visualizarse a continuación, (Friedman et al.1997):

- Paso 1. Por cada variable predictora X_i $i = 1, ..., n$ se debe calcular la cantidad de información mutua de X_i con respecto a la clase $C, I(X_i, C)$
- Paso 2. Para cada par de variables predictoras se debe calcular la cantidad de información mutua condicionada a la clase, $I(X_i, X_j|C)$, con $i \neq j, i, j = 1, ..., n$

- Paso 3. Inicializar en vacío una lista de variables $\mathbf{Y}$
- Paso 4. Inicializar la BN a construir, la misma que debe tener un único nodo, el correspondiente a la variable C
- Paso 5. Repetir hasta que $\mathbf{Y}$ incluya a todas las variables del dominio:
- Paso 5.1. Seleccionar de entre las variables que no están en $\mathbf{Y}$, aquella $X_{\text{máx}}$ con mayor cantidad de información mutua con respecto a $C, I(X_{\text{máx}}, C) = \text{máx}_{X \notin \mathbf{Y}} I(X, C)$
- Paso 5.2. Añadir $X_{\text{máx}}$ a BN
- Paso 5.3. Añadir un arco C a $X_{\text{máx}}$ en BN
- Paso 5.4. Añadir $m = \text{mín}(|\mathbf{Y}|, k)$ arcos de las m variables distintas $X_j \in \mathbf{Y}$ que tengan los mayores valores $I(X_{\text{máx}}, X_j|C)$
- Paso 5.5. Añadir $X_{\text{máx}}$ a $\mathbf{Y}$
- Paso 6. Estimar las probabilidades condicionadas necesarias para especificar la BN

Este es un algoritmo basado en el presentado por (Friedman et al.1997), que permite que cada variable tenga un número de padres, sin considerar la variable clase C, acotado por k.

2.3.4. Clasificador Bayesiano Simple Aumentado con una Red (BAN)

Si se tiene en cuenta la existencia de una variable especial (la variable a clasificar), la exactitud de las redes bayesianas como clasificadores supervisados aumenta. Para que esto suceda de una manera fácil y sencilla se debe trabajar como sucedía con el clasificador Naive Bayes; o sea, a través de la estructura de la red, en el que se contaba con un arco entre la clase y cada uno de los atributos (Cheng y Russell1999).

BAN es un clasificador bayesiano simple aumentado con una red. Es una extensión del clasificador TAN que permite a los atributos formar un grafo arbitrario en vez de solo un árbol (Friedman et al.1997). Para BAN se considera una estructura general de dependencia entre atributos, sin limitaciones, aunque se fuerza que haya un enlace entre cada atributo y la clase. Un clasificador BAN con n atributos se puede considerar un clasificador k-dependiente con $k = n$.

Sean $X_1, X_2, ..., X_n$ las variables predictoras y sea C la variable a clasificar (García2009) propone un esquema general de aprendizaje de un clasificador BAN tal como se indica en el algoritmo 2.2. Así como la estructura de BAN se muestra en la figura 2.5, donde se puede observar que la diferencia con TAN consiste básicamente en que los atributos (X_1, X_2, X_3, X_4) forman una estructura de red, aunque el nodo clase sigue conectado con todos los atributos, (Sucar y Martínez-Arroyo2011).

1 Quedarse sólo con las variables predictoras $X_1, X_2, ..., X_n$ y sus correspondientes casos del conjunto de entrenamiento
2 Construir una red bayesiana con esas variables utilizando algún algoritmo de aprendizaje estructural de redes bayesianas (no de clasificadores)
3 Añadir C como nodo padre de todas las variables X_i de la red. Realizar el aprendizaje paramétrico de la estructura obtenida.

Algoritmo 2.2: Algoritmo construcción BAN

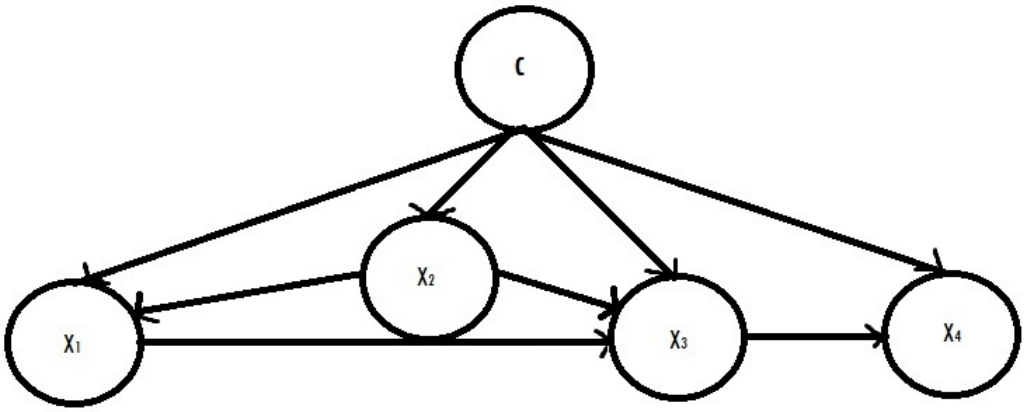

Figura 2.5 Estructura simple BAN

2.3.5. Clasificador Grafos Parcialmente Dirigidos, Acíclicos y Restringidos C-RPDAG

Las redes bayesianas (sin ningún tipo de restricción estructural) también pueden ser usadas para clasificar. En estos casos, cuando se usan como clasificadores, se les denominan redes bayesianas sin restricciones. Cualquier red bayesiana puede ser utilizada en clasificación supervisada, para ello solamente es necesario el manto de Markov de la variable clase. Se debe considerar de que un clasificador bayesiano sin restricciones tiene mayor poder expresivo que un modelo restringido estructuralmente (García2009).

(Acid, De Campos, y Castellano2005), presentaron un clasificador C-RPDAG que construye clasificadores que son redes bayesianas arbitrarias. Se basa en el algoritmo de aprendizaje RPDAG, que es un algoritmo basado en técnicas de métrica + búsqueda. La particularidad del algoritmo de aprendizaje de RPDAGs está en una modificación del espacio de búsqueda. En principio se considera el espacio de las redes parcialmente orientadas (PDAGs) que representan un conjunto de DAGs equivalentes (representan las mismas independencias). Posteriormente este espacio se modifica al espacio de los RPDAGs (PDAGs restringidos) en los que se imponen unas condiciones adicionales a los PDAGs que hace que un mismo PDAG se parta en distintos RPDAGs. El resultado es que se agranda el espacio de búsqueda, pero se gana en eficiencia cuando se recorre dicho espacio.

Los C-RPDAGs siguen la misma filosofía que el RPDAG, pero considera que se tiene un problema de clasificación con clase C. Considera que los RPDAGs son equivalentes si producen

los mismos resultados en relación a C. De esta forma la búsqueda se restringe a un manto de Markov de C con la que resulta ser más eficiente y efectiva.

Al utilizar C-RPDAGs, se obtiene una reducción notable de configuraciones en el espacio de búsqueda respecto a los DAGs, aunque no es la única ventaja. Por otro lado, respecto a los métodos de búsqueda basados en DAGs aumentados, aquellos clasificadores en los que $C \in \Pi(X) \forall X \in \mathbf{X} \setminus \{C\}$ (como TAN y BAN), el espacio de búsqueda de C-RPDAGs no impone la subestructura de Naive Bayes, aunque éste pueda obtenerse si fuera la estructura realmente apropiada. También se pueden obtener estructuras más generales, como aquellas en las que un nodo es padre de un hijo de la clase, y este nodo no está conectado a la clase (García2009).

2.3.6. Árboles de Clasificación

Otro modelo de clasificación no basado en redes bayesianas son los árboles de clasificación. Un árbol de clasificación es un modelo jerárquico de predicción muy popular y potente de descubrimiento de conocimiento.

Un árbol de decisión, o un árbol de clasificación, es usado para aprender una función de clasificación con el que concluye el valor de un atributo dependiente (variable) dado los valores de los atributos independientes (variables) (Bhargava et al.2013).

Un árbol incluye un nodo raíz, nodos hoja que representan las clases, nodos internos que representan las condiciones de prueba (que se aplican en los atributos).

Cada nodo (no terminal) detalla un test de algún atributo de la instancia, mientras que a cada rama que emerge de ese nodo le compete un posible valor de dicho atributo. Las ramificaciones se generan en forma recursiva hasta que se cumplan ciertos criterios de parada. Se denotan los nodos de decisión como nodos circulares y los nodos hoja o nodo terminales como nodos rectangulares (Quinlan1986).

Los factores considerados que han influido en la difusión del uso de los árboles de clasificación son la accesibilidad a diferentes implementaciones, la explicación que entrega de la clasificación, la representación gráfica y la velocidad con la que puede clasificar nuevos patrones. La figura 2.6 muestra un ejemplo de un árbol de clasificación con tres características binarias y una variable clase también binaria.

Cuando un árbol de clasificación va a aprender a partir de un conjunto de datos se debe en primer lugar contestar a una serie de preguntas que se plantean, como por ejemplo ¿De qué manera asigno los valores de la clase a las hojas?, ¿Es necesaria la poda del árbol para reducir su tamaño?. Dependiendo de las respuestas, se pueden determinar diferentes tipos de árboles de clasificación, entre los más destacados encontramos el árbol de clasificación Dirichlet (Abellán y Moral2003), el árbol C4.5 (Quinlan1993) y su antecesor, el ID3 (Quinlan1986).

Un árbol de clasificación para construirse empieza considerando la muestra completa. Luego se separan las observaciones en r_i grupos mutuamente excluyentes dependiendo de los valores del atributo en el nodo raíz. Cada una de estas particiones pueden a su vez ser partida nuevamente

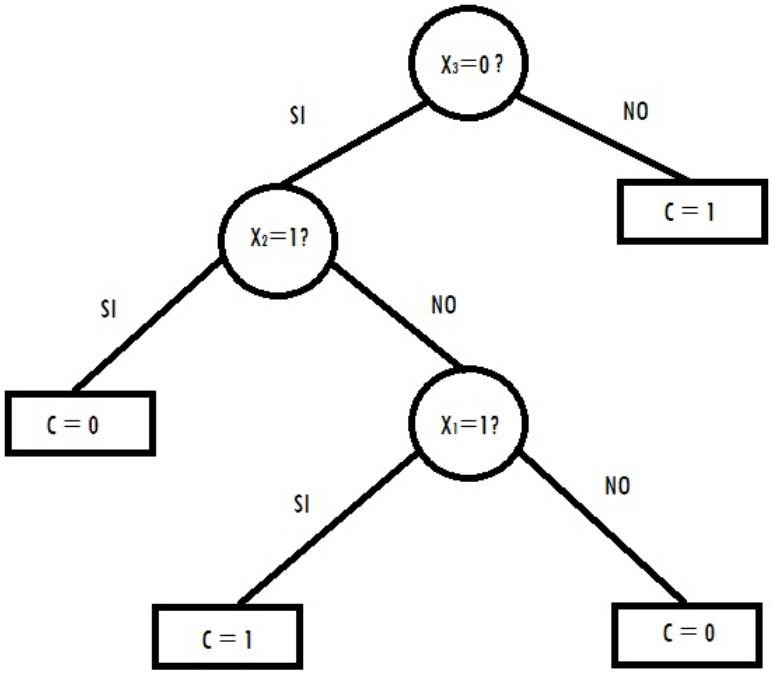

Figura 2.6 Árboles de clasificación

(se considera la parte recursiva del algoritmo) y estarán definidas por el valor que toma una variable en particular. Una de las características que se tiene en la construcción de árboles de clasificación es que solo se puede considerar una variable a la vez.

La pregunta técnica que se genera es ¿Cómo decidimos dónde cortar? y entonces empezaremos a crear las particiones respondiendo a:

- ¿Con respecto a qué variable cortamos?

Obviamente se va a responder a ambas preguntas de una forma tal que maximice el ajuste o que se logre la mejor clasificación posible de las observaciones y para ello necesitamos una métrica que mida el "mejor ajuste", dependiendo el algoritmo de la métrica usada.

El algoritmo 2.3 indica que a partir de un conjunto de casos de entrenamiento $\mathscr{D}$ se puede obtener un árbol de clasificación $\mathbf{T}$ con subárboles $\mathbf{T}_1, ..., \mathbf{T}_{r_i}$.
Al construir los árboles se debe considerar la decisión de la división en un nodo y el criterio de parada en el desarrollo del árbol. Nosotros trabajamos con variables discretas por lo que tendremos que plantearnos si desarrollamos una rama por cada una de las categorías que tiene dicha variable, o agruparla en dos o más conjuntos, (Larrañaga, Inza, y Abdelmalik2000).
Lo peor que puede pasar en una partición es que dos clases estén igualmente representadas y lo mejor que puede pasar es que en una partición solo una de las dos clases esté presente.

(Quinlan1986), introduce uno de los algoritmos de inducción de árboles de clasificación más populares denominado ID3.

En el mismo árbol el principio escogido para seleccionar la variable que más información entrega está basado en la idea de cantidad de información mutua entre esa variable y la variable

1: Entrada: Conjunto de casos de entrenamiento, $\mathscr{D}$, con sus variables y su clase
2: Salida: Árbol de clasificación ($\mathbf{T}$)
3: Begin
4: **do**
5: **if** $\mathscr{D}$ son de la misma clases $C = C_j$ **then**
6: Resultado nodo simple etiquetado como C_j
7: **else**
8: Begin
9: Seleccionar un atributo (X_i con valores $x_{i_1},...,x_{i_r}$ con la métrica considerada
10: Particionar $\mathscr{D}$ en $\mathscr{D}_1,...,\mathscr{D}_r$ de acuerdo a los valores de X_i
11: Construir subárboles $\mathbf{T_1},...,\mathbf{T_r}$ para $\mathscr{D}_1,...,\mathscr{D}_{r_i}$
12: El resultado final es un árbol con raíz $\mathbf{X}_i$ y subárboles $\mathbf{T_1},...,\mathbf{T_{r_i}}$
13: Las ramas entre $\mathbf{X}_i$ y los subárboles están etiquetadas mediante $x_{i_1},...,x_{i_r}$
14: **end if**
15: end

Algoritmo 2.3: Algoritmo general

clase. Los términos usados en este contexto para denominar a la cantidad de información mutua es la de ganancia en información. Esto es debido a que $I(X_i,C) = H(C) - H(C|X_i)$ y lo que viene a representar dicha cantidad de información mutua entre X_i y C es la reducción en incertidumbre en C debido al conocimiento del valor de la variable X_i, (Servente y García2002). La función de información mutua se puede expresar como:

$$I(c,x_i) = \sum_{c,i} \hat{p}(c,x_i) log \frac{\hat{p}(c,x_i)}{\hat{p}(c)\hat{p}(x_i)} \tag{2.17}$$

Como mejora al algortimo ID3 (Quinlan1993), plantea una extensión conocida como C4.5. Este algoritmo genera un árbol a partir de los datos mediante particiones realizadas recursivamente según la estrategia de primero en profundidad (depth-first) (Servente y García2002).

C4.5 considera todas las pruebas que pueda dividir el conjunto de datos y selecciona la prueba que resulta de mayor ganancia de información o en la mayor razón de ganancia de información (Servente y García2002). C4.5 considera una información entre un atributo y la clase normalizada para evitar la tendencia de la ganancia de la información a elegir los atributos con más casos. La razón de ganancia de información entre un atributo y la clase se define como:

$$IR(C,X_i) = \frac{I(C,X_i)}{H(X_i)} \tag{2.18}$$

Donde $H(X_i)$ es la entropía de X_i.

Este tipo de árboles permite trabajar con valores continuos para los atributos, separando los posibles resultados en dos ramas. Además utiliza criterios estadísticos para impedir que el árbol

sobreajuste los datos (que crezca demasiado).

2.3.6.1. Árboles J48

J48 es una implementación en código abierto del algoritmo C4.5 desarrollado por (Quinlan1993) para Weka (Waikato Environment for Knowledge Analysis) (Garner1995). Este algoritmo construye árboles de decisión desde los casos o instancias de entrenamiento usando la razón de ganancia de información e incorpora un mecanismo de post-poda para hacerlos más comprensibles y para mejorar la generalización del modelo. Es capaz de extraer conocimiento de conjuntos de datos con atributos categóricos y continuos, (Quinlan1993).

La manera más frecuente de limitar el problema de sobreajuste es modificar los algoritmos de aprendizaje de tal manera que obtengan modelos más generales. En el contexto de los árboles de decisión y conjuntos de reglas, generalizar significar eliminar condiciones de las ramas del árbol o de algunas reglas. En el caso de los árboles de decisión dicho procedimiento se puede ver gráficamente como un proceso de 'poda'.

La poda es una de las primeras y más simples modificaciones que se han ideado para mejorar el comportamiento de los árboles de decisión. Con posterioridad se han definido otra serie de operadores y modificaciones, generalmente denominados operadores de 're-estructuración'. Por ejemplo, el C4.5 realiza lo que se conoce como 'colapso' (collapsing) (Quinlan1993). Otros operadores de modificación de la topología del árbol son la 'transposición', la 'transposición recursiva' y la 'poda virtual' (Utgoff, Berkman, y Clouse1997).

2.3.6.2. Árboles Bosque Aleatorio

Bosque Aleatorio es un clasificador en el que un conjunto de árboles de decisión son aprendidos para asignar una etiqueta (o clasificación) a una muestra de entrada.

Bosque Aleatorio es una combinación de estructuras de árboles predictores en la que cada árbol depende de unos valores elegidos de forma independiente a partir de la muestra original $\mathscr{D}$ y con la misma distribución para cada uno de estos. Para clasificar, cada árbol emite un voto unidad para un valor de la clase. La predicción es el valor con más votos. Al tener un conjunto cada vez mayor de árboles y dejar que ellos voten por la clase más popular nos permite decir que se han logrado significativas mejoras en la precisión de clasificación.

Para construir la matriz de datos, se van tomando los datos de manera aleatoria por lo que no todos son considerados (luego ayudarán a la poda del árbol) y algunos pueden hasta ser considerados más de una vez. A partir de aquí se construye el árbol.

La aleatorización ha sido una herramienta poderosa para obtener diferencias en los clasificadores. Anteriormente se han utilizado para inicializar algoritmos de entrenamiento con diferentes configuraciones que eventualmente producen diferentes clasificadores. Este algoritmo mejora la precisión en la clasificación mediante la incorporación de aleatoriedad en la construcción de

cada clasificador individual. Esta aleatorización puede introducirse en la partición del espacio (construcción del árbol), así como en la muestra de entrenamiento (Breiman2001).

El diseño de un árbol de decisión requiere la elección de un atributo y un método de poda. Hay muchos enfoques para la selección de los atributos utilizados en la inducción de árbol de decisiones y la mayoría de estos enfoques asignan una medida de calidad directamente al atributo (Quinlan1993).

En bagging cada clasificador es construido de manera individual para generar un entrenamiento formado por un conjunto de datos N extraídos al azar con remplazamiento, donde N es el tamaño original del conjunto de datos (Breiman2001).

En árboles individuales, se suele considerar todos los atributos como candidatos para particionar un nodo interior, pero para aumentar la diversidad Random Forest selecciona un subconjunto de candidatos de forma aleatoria.

El esquema de ejecución del algoritmo de Random Forest es:

- Aleatoriamente se crea (seleccionando con reemplazamiento) un conjunto de entrenamiento de igual tamaño que el conjunto original. Al seleccionarse aleatoriamente con reemplazo no todos los datos de conjunto inicial estarán en el conjunto de entrenamiento. La probabilidad de que un dato particular esté en el conjunto de entrenamiento es aproximadamente e^{-1} cuando el tamaño de la muestra tiende a infinito.
- En cada punto de división del árbol o nodo, la búsqueda de la mejor variable para dividir los datos no se realiza sobre todas las variables sino sobre un subconjunto, m, de las mismas. La elección del subconjunto de variables se realiza de forma aleatoria.
- Se busca la mejor división de los datos de entrenamiento teniendo en cuenta solo a m variables seleccionadas. Para esta tarea se debe considerar una función objetivo como la información mutua o la ganancia de información.
- Los anteriores procesos son repetidos varias veces, de forma que se tienen un conjunto de árboles de decisión entrenados sobre diferentes conjuntos de datos y de atributos.
- Una vez el algoritmo entrenado, la evaluación de cada nueva entrada es realizado con el conjunto de árboles. La categoría final de la clase (clasificación) se obtiene por el voto mayoritario del conjunto de árboles, y en caso de regresión por el valor promedio de los resultados.

Random Forest engloba cualquier multiclasificador basado en conjuntos de árboles, en el que se varía algún parámetro de cada árbol de forma aleatoria. Como característica distintiva, cada clasificador base mantiene la diversidad que le viene dada por el re-muestreo con reemplazo, la cual aumenta teniendo en cuenta que en cada nodo varía de forma aleatoria los atributos a considerar. Este clasificador es robusto al ruido y altamente paralelizable (Breiman2001).

2.3.7. Reglas de clasificación

El aprendizaje de reglas de clasificación es un problema clásico del aprendizaje automático. La mayoría de los métodos tratan de generar reglas siguiendo una estrategia de cubrimiento secuencial. Estos métodos utilizan un conjunto de entrenamiento (o aprendizaje) compuesto por objetos descritos por atributos de condición y el rasgo de decisión (clase) (Filiberto et al.2011).

Las reglas de clasificación inducidas por los sistemas de aprendizaje automático son juzgadas por dos criterios: la precisión de la clasificación en un conjunto de pruebas independientes ('precisión'), y su complejidad. La relación entre estos dos criterios es, por supuesto, de gran interés para la comunidad de aprendizaje automático (Holte1993).

Existen en la literatura algunos indicios de que las reglas de clasificación muy simples pueden lograr sorprendentemente alto grado de exactitud en muchos conjuntos de datos. Por ejemplo, (Rendell y Seshu1990), manifiesta que ocasionalmente muchos conjuntos de datos del mundo real tienen 'pocos picos (a menudo sólo uno)' y así son 'fáciles de aprender'.

En la presente tesis se realizan experimentos en los que se clasifica considerando el uso de tablas de decisiones que lo estudiaremos en la subsección 2.3.7.1 y del clasificador ZeroR en 2.3.7.2.

2.3.7.1. Tablas de decisión

Son conjuntos de reglas de la forma 'Si $\mathbf{Y} = \mathbf{y}$ y entonces $C = c$' donde $\mathbf{Y} \subseteq \mathbf{X}$. Estas reglas no deben de ser inconsistentes al asignar a un caso $\mathbf{X} = x$ dos valores de la clase distintas. En esta tesis se realizan experimentos utilizando un clasificador de tablas de decisión, que no es más que un modelo que genera un conjunto de reglas sencillas según las cuales se decide la clase. En otras palabras se puede determinar como un predictor fácil y rápido, pero que en condiciones correctas puede dar resultados a la altura de modelos más complejos (Carra2016). Se debe tratar de establecer dentro del problema las acciones a efectuar y los diferentes requisitos que se deben cumplir para que dichas acciones sean ejecutadas (Kohavi1995a).

2.3.7.2. ZeroR

Es el clasificador base, se usa para comparar con otros métodos. Simplemente predice la clase mayoritaria, y es útil para tener una referencia del límite inferior de eficiencia y se usará como referencia para los demás clasificadores. Si la clase es numérica se predice la media, si la clase es nominal se predice la moda. Si un método es peor que este, posiblemente haya sobreajuste (Carra2016).

2.3.8. Reglas de asociación

Las reglas de asociación es una propuesta de (Agrawal, Imieliñski, y Swami1993), y son similares a las reglas de clasificación, pero se diferencian porque pueden predecir cualquier atributo o combinación de atributos. La obtención de reglas de asociación consiste en encontrar un modelo que describa dependencias significantes (o asociaciones) entre ítems de una base de datos (Han, Pei, y Kamber2011). Una de las aplicaciones más conocidas es el análisis de la cesta de mercado, porque puede ser semejante al análisis de artículos que frecuentemente se reúnen en una cesta por compradores en un mercado (Agrawal, Imieliñski, y Swami1993).

Las reglas de asociación en la minería de datos se utilizan para encontrar hechos que ocurren en común dentro de un conjunto de datos. Estas reglas son expresiones del tipo Si '$\mathbf{Y} = \mathbf{y}$' entonces '$\mathbf{Z} = \mathbf{z}$', siendo $\mathbf{Y}$ y $\mathbf{Z}$ conjuntos de atributos (valor) que cumplen que $\mathbf{Y} \bigcap \mathbf{Z} = \oslash$ (Agrawal, Imieliñski, y Swami1993).

Existen algunos algoritmos bien conocidos como Apriori de (Srikant y Agrawal1996), Eclat de (Zaki2000) y FP-Tree de (Han et al.2004). El algoritmo, denominado Apriori, fue el primer algoritmo que se usó para la extracción de reglas de asociación de manera satisfactoria. En este algoritmo se extraen los conjuntos de elementos frecuentes y, a partir de estos, se obtiene reglas de asociación. Para reducir el coste computacional, el algoritmo Apriori mantiene que si un conjunto de elementos es frecuente, entonces todos sus subconjuntos son frecuentes. Si un conjunto no es frecuente, sus superconjuntos no serán frecuentes, por lo que se puede reducir el coste computacional eliminando dichos elementos (Luna2014).

2.4. Clasificación no supervisada

`Definición de aprendizaje no supervisado`, dado un conjunto de datos $\mathscr{D} = \{\mathbf{x}^i, i = 1, ..., N\}$ el objetivo es encontrar una descripción compacta de los datos. En este tipo de aprendizaje no hay una variable de predicción especial. Desde el punto de vista probabilístico, el interés es en modelizar la distribución $p(\mathbf{x})$. La probabilidad del modelo para generar los datos es una medida popular de la exactitud de la descripción (Anderson et al.1986).

El aprendizaje no supervisado, se realiza a partir de un conjunto de observaciones que no tienen clases asociadas y la finalidad es detectar regularidades de los datos de cualquier tipo. Son muy usados para la compresión de datos y agrupación.

Algunas técnicas de aprendizaje no supervisado son clustering (agrupan objetos en regiones donde la similitud mutua es elevada), visualización (permite observar el espacio de instancias en un espacio de menor dimensión), reducción de la dimensionalidad (los datos de entrada son agrupados en subespacios de una dimensión más baja que la inicial), extracción de características (construyen nuevos atributos a partir de los muchos atributos originales).

En este trabajo se usará la técnica de clustering para descubrir la estructura en los datos de entrada buscando agrupamientos entre las variables y grupos de casos de forma que cada

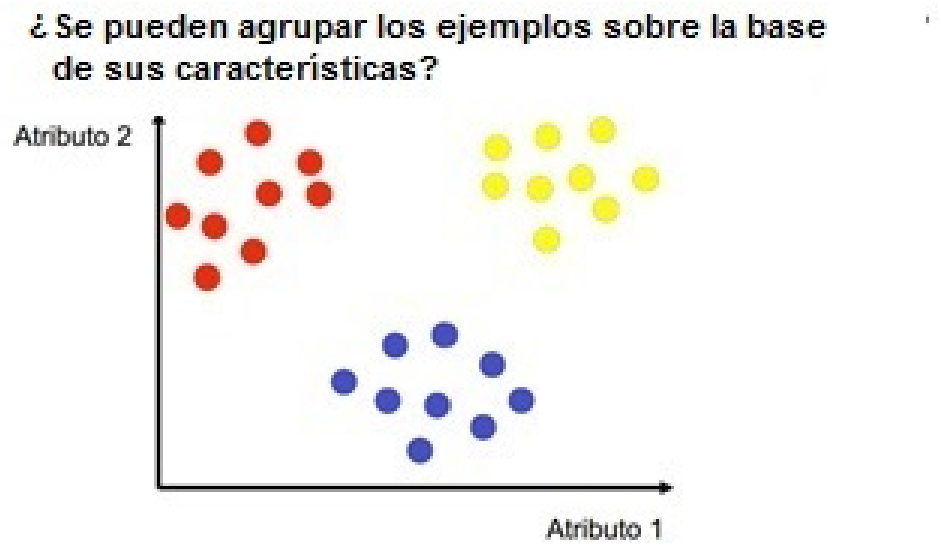

Figura 2.7 Ejemplo de aprendizaje no supervisado

grupo sea homogéneo y distinto de los demás. Hay varios métodos de clustering entre los más conocidos tenemos: jerárquicos (los datos se agrupan de manera arborescente), no jerárquicos (genera particiones a un solo nivel), paramétricos (asume que las densidades condicionales de los grupos tienen cierta forma paramétrica conocida y se reduce a estimar los parámetros), no paramétricos (no asumen nada sobre el modo en el que se agrupan los objetos).

En muchas ocasiones es posible que no se hayan determinado algunas variables que eran importantes para la explicación del fenómeno bajo estudio, y es evidente que existe un influencia en los datos que no es percibida fácilmente, entonces es posible postular la existencia de alguna o algunas variables ocultas como responsables de explicar las relaciones existentes entre los datos (Acosta et al.2014).

Los métodos de aprendizaje de redes genéricas se pueden interpretar como métodos de clasificación no supervisada. sin embargo, es más común considerar que se trata de problemas de aprendizaje donde hay una variable clase C oculta o no observada.

Este tipo de clasificador se da a partir de un conjunto de variables $\mathbf{X}$ y se trata de asignar a cada caso un grupo valor de C cuando las clases son desconocidas, teniendo como objetivo dividir un conjunto de N instancias en k clases, de tal forma que similares instancias se asignen a la misma clase (Cooper y Herskovits1992).

Puede verse una clasificación no supervisada como supervisada donde todos los valores de la variable clase son desconocidos. Esto se interpreta como que se tendría un aprendizaje de datos incompletos. Se pueden determinar varios métodos de aprendizaje, usando redes bayesianas partiendo de datos incompletos (Cheeseman et al.1996).

2.4.1. Clasificación no supervisada con variables ocultas: Autoclass

AutoClass utiliza razonamiento bayesiano dado un conjunto de datos de entrenamiento y sugiere un conjunto de clases posibles. Los agrupamientos naturales obtenidos mediante esta técnica de agrupamiento usando similitud resulta muy útil a la hora de construir clasificadores cuando no están bien definidas las clases. AutoClass es un Naive Bayes con una variable clase oculta y que permite encontrar la hipótesis más probable, dados los datos e información a priori. En este caso de modelo se hace una estimación de máxima verosimilitud o máxima probabilidad a posteriori de los parámetros. Normalmente se busca un balance entre lo bien que se ajustan los datos a las clases y la complejidad de las clases. Para equilibrar estos factores se puede utilizar una métrica como la BIC (Schwarz1978).

2.4.2. Algoritmo EM

Dada una variable clase oculta con número finito de clases, para estimar los parámetros se considera el algoritmo EM (Expectation Maximization) (Dempster, Laird, y Rubin1977), que es un método para encontrar el estimador de máxima verosimilitud de los parámetros de una distribución de probabilidad. Este algoritmo es muy útil cuando hay datos perdidos en los datos de aprendizaje $\mathscr{D}$. En clasificación no supervisada podemos considerar que todos los valores de la variable clase están perdidos. Para encontrar estos parámetros óptimos se deben realizar dos pasos: primero (Esperanza) calcular la esperanza del logaritmo de la verosimilitud con respecto a la información conocida y unos parámetros propuestos, luego (Maximización) maximizar con respecto a los parámetros; estos dos pasos se repiten hasta alcanzar la convergencia (Sierra2006).

Supondremos un conjunto de variables $\mathbf{Z} = (\mathbf{X}; \mathbf{Y})$, donde los datos $\mathbf{X}$ son visibles, pero los datos $\mathbf{Y}$ están ocultos. Tenemos que:

$$p(\mathbf{z}|\Theta) = p(\mathbf{x},\mathbf{y}|\Theta) = p(\mathbf{x}|\mathbf{y},\Theta)p(\mathbf{y}|\Theta) \tag{2.19}$$

Donde Θ es un conjunto de parámetros de la red bayesiana (distribuciones a priori y distribuciones condicionadas). Supongamos que $\mathscr{D} = \{\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^N\}$ que se distribuye según una distribución de probabilidad $p(\mathbf{X}|\Theta)$ y la función de verosimilitud de los parámetros dada la muestra es:

$$L(\Theta|\mathscr{D}) = \prod_{i=1}^{N} p(\mathbf{x}^i|\Theta) \tag{2.20}$$

Y el logaritmo de verosimilitud es:

$$LL(\Theta|\mathscr{D}) = \Sigma_{i=1}^{N} \log p(\mathbf{x}^i|\Theta) \tag{2.21}$$

El problema es que en general tenemos una expresión sencilla para $p(\mathbf{x},\mathbf{y}|\Theta)$, pero no para $p(\mathbf{x}|\Theta)$. Si $\mathbf{Y}$ estuviese observado podríamos calcular $LL(\Theta|\mathbf{X},\mathbf{Y}) = \Sigma_{i=1}^{N} \log p(\mathbf{x},\mathbf{y}|\Theta)$. El algoritmo EM lo que hace es suponer un valor para los parámetros Θ^t y calcular el valor esperado de la verosimilitud $LL(\Theta|\mathbf{X},\mathbf{Y})$ de los parámetros:

$$Q(\Theta,\Theta^t) = \sum_{i=1}^{N} E[\log p(\mathbf{x}^i,\mathbf{y}^\mathbf{i}|\Theta)|\mathbf{x}^i,\Theta^t)] \tag{2.22}$$

Donde Θ^t son los valores de los parámetros actuales (inicialmente un valor arbitrario)

Para encontrar los parámetros óptimos Θ^* a partir de $L(\Theta|\mathbf{X},\mathbf{Y})$, hay que proceder en dos pasos:

- Paso E (Esperanza): Calcular la esperanza de la verosimilitud con respecto a la información conocida y unos parámetros propuestos $\Theta^{(t)}$:

$$Q(\Theta,\Theta^t) = \sum_{n=1}^{N} E[\log p(\mathbf{x}^i,\mathbf{y}^i|\Theta)|\mathbf{x}^i,\Theta^t] \tag{2.23}$$

- Paso M (Maximización): Maximizar Q con respecto a los parámetros:

$$\Theta^{t+1} = \arg\max_{\Theta} Q(\Theta,\Theta^t) \tag{2.24}$$

- Repetir E y M de forma iterativa con Θ^{t+1} en el puesto de Θ^t hasta alcanzar convergencia donde los parámetros se diferencian poco.

En otras palabras, para aprender se debe suponer que hay variables ocultas y que están relacionadas con las otras variables. Esto es muy útil cuando contamos con un conjunto de variables que dependen de una variable no observada.

1. Iniciar los parámetros desconocidos con valores aleatorios
2. Utilizar los datos conocidos con los parámetros actuales para estimar los valores de las variables ocultas

3. Utilizar los valores estimados para completar la tabla de datos
4. Re-estimar los parámetros con los nuevos datos
5. Repetir del paso 2 al 4 hasta que no haya cambios significativos en las probabilidades.

2.5. Conclusiones parciales

En este capítulo hemos realizado una revisión de los modelos gráficos probabilísticos y los métodos de aprendizaje. Hemos distinguido tres enfoques principales:

1. Estimación de la probabilidad conjunta: es encontrar una red genérica que mejor se ajuste a los datos.
2. Clasificación supervisada: cuando existe una variable distinguida llamada clase cuyos valores queremos predecir en función de los otros atributos.
3. Clasificación no supervisada: suponiendo la existencia de una variable oculta no observada, cuyos valores representan los distintos grupos que se supone que existen en la población.

También se ha realizado una breve descripción de otros modelos de clasificación supervisada como árboles de clasificación y reglas de asociación que serán usados en otros capítulos de esta memoria.

Parte II

Contribuciones

3 Agrupamiento jerárquico

En este capítulo se presenta una propuesta para aplicar un método de clúster jerárquico para analizar datos en los que se suponga que las relaciones existentes entre los mismos se deban a la existencia de variables ocultas. Este suele ser el caso de muchos conjuntos de datos entre los que están los datos educativos (socioeconómicos, aprobación y deserción) de los estudiantes legalmente matriculados en el periodo 2012-2013 de la Facultad de Ciencias de la Ingeniería de la Universidad Técnica Estatal de Quevedo; se propone un nuevo método basado en clasificación no supervisada para la construcción de una jerarquía de variables artificiales a partir de las variables observadas $\mathbf{X} = \{X_1, \ldots, X_n\}$. La idea básica es similar a AUTOCLASS pero en lugar de usar todas las variables, primero hacemos una clasificación de las variables observadas por grupos que tengan un alto grado de dependencia entre sí. Por cada grupo de variables se estima una variable oculta con AUTOCLASS y se repite el proceso con las variables ocultas introducidas. Se obtiene así una jerarquía de variables ocultas que determinan las relaciones entre las variables observadas de manera simple. El resultado final será una red bayesiana $\mathscr{B}$, que inicialmente contiene variables $\mathbf{X}$ y se completa con las variables ocultas (Oviedo, Moral, y Puris2016).

En el capítulo anterior se describen los aspectos fundamentales del aprendizaje bayesiano, incluyendo el aprendizaje no supervisado muy usado para la compresión de datos y agrupación. Se considera que el agrupamiento o 'clúster' es el hecho de clasificar a los elementos de un conjunto de observaciones formando grupos con ellas, de manera tal, que los individuos dentro de cada grupo sean similares entre sí. Se supone que se tiene un conjunto de datos $\mathscr{D} = \{\mathbf{x}^1, ..., \mathbf{x}^n\}$, y hay que dividirlos categorías de tal modo que los casos dentro de un mismo grupo que son más parecidas entre sí que a los casos de otros grupos. Estos casos 'parecidos' son consideradas con una noción de similitud o distancia entre muestras.

El procedimiento de este capítulo se puede considerar como un método de agrupamiento multidimensional donde cada variable oculta muestra una forma de agrupar los datos. Ante un nuevo caso, se puede obtener mediante algoritmos de propagación, la probabilidad de pertenecer a cada uno de los grupos asociados a los valores de las distintas variables ocultas.

El trabajo está estructurado de la siguiente manera: la primera sección nos permite determinar las variables de agrupamiento, luego se indica la manera de adicionar variables artificiales

ocultas para llegar a aplicar un cálculo recursivo. Posteriormente en la sección 3.4 se realiza un estudio de estimación de parámetros y optimización de los números de casos de las variables artificiales. En la sección 3.5 se realiza un estudio experimental entres direcciones:

- Comparación con otros algoritmos de aprendizaje con bases de datos de la UCI.
- Datos de deserción de la facultad de Ciencias de la Ingeniería de la Universidad Técnica Estatal de Quevedo en el periodo 2012-2013.
- Basado en datos de evaluación de encuestas realizadas por los estudiantes.

Finalmente se indican las conclusiones parciales.

3.1. Variables de agrupamiento

Se supone un conjunto de variables $X = (X_1, ..., X_n)$ y un conjunto de observaciones $\mathscr{D} = \mathbf{x}_1, ..., \mathbf{x}_N$. Se realiza el cálculo de una matriz A, $n \times n$, donde a_{ij} representa el grado de dependencia de las variables X_i y X_j. Este grado de dependencia es calculado con una métrica BDEu. Si $\mathscr{D}$ es el conjunto de datos observados, se tiene:

$$a_{ij} = Dep(X_i, X_j|\mathscr{D}) = Score(X_i, \{X_j\}|\mathscr{D}) - Score(X_i, \emptyset|\mathscr{D}) \tag{3.1}$$

Donde *Score* es calculado con BDEu dado un parámetro S. Conocidas las propiedades de BDEu (grafos equivalentes que tienen igual score), tenemos que $a_{ij} = a_{ji}$; es decir, la matriz A es simétrica. Cuando $i = j$, podemos considerar que $Score(X_i, \{X_i\}|\mathscr{D}) = 0$, y $Dep(X_i, X_i|\mathscr{D}) = -Score(X_i, \emptyset|\mathscr{D})$. Sin embargo, este valor no es importante para nosotros y no va a ser calculado, ajustando $a_{ii} = 0$.

Luego, definimos la siguiente relación en $\mathbf{X}$:

$X_i \to X_j$ si y solo si $X_j = \arg\max_k a_{ik}$, $i \neq j$, y $a_{ij} > 0$.

es decir cada variable X_i apunta a las variables X_j con un mayor grado de dependencia con X_i, dado que este grado de dependencia sea mayor que 0.

Ahora, está relación es ampliada para ser una relación simétrica:

$X_i \leftrightarrow X_j$ si y solo si $X_i \to X_j$ ó $X_j \to X_i$.

Finalmente, esta relación se extiende para ser transitiva y reflexiva:

$X_i \equiv X_j$ si y solo si existe una secuencia finita $X_{k_1}, \ldots X_{k_m}$, tal que $X_i = X_{k_1}, X_j = X_{k_l}$ y $X_{k_{l-1}} \leftrightarrow X_{k_l}, \forall\ 2 \leq l \leq m$.

Construimos entonces una partición del conjunto de variables a partir de las clases de equivalencia de la relación ($\equiv$). Esta partición verifica que cada grupo de variables $\mathbf{C_l}$ es el más pequeño conjunto no vacío de variables tal que cualquier variable $X_i \in \mathbf{C_l}$, y la variable X_j con mayor grado de dependencia con X_i está en el mismo grupo ($X_j \in \mathbf{C_l}$) si $a_{ij} > 0$.

Para calcular esta partición del conjunto de las variables $\mathbf{X}$ podemos usar un procedimiento sencillo que empiezan con una partición $\mathscr{P} = \{\{X_1\},\ldots,\{X_n\}\}$, es decir, en la partición inicial los grupos de variables solo contiene una sola variable.

A continuación, dos grupos $\mathbf{C_1}$ y $\mathbf{C_2}$ son unidos, si hay una variable $X_i \in \mathbf{C_1}$ de manera tal que la variable con mayor grado de dependencia con X_j está en el otro grupo $\mathbf{C_2}$ y $a_{ij} > 0$.

Mayores detalles se pueden visualizar en el Algoritmo 3.1.

```
1:  Conjunto 𝒫 ← {{X_1},...,{X_n}}
2:  for all par de variables X_i, X_j (i ≠j) do
3:     if a_ij = máx_{k≠j} a_ik y a_ij > 0 then
4:        Sea C_1 ∈ 𝒫, tal que X_i ∈ C_1
5:        Sea C_2 ∈ 𝒫, tal que X_j ∈ C_2
6:        if C_1 ≠ C_2 then
7:           C ← C_1 ∪ C_2
8:           𝒫 ← (𝒫 ∪ {C}) \ {C_1, C_2}
9:        end if
10:    end if
11: end for
12: return 𝒫
```

Algoritmo 3.1: Calculando grupo de variables

3.2. Adición de variables artificiales ocultas

Se comienza con una red bayesiana en la que se añaden las variables iniciales X_i, totalmente desconectadas entre si. Toda vez que la partición $\mathscr{P}$ ha sido calculada, consideramos una variable Y_i para cada grupo $\mathbf{C_i} \in \mathscr{P}$ si el número de elementos en $\mathbf{C_i}$ es mayor que 1: $|\mathbf{C_i}| > 1$. Esas variables Y_i asociadas a los grupos $\mathbf{C_i}$ son añadidas a la red bayesiana $\mathscr{B}$ y se añade un arco desde Y_i a cada variable X_j tal que $X_j \in \mathbf{C_i}$. Y_i es una variable oculta con observaciones no reales. La idea es que Y_i contenga un valor por cada agrupamiento o grupo de individuos tomando en cuenta solo variables en $\mathbf{C_i}$. En lugar de hacer solo una clasificación como en AutoClass, se hace una clasificación diferente por cada grupo de variables $\mathbf{C_i} \in \mathscr{P}$.

Para asignar un número inicial de casos variables a variable Y_i, primero calculamos una variable representativa en $\mathbf{C_i}$. Para hacer esto, calculamos para cada variable $X_j \in \mathbf{C_i}$ el valor $c_j = \sum_{X_k \in C_i} a_{jk}$ y seleccionamos la variable $X_{i_j} \in \mathbf{C_i}$ con mayor c_j entre todas las variables en $\mathbf{C_i}$. La idea es que una variable representativa sea una variable con el más alto grado de dependencia que el resto de variables en el grupo. Una vez que, X_{i_j} sea calculado se asigna a Y_i un número de casos igual al número de casos de X_{i_j}.

3.3. Cálculo recursivo

Si el número de variables artificiales Y_i es mayor que uno, todo se repite de nuevo (agrupamiento de variables y adición de variables artificiales), pero usando el conjunto de variables ocultas $\mathbf{Y} = \{Y_i : \mathbf{C_i} \in \mathscr{P}, |\mathbf{C_i}| > 1\}$ en lugar del conjunto inicial de variables $\mathbf{X}$. Para aplicar el procedimiento indicado arriba, necesitamos determinar una nueva matriz A' de dependencia entre las variables en $\mathbf{Y}$. Asumimos de que como Y_i resume la información común de las variables en el grupo $\mathbf{C_i}$, la dependencia entre Y_i y Y_j es estimada como el promedio de las dependencias entre las variables en $\mathbf{C_i}$ y las variables en $\mathbf{C_j}$. A saber,

$$a'_{ij} = \frac{1}{|\mathbf{C_i}| \cdot |\mathbf{C_j}|} \sum_{X_l \in \mathbf{C_i}, X_k \in \mathbf{C_j}} a_{lk} \tag{3.2}$$

Con las variables $\mathbf{Y}$ y la matriz A' repetimos los pasos de agrupamiento de variables y de adición de variables artificiales hasta que el número de variables artificiales sea menor o igual que 1. Cada vez que una variable artificial es añadida, un arco es incorporado desde la variable artificial a cada una de las variables en el grupo asociado.

3.4. Estimación de parámetros y optimización de casos

Se supone ahora que se tiene calculada la estructura de la red bayesiana con las variables originales y artificiales, lo cual en general es un árbol o un bosque de árboles (si algún grupo solo contiene una variable, la variable artificial no será añadida y esta parte estará desconectada del resto). Luego, debemos estimar los parámetros y optimizar el número de casos de cada variable artificial. Se asume que $\mathbf{Z} = \mathbf{X} \cup \mathbf{Y}$ es el conjunto de variables observadas $\mathbf{X}$ y variables artificiales $\mathbf{Y}$.

Para cada probabilidad condicional $P(Z_i|\pi_{ik})$, donde π_{ik} es la k-th configuración de los padres de Z_i, Π_i, consideramos el vector de parámetros $\Theta_{ik} = (\theta_{i1k}, \ldots, \theta_{ir_ik})$ donde $\theta_{ijk} = P(Z_i = z_i^j|\pi_{ik})$ y asumimos que este vector de parámetros de cada probabilidad condicional $P(Z_i|\pi_{ik})$ sigue una distribución Dirichlet $D(1, \ldots, 1)$, lo que implica que los parámetros son estimados con la corrección de Laplace. Luego, aplicamos el algoritmo EM para maximizar la probabilidad a posteriori dados los datos. Para esto, se necesita una estimación inicial de parámetros. Teniendo en cuenta que EM garantiza una convergencia a un máximo local, la calidad de los parámetros iniciales pueden tener una influencia en la calidad de los valores finales, (Oviedo, Moral, y Puris2016).

Por lo expuesto se propone un procedimiento basado en la asignación de una variable observada a cada variable artificial. Esto se hace de una manera recursiva. La primera vez que los grupos son calculados y las variables artificiales son añadidas, cada variable Y_i es asociada con

un grupo de variables observadas C_i, luego Y_i es asociada con la variable representativa X_{i_j} de este grupo tal como fue calculado en la Subsección 3.2.

En los siguientes pasos, los grupos C_i serán compuestos de variables artificiales (cada una de ellas con una variable observada asociada). A continuación, se procede de manera similar que en la etapa inicial con la única diferencia de que la variable Y_i es añadida para el grupo C_i, se tendrá como variable observada asociada la misma variable observada que está asociada al grupo C_i.

Ahora, para dar una estimación inicial de los parámetros de $P(Z_i|\pi_k)$ calculamos para cada valor z_i^j de Z_i el número de ocurrencias N_{ijk} en los datos $\mathscr{D}$, donde cada variable artificial es reemplazada por estas variables observadas asociadas (de esta manera las frecuencias pueden ser calculadas). Toda vez que estas frecuencias se calculan, la estimación inicial es

$$\theta_{ijk}^0 = \frac{N_{ijk} + \log(N+1) + (R_{ijk})}{\sum_j (N_{ijk} + \log(N+1) + (R_{ijk}))}$$

.

dónde R_{ijk} es un número aleatorio entre 0 y 1. De esta manera, asumimos que la variable artificial asociada a un grupo de variables tiene un comportamiento similar a la variable más significativa en el grupo, pero se añade un factor aleatorio (para evitar convergencias debido a idénticos parámetros iniciales para diferentes valores de las variables) y un factor de smoothing ($\log(N+1)$) similar a la corrección de Laplace pero aumentando por el tamaño de la muestra, para evitar iniciar el algoritmo con parámetros iniciales demasiado extremos (muy cerca a 0 ó 1).

Una vez que tenemos una primera estimación de los parámetros, procedemos con el algoritmo EM para la optimización de probabilidades subsiguientes con corrección de Laplace. Los dos pasos para mejorar un conjunto de parámetros Θ^t son los siguientes:

- **Esperanza:** Calculamos $N_{ijk}^t = E[N_{ijk}|\Theta^t] = \sum_{\mathbf{x}\in\mathscr{D}} P(Z_i = z_i^j, \Pi_i = \pi_{ik}|\mathbf{X} = \mathbf{x})$.
  Estas probabilidades condicionales son calculadas mediante propagación en redes bayesianas con el conjunto de parámetros actuales Θ^t.
- **Maximización:** Actualiza el vector de parámetros a nuevos valores maximizando la probabilidad 'a posteriori'.

$$\theta_{ijk}^{t+1} = \frac{N_{ijk}^t + 1}{\sum_{j=1}^{r_i} (N_{ijk}^t + 1)}$$

Estos dos pasos son repetidos hasta que la diferencia $|\theta_{ijk}^{t+1} - \theta_{ijk}^t|$ sea menor o igual que un umbral dado ε para todos los parámetros.

Finalmente, pasamos a optimizar el número de casos para cada variable artificial Y_l. Para hacer esto, consideramos la métrica BIC, (Schwarz1978) o la de Akaike (Akaike1974). En este caso, los scores son calculados como en las ecuaciones 2.5 y 2.7, donde $\sum_{i=1}^{n}\sum_{j=1}^{r_i}\sum_{k=1}^{s_i} N_{ijk}\log\frac{N_{ijk}}{N_{ik}}$ es reemplazado por $\sum_{i=1}^{n}\sum_{j=1}^{r_i}\sum_{k=1}^{s_i} N_{ijk}\log\theta_{ijk}$ y θ_{ijk} son los parámetros estimados con el

algoritmo EM.

Para hacer esto, tratamos de ampliar el número de casos de variable Y_l mientras haya mejora en la métrica considerada. Si no se ha obtenido mejora mediante el incremento de números de casos de Y_i, entonces se tratará de optimizar la métrica BIC o Akaike reduciendo el número de casos.

Es importante remarcar que cada vez que el número de casos de una variable artificial cambia, hay que recalcular los parámetros óptimos usando el algoritmo EM con el fin de calcular los scores de BIC y Akaike. De nuevo, los parámetros iniciales pueden ser importantes para una rápida convergencia.

Si se cambia el número de casos de la variable Y_l, con el fin de seleccionar los parámetros iniciales por cada probabilidad condicional $\theta_{ijk} = P(Z_i = z_i^k | \pi_{ij})$, vamos a seguir las siguientes reglas:

- Si Y_l no está implicado en $P(Z_i|\Pi_i)$, es decir no es Z_i y no está incluida en Π_i, entonces los parámetros iniciales son los obtenidos en aplicaciones previas de el algoritmo EM antes de cambiar el número de estados de Y_l.
- Si Y_l es Z_i, el número de casos de Z_i cambia, pero no el número de configuraciones de Π_i, entonces
  - Si el número de casos de Z_i incrementa, es decir cambia desde $\{z_i^1, \ldots, z_i^{r_i}\}$ a $\{z_i^1, \ldots, z_i^{r_{i+1}}\}$, los parámetros iniciales son calculados como:

$$\theta_{ijk}^0 = \begin{cases} \theta_{ijk} \frac{r_i}{r_i + 2R_{ik}} & \text{if } j \leq r_i \\ \frac{2R_{ij}}{r_i + 2R_{ik}} & \text{if } j = r_i + 1 \end{cases}$$

    Donde R_{ij} es un número aleatorio entre 0 y 1 y θ_{ijk} son los parámetros previos antes de añadir un nuevo valor. La idea es que el nuevo valor $z_i^{r_i+1}$ tiene un comportamiento aleatorio, y los otros valores de Z_i tienen probabilidades similares a los antiguos r_i valores de Z_i. Con esta evaluación tenemos que si las probabilidades Z_i se condicionan al hecho de que Z_i toma uno de sus antiguos valores, entonces esas probabilidades son las mismas que había antes de aumentar el número de valores de Z_i.
  - Si el número de casos de Z_i decrece, es decir cambia de $\{z_i^1, \ldots, z_i^{r_i}\}$ a $\{z_i^1, \ldots, z_i^{r_i-1}\}$, los parámetros iniciales son calculados como:

$$\theta_{ijk}^0 = \begin{cases} \theta_{ijk} & \text{if } j < r_i - 1 \\ \theta_{ijk} + \theta_{i(j+1)k} & \text{if } j = r_i - 1 \end{cases}$$

    donde θ_{ijk} son los parámetros previos antes de eliminar el nuevo valor. La idea es que el nuevo valor , $z_i^{r_i-1}$ desempeñe el papel de unión de los valores antiguos , $z_i^{r_i-1}$ y $z_i^{r_i}$, por lo que la probabilidad del nuevo valor es la adición de las probabilidades previas de los dos elementos.

- Si Y_l es diferente de Z_i y $Y_l = \Pi_i$, es decir Y_l es el único padre de Z_i (como la red es un árbol, si Y_l aparece en el conjunto de padres, este será el único padre),

  - Si el número de casos de Y_l incrementa, es decir cambia de $\{y_l^1, \ldots, y_l^{r_l}\}$ a $\{y_l^1, \ldots, y_r^{r_l+1}\}$, entonces los parámetross iniciales son calculados como:

$$\theta_{ijk}^0 = \begin{cases} \theta_{ijk} & \text{if } k \leq r_l \\ \frac{0.5\frac{\sum_{k=0}^{r_l}\theta_{ijk}}{r_l} + R_j}{0.5+\sum_j R_j} & \text{if } k = r_l + 1 \end{cases}$$

    donde R_j es un número aleatorio entre 0 y 1 y θ_{ijk} son los parámetros previos antes de añadir un nuevo valor. La idea es que cuando se condicione a los valores antiguos de Y_l, la probabilidad condicional no cambia, y cuando se condicione a los nuevos valores $y_l^{r_l+1}$, la probabilidad condicional sea proporcional al promedio de las probabilidades condicionales más un factor aleatorio (el denominador $(0.5 + \sum_j R_j)$ es un factor de normalización para que las probabilidades sumen 1 cuando se suman en j).
  - Si el número de casos de Y_l decrece, es decir cambia de $\{y_l^1, \ldots, y_l^{r_l}\}$ a $\{y_l^1, \ldots, y_l^{r_l-1}\}$, entonces los parámetros iniciales son calculados como:

$$\theta_{ijk}^0 = \begin{cases} \theta_{ijk} & \text{if } k < r_l - 1 \\ 0.5(\theta_{ijk} + \theta_{ij(k+1)}) & \text{if } k = r_l - 1 \end{cases}$$

    donde θ_{ijk} son los parámetros previos antes de la eliminación del nuevo valor. Se continúa con la misma idea, esto es que cuando decrezca el número de valores: $y_l^{r_l-1}$ desempeñe el papel de la unión de los valores antiguos $y_i^{r_l-1}$ y $y_l^{r_l}$, así que cuando se condicione a los nuevos valores, se calcule la probabilidad promedio del condicionamiento para los dos valores antiguos .

La optimización de los valores de las variables artificiales, se realiza variable por variable, así es posible que si Y_i es optimizado después que Y_j, después de optimizar Y_i, el número de valores óptimos de Y_j puede cambiar, entonces el proceso de optimización de los números de valores de una variable debería repetirse mientras que haya cambios en el número de casos de el resto de las variables. Sin embargo, en la práctica nos hemos dado cuenta de que esto rara vez ocurre.

Cuando decrece el número de casos de una variable, podría ser el caso de que el número óptimo de casos sea 1. Una variable con solamente un caso no influye en el resto de variables y es equivalente a la eliminación de ella y de los enlaces con las otras variables.

3.5. Análisis experimental comparativo

En esta sección llevamos a cabo un estudio experimental para probar el rendimiento del procedimiento propuesto usando el programa Elvira (Elvira2002). Hemos llevado a cabo tres experimentos. En todos comparamos nuestro procedimiento de agrupación con otros algoritmos de aprendizaje como K2, PC, Naive Bayes (la última variable es la clase) y Autoclass. En el primero de ellos, tratamos de evaluar las capacidades de nuestro procedimiento para representar una distribución de probabilidad conjunta. Por lo que se compara el rendimiento de los diferentes algoritmos en una base de datos estándar de la UCI (Lichman2013). En el segundo, analizamos los datos de deserción estudiantil e indicadores socio-económicos de los estudiantes legalmente matriculados en la Facultad de Ciencias de la Ingeniería de la Universidad Técnica Estatal de Quevedo (Ecuador) en el periodo 2012-2013. En el tercer experimento hemos llevado a cabo un estudio similar al de los datos de deserción estudiantil, pero ahora con la base de datos de encuestas del profesorado por estudiantes de la UCI. La comparación está hecha con una validación cruzada de 10 tandas.

En virtud de que el objetivo es representar la probabilidad conjunta de las variables en el problema, para cada caso en los datos de prueba, se ha tenido que medir el logaritmo de la probabilidad de el nuevo caso asignado por el modelo de red bayesiana aprendida. Finalmente se presenta la suma de estos valores en todos los casos de prueba.

3.5.1. Comparación con la base de datos de la UCI

En esta sección se presentan los resultados de las probabilidades calculadas en la base de datos de la UCI con una validación cruzada de 10 tandas. Las bases se presentan en la Tabla 3.1, donde se describen sus características. AClass *i* es el procedimiento Autoclass con *i* valores de la variable clase. HNBayesEM A es nuestro procedimiento jerárquico con una métrica Akaike para optimizar el número de casos de las variables y HNBayesEM B corresponde al uso de la métrica BIC.

Se ha llevado a cabo una prueba de Friedman no paramétrico en el que hay diferencias significativas entre los procedimientos. El análisis post hoc presenta diferencias significativas entre PC y AClass 2, AClass 5, HNBayesEM A y HBayesEM B (muy significativo) y entre PC y NaiveBayes (moderada significancia). PC es el algoritmo con peor rendimiento de aprendizaje.

Los resultados obtenidos al aplicar el test de Friedman, indican que AClass5 es el que mejor valor de agrupación entrega; pero, muy de cerca nuestro método con las métricas tanto Akaike y luego BIC entregan excelentes resultados. El de peor ranking es el algoritmo PC. Esto es con un P-valor = $4.96E-6$.

Tabla 3.1 Log-probabilidad con validación cruzada =10

Datasets	PC	K2	NaiveBayes	AClass 2	AClass 5	HNBayesEM A	HBayesEM B
mammographic	-8440.69	-7899.52	-7858.38	-7649.42	-7615.95	-7885.965	-7898.26
Lung-cancer	-1440.19	-1948.59	-1489.05	-1418.50	-1410.84	-1366.07	-1366.38
hepatitis	-4262.48	-4567.96	-4358.94	-3606.53	-3558.61	-3603.19	-3608.51
e colic	-5300.78	-4898.12	-5008.56	-3470.97	-3240.55	-3450.27	-3452.84
breast-cancer-w	-18404.49	-16680.98	-16317.35	-12871.11	-12694.05	-12922.98	-13046.08
contact-lenses	-107.90	-95.83	-101.19	-104.58	-104.13	-102.51	-102.50
hayes-roth-m	-1974.11	-1984.55	-1918.73	-1442.43	-1393.92	-1431.70	-1433.62
Monk 1	-1947.65	-1422.21	-1920.35	-1466.90	-1454.540	-1466.74	-1466.93
vote	-4918.90	-3328.22	-3768.07	-3497.17	-3371.00	-3305.59	-3338.69
Balance-scale	-4616.48	-4649.34	-4419.63	-4426.84	-4434.59	-4424.79	-4424.75
tic-tac-toe	-9827.24	-9324.91	-9738.50	-9761.78	-9543.61	-9415.01	-9539.52
iris	-1622.46	-1369.90	-1255.81	-1365.65	-1270.99	-1291.65	-1371.01
labor	-650.00	-661.86	-626.98	-630.17	-602.03	-600.86	-612.31
soybean	-922.11	-600.63	-528.67	-810.13	-584.90	-606.99	-659.44

3.5.2. Datos de deserción

Este conjunto de datos contiene variables académicas y socio económicas medidas en 773 estudiantes de la Facultad de Ciencias de la Ingeniería de la Universidad Técnica Estatal de Quevedo (Ecuador) durante el periodo académico 2012/13. Las variables medidas pueden ser observadas en la tabla 3.2. La variable carrera (*A*) tiene como valores las diferentes titulaciones que se pueden tomar en esta Facultad: Ingeniería en Sistemas, ingeniería en Diseño Gráfico, Ingeniería Mecánica, Ingeniería Industrial, Ingeniería en Telemática, Ingeniería Eléctrica, Ingeniería Agroindustrial, e Ingeniería en Seguridad Industrial y Salud Ocupacional. La variable *R* (aprobado) determina si el estudiante aprobó el año académico anterior y la variable *S* (deserción) determina si el estudiante abandona el curso.

Tabla 3.2 Variables y sus descripciones

Variable	Descripción
A	Carrera
B	Curso
D	Discapacidad
E	Costo de la educación
F	Vive separado de la familia
G	Tipo de vivienda de la familia
H	Propietario de la vivienda
I	Servicio de TV Cable
J	Servicio de tarjeta de crédito
K	Servicio de Acceso a Internet
L	Servicios Básicos
M	Servicio de transporte privado
N	Servicio de plan celular
O	Servicio de carro propio
P	Viene en carro propio
Q	Trabaja actualmente
R	Aprobó
S	Desertó

Las variables continuas han sido discretizadas en intervalos significativos. Por ejemplo, la variable costo de la educación representada por la letra E ha sido discretizada en valores por debajo de 200 dólares, de 200 - 800, y más de 800.

Los resultados de verosimilitud con validación cruzada igual a 10 tandas se encuentran en la tabla 3.3. Los mejores resultados son obtenidos con el algoritmo K2. Nuestro procedimiento obtiene resultados similares, especialmente con la métrica Akaike. El rendimiento de otros procedimientos es más pobre.

Adicionalmente, nuestro procedimiento tiene una ventaja adicional sobre K2 y otros procedimientos de aprendizaje de una red bayesiana genérica. En nuestro caso, el grafo es siempre un árbol (o bosque de árboles) haciendo que la interpretación y los cálculos sean más fáciles. Al ver la figura 3.1 podemos visualizar la red aprendida con el algoritmo K2 usando toda la base de datos y en la figura 3.2 la red aprendida con nuestro procedimiento de agrupamiento jerárquico con la métrica Akaike.

Podemos ver que en ambos métodos se determina igual conjunto de variables que son dependientes de S (deserción) y estas son R (aprobado), A (carrera), B (curso), y E (costo de la educación). Pero en el algoritmo K2 todas las dependencias desaparecen cuando condicionamos en R y esta es la variable que afecta directamente a S. Sin embargo nuestro modelo, aún siendo muy simple, es capaz de representar que dado R, el costo de la educación está afectando S a través de la variable $AuxNode5$. Los costos más altos implican un crecimiento en la probabilidad de deserción, incluso si R es conocido. Por ejemplo, en nuestro modelo, conociendo que un estudiante no aprueba, la probabilidad de deserción va desde 0.19 a 0.34, dependiendo el costo de la educación. La carrera en la que está matriculado el estudiante (variable A) tiene una influencia en la probabilidad de deserción. Sin embargo, esta influencia es más débil (ya que pasa por las variables ocultas $AuxNode0, AuxNode7, AuxNode5$). Incluso si sabemos que el estudiante aprueba, esta variable tiene una influencia sobre la deserción, aún siendo débil desde el punto de vista cuantitativo.

Por otro lado, las relaciones entre las otras variables son resumidas mediante el uso de variables auxiliares ocultas. Por ejemplo, $AuxNode4$ considera las variables relacionadas con tener vehículo propio. $AuxNode3$ resume todos los servicios relacionados con tecnología. Finalmente, todas las variables socio-económicas son resumidas en una variable $AuxNode6$. Las relaciones de estas variables en el algoritmo K2 son más complejas y difíciles de interpretar. Aunque hay una tendencia a tener enlaces directos entre las variables del mismo grupo, la forma en que se relacionan las variables es menos sencilla.

Se destaca también que AutoClass solo introduce una variable oculta que es el padre de todas las variables observadas. Para problemas como éste, en el que se tiene diferentes grupos de variables, esto no es apropiado ya que obliga a que todas las variables tengan una fuerte relación entre ellas: por ejemplo, si P y O tienen un alto grado de dependencia, como esta dependencia se obtiene solamente a través de la variable oculta, ambas tendrán un alto grado de dependencia con la variable oculta, y entonces tendrán dependencia con todas las otras variables que son

Tabla 3.3 Logaritmo de verosimilitud con validación cruzada=10. Datos de deserción.

PC	K2	NBayes	AClass 2	AClass 5	HNBayesEM A	HBayesEM B
-8832.76	-8403.18	-8780.48	-8660.89	-8607.18	-8452.37	-8494.48

dependientes de la clase oculta. Esto podría explicar los malos resultados de AutoClass en los experimentos (ver tabla 3.3). Una jerarquía de variables ocultas parece ser más apropiado en este caso.

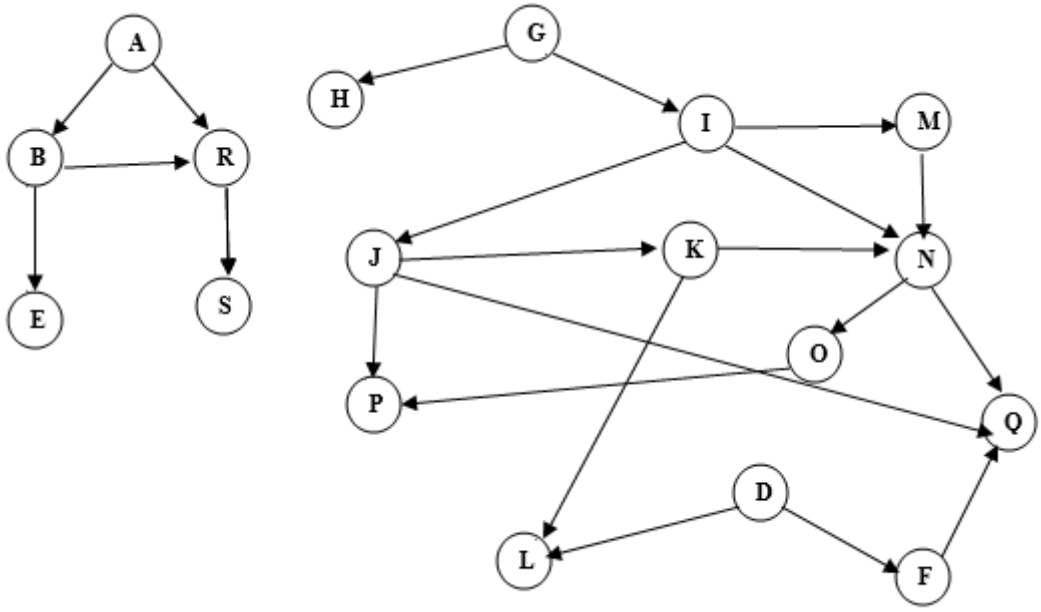

Figura 3.1 Red bayesiana con K2

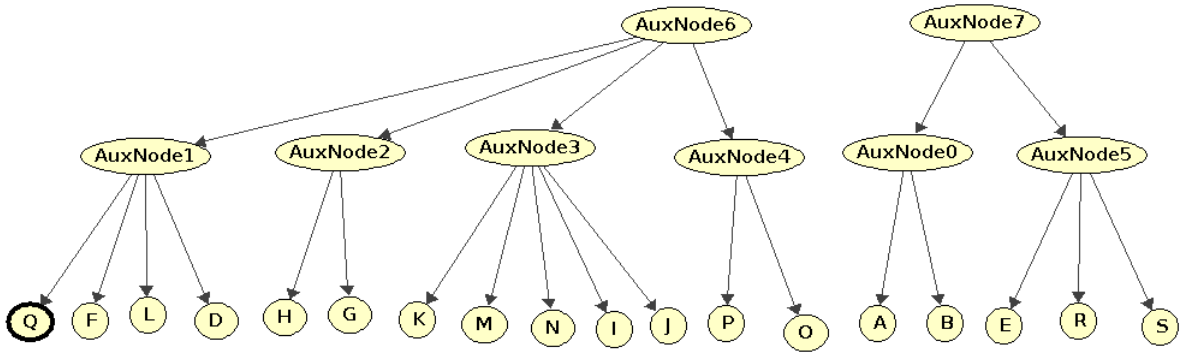

Figura 3.2 Redes bayesianas con agrupamiento jerárquico

3.5.3. Evaluación estudiantil

En este caso usamos el conjunto de datos de la UCI *Turkiye Student Evaluation Data Set* (Gunduz y Fokoue2013) con los resultados de evaluaciones de los estudiantes a los docentes

Conjunto de datos	PC	K2	NaiveBayes	AClass 2	AClass 3
mammographic	-8440.69	-7899.52	-7858.38	-7649.42	-7603.67
Lung-cancer	-1440.19	-1948.59	-1489.05	-1418.50	-1399.04
hepatitis	-4262.48	-4567.96	-4358.94	-3606.53	-3558.56
e colic	-5300.78	-4898.12	-5008.56	-3470.97	-3277.58
breast-cancer-w	-18404.49	-16680.98	-16317.35	-12871.11	-12729.23
contact-lenses	-107.90	-95.83	-101.19	-104.58	-103.24
hayes-roth-m	-1974.11	-1984.55	-1918.73	-1442.43	-1412.88
Monk 1	-1947.65	-1422.21	-1920.35	-1466.90	-1463.60
vote	-4918.90	-3328.22	-3768.07	-3497.17	-3383.74
Balance-scale	-4616.48	-4649.34	-4419.63	-4426.84	-4424.61
tic-tac-toe	-9827.24	-9324.91	-9738.50	-9761.78	-9721.77
iris	-1622.46	-1369.90	-1255.81	-1365.65	-1274.50
labor	-650.00	-661.86	-626.98	-630.17	-631.58
soybean	-922.11	-600.63	-528.67	-810.13	-633.24

Conjunto de datos	AClass 4	AClass 5	AClass 6	AClass 7	AClass 8
mammographic	-7611.65	-7617.36	-7611.28	-7615.95	-7613.49
Lung-cancer	-1414.33	-1410.84	-1405.98	-1406.40	-1407.43
hepatitis	-3551.14	-3558.61	-3546.86	-3536.56	-3534.93
e colic	-3247.57	-3240.55	-3237.93	-3233.11	-3233.22
breast-cancer-w	-12717.54	-12694.05	-12688.22	-12715.24	-12716.71
contact-lenses	-103.65	-104.13	-104.07	-104.05	-104.53
hayes-roth-m	-1405.27	-1393.92	-1394.02	-1394.26	-1395.74
Monk 1	-1461.11	-1454.540	-1443.35	-1439.98	-1435.94
vote	-3378.19	-3371.00	-3370.65	-3370.50	-3369.67
Balance-scale	-4435.12	-4434.59	-4445.59	-4445.72	-4442.96
tic-tac-toe	-9608.40	-9543.61	-9504.35	-9504.59	-9505.37
iris	-1273.41	-1270.99	-1280.77	-1282.40	-1283.56
labor	-600.79	-602.03	-605.43	-605.71	-609.06
soybean	-583.90	-584.90	-585.89	-588.27	-587.79

Conjunto de datos	AClass 9	AClass 10	HNBayesEM 0	HBayesEM 1
mammographic	-7619.95	-7620.33	-7885.965	-7898.26
Lung-cancer	-1407.74	-1406.64	-1366.07	-1366.38
hepatitis	-3533.93	-3535.88	-3603.19	-3608.51
e colic	-3233.27	-3233.32	-3450.27	-3452.84
breast-cancer-w	-12713.04	-12713.92	-12922.98	-13046.08
contact-lenses	-104.23	-104.09	-102.51	-102.50
hayes-roth-m	-1394.29	-1394.40	-1431.70	-1433.62
Monk 1	-1435.69	-1436.56	-1466.74	-1466.93
vote	-3359.71	-3359.97	-3305.59	-3338.69
Balance-scale	-4439.01	-4435.03	-4424.79	-4424.75
tic-tac-toe	-9511.65	-9512.17	-9415.01	-9539.52
iris	-1284.58	-1288.82	-1291.65	-1371.01
labor	-613.68	-613.02	-600.86	-612.31
soybean	-590.13	-591.02	-606.99	-659.44

de la universidad de Gazi en Ankara (Turquía). Las variables están descritas en la tabla 3.4. Está claro que todas las variables están relacionadas y que podemos considerar que miden diferentes aspectos del grado general de satisfacción con el curso y el instructor. Si tratamos de aprender una red bayesiana con los procedimientos de aprendizaje clásicos (Neapolitan2004), obtendremos un gráfico denso y complejo. AutoClass trata de evitar esto, por considerarlas como variables ocultas que codifica este grado de satisfacción.

Sin embargo, AutoClass construye un simple modelo Naive Bayes en el que hay un arco desde la variable oculta a cada una de las variables observadas. Esto pone todas las variables iniciales en el mismo nivel. Pero, viendo estas variables, se puede determinar que hay grupos de variables que tienen relaciones más fuertes entre ellos que con otras variables en otro grupo. Por ejemplo, las variables Q1 y Q2 miden aspectos relacionados con la información dada en el inicio, y las variables Q13 y Q14 miden la preparación de las clases por el instructor. Entonces, en este modelo debería haber una relación más fuerte entre los pares Q1, Q2 y Q13, Q14, que entre las variables de diferentes pares. Esto no es posible en AutoClass y esta es la razón de introducir nuestro nuevo procedimiento de agrupamiento jerárquico. El mismo que consiste en hacer una primera agrupación de variables tratando de encontrar grupos de variables que tienen fuertes relaciones con las variables del mismo grupo. En este caso, por ejemplo, las variables Q1, Q2, Q3, y Q4 están en el mismo grupo, luego una variable oculta es asociada a cada grupo de variables con la idea de resumir los valores de esas variables. El proceso es recursivo repitiéndose con las diferentes variables ocultas introducidas por cada grupo hasta tener una sola variable (Oviedo, Moral, y Puris2016).

El resultado final es un árbol (o conjunto de árboles) con una jerarquía de variables auxiliares que está más de acuerdo con la estructura real de las variables, ver figura 3.3 para el árbol final obtenido.

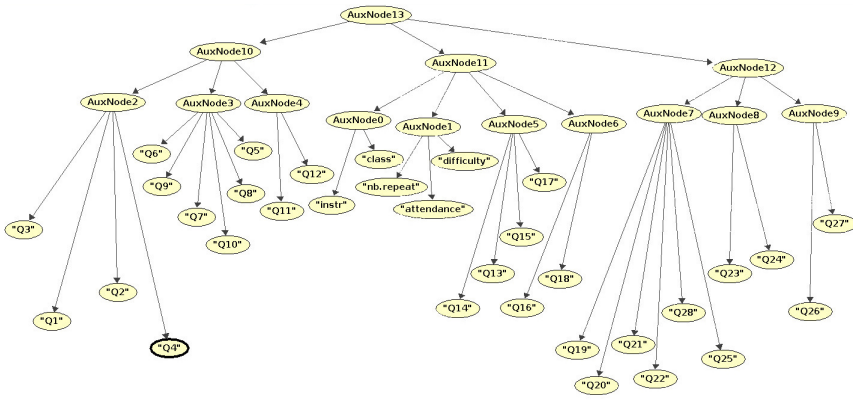

Figura 3.3 Red bayesiana con agrupamiento jerárquico para evaluación de estudiantes

Tabla 3.4 Variables y sus descripciones (Turkiye Studet Evaluation)

Variable	Descripción
instr	Identificación del instructor; se toman valores de 1,2,3
clase	Código del curso (descriptor); se toman valores de 1-13
repeticiones	Número de veces que el estudiante toma este curso; se toman valores de 0,1,2,3,...
asistencia	Código del nivel de asistencia; se toman valores de 0, 1, 2, 3, 4
dificultad	Dificultad del curso según el estudiante; se toman valores de 1,2,3,4,5
Q1	El contenido, método de enseñanza y sistema de evaluación se proporcionó al inicio.
Q2	El curso y sus objetivos estaban claramente establecidos al inicio del periodo.
Q3	El curso merecía la cantidad de créditos asignados.
Q4	El curso fue impartido de acuerdo al syllabus entregado el primer día.
Q5	Las discusiones en clases, tareas asignadas, y aplicaciones fueron satisfactorios.
Q6	Los libros y otros recursos del curso fueron suficientes y actualizados.
Q7	El curso permitió trabajo de campo, aplicaciones en laboratorio, análisis y otros.
Q8	Las pruebas, tareas, proyectos y exámenes contribuyeron a ayudar al aprendizaje.
Q9	Disfruté mucho de la clase y deseaba participar activamente en las conferencias.
Q10	Mis expectativas iniciales sobre el curso se cumplieron al final del período o año.
Q11	El curso fue pertinente y beneficioso para mi desarrollo profesional.
Q12	El curso me ayudó a ver la vida y el mundo con una nueva perspectiva.
Q13	Los conocimientos del instructor eran relevantes y actualizados.
Q14	El instructor vino preparado a las clases.
Q15	El instructor enseñó de acuerdo al plan de estudios entregado.
Q16	El instructor estaba comprometido con el curso y era comprensible.
Q17	El instructor llegó a tiempo para las clases.
Q18	El instructor tiene una voz suave y fácil de entender.
Q19	El instructor hizo uso efectivo de las horas de clases.
Q20	El instructor explicó el curso y estaba dispuesto a ayudar a los estudiantes.
Q21	El instructor demostró un enfoque positivo de los estudiantes.
Q22	El instructor estaba abierto y respetuoso de las opiniones de los estudiantes.
Q23	El instructor incentivó la participación en el curso.
Q24	El instructor entregó tareas/proyectos y mantuvo ayuda guiada a estudiantes.
Q25	El instructor respondió a las preguntas sobre el curso dentro y fuera del aula de clases.
Q26	El sistema de evaluación permite medir de manera efectiva los objetivos del curso.
Q27	El instructor proporcionó la solución de los exámenes y los discutió en clases.
Q28	El instructor trató a todos los estudiantes de manera correcta y objetiva.

Todas las variables Q_i toman cinco valores $1,...,5$. El resultado del logaritmo de la probabilidad en una validación cruzada de 10 tandas se muestra en la Tabla 3.5

En este caso los mejores resultados son obtenidos usando nuestro procedimiento con la

Tabla 3.5 Log-likelihood con validación cruzada=10. Datos de deserción.

PC	K2	NBayes	AClass 2	AClass 6	AClass 12
-263425.48	-130190.69	-171900.24	-228961.76	-140751.66	-128164.73
HNBayesEM A	HNBayesEM B				
-113482.01	-114998.16				

métrica Akaike. K2 provee también buenos resultados. El rendimiento de AutoClass depende del número clases ocultas, siendo muy pobre para 2 casos, pero va mejorando con mayor número de casos. Con 12 valores ocultos es mejor que K2.

Este caso parece ser el más apropiado para nuestro método, ya que hay un alto número de variables que se relacionan por grupos, pero que miden diferentes aspectos del grado general de satisfacción de los estudiantes con un curso. Así, solo una variable oculta no parece ser lo suficientemente adecuada. Esto es más evidente en el árbol aprendido con las observaciones y variables ocultas que se representan en la figura 3.3 donde se utiliza la métrica Akaike. Se pueden observar los siguientes hechos:

- La variable $AuxNode2$ resume las variables $Q1, Q2, Q3, Q4$, que son variable relacionadas con la información proporcionada al estudiante. La variable $AuxNode3$ resume las variables $Q5, Q6, Q7, Q9, Q10$ las mismas que están relacionadas con los materiales utilizados por el instructor. La variable $AuxNode4$ está relacionada con las variables $Q11$ y $Q12$ que hacen referencia al beneficio esperado por el estudiante en relación al curso. Al mismo tiempo que estas tres variables $AuxNode2, AuxNode3, AuxNode4$ constituyen un nuevo grupo con la variable artificial $AuxNode10$.
- $AuxNode0$ es la variable artificial para el grupo de variables relacionadas con el curso y el instructor, mientras que $AuxNode1$ corresponde al grupo de variables relacionadas con la dificultad de el curso. $AuxNode5$ es la variable para el grupo $\{Q13, Q14, Q15, Q17\}$, que son variables relacionadas con la preparación del profesor. $AuxNode6$ hace referencia a las habilidades pedagógicas para la exposición del instructor. Todas estas variables ocultas son agrupadas en la variable $Aux11$.
- $AuxNode7$ cubre las variables $Q19, Q20, Q21, Q22, Q25, Q28$, que hacen referencia a la actitud del instructor con los estudiantes. $AuxNode8$ es la variable asociada a las variables $Q23, Q24$, las mismas que están asociadas a la participación del estudiante, mientras que $AuxNode9$ depende de $Q26, Q27$, que son variables relacionadas con la evaluación del estudiante. Todas estas 3 variables ocultas son agrupadas con la variable artificial $AuxNode12$.

3.6. Conclusiones parciales

En este capítulo hemos propuesto un nuevo procedimiento para inducir modelos gráficos probabilísticos con variables ocultas de los datos: 'un agrupamiento jerárquico'. En lugar de considerar sólo una variable oculta como en AutoClass, se ha considerado un árbol de variables ocultas.

Experimentos: Se ha demostrado que este modelo tiene un buen rendimiento para estimar una distribución de probabilidad conjunta de un conjunto de datos observados y tiene algunas ventajas adicionales en comparación con los procedimientos generales de aprendizaje de redes bayesianas. Este modelo produce árboles (o un bosque de árboles) que son fáciles de interpretar y rápidos para el cálculo.

El rendimiento de este modelo depende del caso particular donde está siendo aplicado. En general, es apropiado en problemas que hay variables relacionadas entre ellas y que esto es debido al hecho de que dependen de factores ocultos no observados. Este es el caso del conjunto de datos de los estudiantes de la Universidad Técnica Estatal de Quevedo (Ecuador) donde se tiene algunas variables que manifiestan dependencia del nivel económico de la familia de los estudiantes, otras variables están relacionadas con las características académicas de los estudiantes, etc.

Los resultados obtenidos al aplicar nuestro algoritmo jerárquico con la métrica Akaike son mejores que los obtenidos por los otros del estado del arte. A mayor número de variables ocultas mejoran los resultados.

En resumen, nuestro procedimiento produce una muy buena estimación de la distribución de probabilidades conjunta y al mismo tiempo genera un entorno natural y fácil de interpretar el agrupamiento de la estructura gráfica de las variables observables en una jerarquía.

En el futuro, se ha planificado extender el estudio de deserción incluyendo más variables académicas que parecen tener más influencia en las variables de interés. También se considera el problema de deserción como un problema de clasificación supervisada, adaptando nuestro procedimiento jerárquico para clasificación y comparando el desarrollo de este con otros modelos.

También se pretende analizar otros problemas prácticos en los que nuestro modelo puede tener un papel importante, como puede ser el resumen de respuestas que se obtienen de un cuestionario.

4 Optimización de malla variable aplicada al aprendizaje de clasificadores bayesianos

En este capítulo se define una nueva forma de aprendizaje estructural para clasificadores bayesianos. Hay clasificadores como el Naive Bayes que tienen una estructura fija y, por lo tanto, sólo necesitan estimar los parámetros. Pero otros necesitan determinar la 'mejor' estructura. Nosotros vamos a construir clasificadores que son redes bayesianas arbitrarias por lo que estamos en el segundo caso.

Existen dos enfoques fundamentales para llevar a cabo esta tarea:

- Métodos de filtrado.- Estos métodos se basan en el uso de medidas de asociación o métricas que se calculan en los datos. El clasificador se construye en base a estas medidas, por ejemplo, optimizando una métrica como se hace en la construcción del clasificador TAN (Friedman, Geiger, y Goldszmidt1997). Hay dos enfoques principales dentro del aprendizaje de clasificadores bayesianos a partir de un conjunto de observaciones: pruebas de independencia (Spirtes, Glymour, y Scheines2000) y métodos de score + búsqueda (Cooper y Herskovits1992). La primera, está basada en hacer test estadísticos relacionados con los datos que están representados en el grafo. En el segundo existe una métrica que permite medir la idoneidad de un grafo dado un conjunto de datos propuestos.
- Métodos de envolvente.- El mejor clasificador se calcula optimizando su comportamiento en los datos de entrenamiento usando un método de validación cruzada (Mosier1951) por ejemplo usando la validación basada en 10 tandas.

En nuestro caso vamos a usar un método envolvente para el aprendizaje de los clasificadores, donde se tiene como problema el encontrar la topología que mejor clasifique los datos. Este problema de optimización es complejo, ya que el espacio de búsqueda es el espacio de todas las redes bayesianas que es del orden de $n!2^{\binom{n}{2}}$, siendo n el número total de variables. De hecho el problema de aprender una red bayesiana que optimice una métrica como BDEu es un problema NP-difícil (Chickering1996). Por ese motivo, es conveniente usar métodos de optimización combinatoria que permitan obtener buenas soluciones en tiempos razonables.

Uno de los métodos de optimización más utilizadas son las metaheurísticas poblacionales, las mismas que son algoritmos iterativos de búsqueda en el espacio de soluciones *S* que emplean un conjunto de soluciones (población) en cada iteración del algoritmo. Proporcionan de forma exclusiva un mecanismo que explora de manera paralela el espacio de soluciones, y su eficiencia depende en gran medida de cómo se manipule la población (Duarte2007).

En los últimos años ha habido un crecimiento espectacular en el desarrollo de procedimientos heurísticos para resolver problemas de optimización. Este hecho queda claramente reflejado en la creación de revistas especializadas para la difusión de este tipo de procedimientos; como es el caso de la revista 'Journal of Heuristics' editada por primera vez en el año 1995.

En este capítulo, se propone aprender clasificadores bayesianos que a través de una nueva metaheurística conocida como optimización basada en mallas variables (VMO) de solución al problema de optimización. VMO es una metaheurística poblacional con características de los algoritmos evolutivos, donde un conjunto de nodos representa las soluciones potenciales a un problema de optimización y forman una malla (población) que de manera dinámica crece y se desplaza por el espacio de búsqueda (evoluciona) (Puris et al.2012). La optimización en mallas variables ha sido aplicada a diferentes problemas de optimización tales como el problema del viajante de comercio, el mismo que ya ha sido tratado por (Oviedo et al.2014), (Molina et al.2013), entre otros.

El algoritmo VMO fue incorporado dentro del software Elvira. Se utilizaron algunas de las clases principales y se crearon otras nuevas que funcionan acorde a las distintas necesidades funcionales del algoritmo.

El estudio empieza analizando brevemente los problemas de optimización y se determina la importancia de considerar a los operadores que permitirán mover a la población a las zonas favorables del espacio de solución.

Luego se describe de manera general la metaheurística optimización de malla variable, explicando como genera la malla inicial, los nodos en dirección a los extremos locales y al extremo global, así cómo la generación de los nodos a partir de las fronteras de la malla. Luego se analizan características del método y sus operaciones. Finalmente se aplica el método para estimar clasificadores bayesianos y se realizan experimentos comparando con otros clasificadores. Vamos a usar un método de envolvente y la función que se optimiza es el comportamiento en el conjunto de entrenamiento mediante validación cruzada. Algunos algoritmos de aprendizaje estructural de redes bayesianas con los que se van a comparar los resultados son ByNet (Chávez2008), BayesChaid (Chávez2008), BayesPSO (Chávez2008).

4.1. Problemas de aprendizaje estructural. Definición y notación

El aprendizaje estructural de redes bayesianas ha sido un tema de investigación activo en los últimos 30 años, debido a que representa un problema de optimización NP-completo, en el caso general: dado un conjunto de datos $\mathscr{D}$, que contiene múltiples muestras de un conjunto de variables aleatorias, el objetivo es encontrar la estructura más probable, en un espacio exponencial de todas las estructuras posibles.

Un problema de optimización se define matemáticamente como: dados un espacio de soluciones S y una función objetivo $FO : S \to \mathbb{R}$, encontrar el valor $n^* \in S$ que maximiza (o minimiza) la función objetivo:

$$n^* = \arg\max_{n \in S} FO(n)$$

A los elementos de S les llamaremos soluciones factibles y también nodos (no confundir con los nodos de las redes bayesianas). Habitualmente, cada elemento $n \in S$ viene descrito por un vector m-dimensional $n = (v_1, \ldots, v_m)$, aunque también se pueden considerar otras representaciones.

Las metaheurísticas son algoritmos genéricos de búsqueda de la mejor solución, que son apropiados en el caso de que el problema de optimización sea NP-difícil. Para realizar dicha búsqueda es habitual que exista una función que asigna a cada nodo $n \in S$ un subconjunto $V(n) \subseteq S$ de nodos vecinos. Esta función puede venir dada a partir de una función de similaridad $similitud : S \times S \to \mathbb{R}$ que asigna a cada pareja de nodos un valor real en función de su parecido. En ese caso $V(n)$ se puede definir como:

$$\{n' \in S \mid n' \neq n, similitud(n', n) = \text{máx}\{n'' \in S \setminus \{n\} | similitud(n, n'')\}\},$$

o seleccionando un entero positivo K y suponiendo que $V(n)$ es el conjunto de los K nodos $n' \in S$ que tienen un valor mayor de $similitud(n, n')$. Algunas veces vamos a hablar de distancia que es una función decreciente de la similitud.

Un extremo local es un nodo n tal que $FO(n) \geq FO(n'), \forall n' \in V(n)$ (en problemas de maximización).

A un conjunto de nodos $S' \subseteq S$ se le llama población o malla.

4.2. Descripción general de la metaheurística de optimización de malla variable

La optimización de malla variable (VMO), (Puris et al.2012), es una metaheurística poblacional donde las soluciones son distribuidas como una malla en el espacio m-dimencional. La población

está compuesta por p nodos o soluciones candidatas $(n_1, n_2, ..., n_p)$, codificadas como un vector de puntos flotantes de m componentes, $n_i = (v_1^i, v_2^i, ..., v_m^i) = v_j^i, j = 1..m$.

Esta metaheurística aparece como un esfuerzo de utilizar de forma directa las soluciones encontradas más promisorias (extremos locales, extremo global y soluciones fronteras) para dirigir el proceso de búsqueda del modelo. Bajo este principio, la población puede converger de manera prematura a zonas ya identificadas como prometedoras y dejar de explorar otras zonas desconocidas. Para evitar esta situación desfavorable para la búsqueda y en aras de propiciar la diversidad en la población, VMO implementa un mecanismo de limpieza adaptativa que filtra la población, manteniendo solamente los nodos más representativos de cada zona explorada. El componente adaptativo se adquiere a partir de la forma en que se calcula la distancia de separabilidad permitida en cada iteración del algoritmo (Oviedo et al.2016).

De forma general el proceso de búsqueda desarrollado por VMO se realiza a partir de las siguientes operaciones:

- Expansión: Proceso que se lleva a cabo en cada iteración para explorar las zonas representadas por las mejores soluciones obtenidas (extremos locales, el extremo global y las soluciones fronteras).
- Contracción: Mecanismo aplicado para reducir la población luego del proceso de expansión, manteniendo en la población los nodos más representativos.

Ambas operaciones son las encargadas de incorporar exploración y explotación del espacio de búsqueda. A continuación se detallan los pasos que se realizan en cada proceso.

4.2.1. Proceso de expansión

El algoritmo realiza el proceso de expansión moviendo la población a zonas identificadas como prometedoras. Para ello, crea nuevas soluciones utilizando como base los mejores nodos (calidad) encontrados. A continuación se describe cada paso involucrado en dicho proceso:

- *Generación de la malla inicial*: la malla inicial consta de p nodos, los cuales son generados de forma aleatoria o por otro método que garantice obtener soluciones diversas (Puris et al.2012).
- *Generación de nodos en dirección a los extremos locales*: En este proceso se identifican los k vecinos más cercanos de cada nodo n_i, utilizando una función de semejanza o distancia. Luego se identifica el extremo local $nl^{(i)}$ de cada vecindad para crear nuevas soluciones utilizando cada nodo y su extremo local, solo si $n_i \neq nl^i$ (Molina et al.2013).
  El cálculo del nuevo nodo se lleva a cabo utilizando la ecuación 4.1:

$$n^* = f(n_i, nl^{(i)}, Pr(n_i, nl^{(i)})) \tag{4.1}$$

Donde f es una función que involucra al nodo actual, a su extremo local y un factor Pr que determina la relación entre ambos nodos en cuanto calidad, ver ecuación 4.2. La función f fue definida en (Puris et al.2012) para problemas continuos, pero puede ser redefinida para dominios discretos como se verá más adelante.

$$P_r(n_i, nl^{(i)}) = \frac{1}{1 + |FO(n_i) - FO(nl^{(i)})|} \tag{4.2}$$

- *Generación de nodos en dirección hacia el extremo global*: En este paso cada nodo de la malla es utilizado conjuntamente con el nodo de mejor calidad de todos ng, para generar nuevas soluciones.

$$n^* = g(n_i, ng, Pr(n_i, ng)) \tag{4.3}$$

La funcion g (ecuación 4.3) al igual que la función f, fue definida para dominios continuos en (Puris et al.2012) donde los autores sugieren que al aplicarlo en otros dominios, debe de garantizar que mientras mayor sea la diferencia entre la calidad de cada nodo involucrado (determinado por Pr) mayor será la semejanza del nuevo nodo a ng. Más adelante se realizará la definición para el problema de entrenamiento estructural de redes bayesianas.

- *Generación de nodos a partir de los nodos fronteras de la malla*: Este proceso de generación de nuevos nodos tiene lugar con el objetivo de explorar el espacio de búsqueda en dirección a las fronteras de la malla. Para ello, se selecciona el nodo más y menos distante (similar) de todos los demás, para luego utilizando la función h crear dos nuevas soluciones (una por cada frontera, nf_1, nf_2) según la ecuación 4.4.

$$n_1^* = h(nf_1, w)\ ,\ n_2^* = h(nf_2, w) \tag{4.4}$$

Donde w se conoce como desplazamiento y fue introducido en (Puris et al.2012) en dominios continuos con el objetivo de explorar espacios que presentan los óptimos en las fronteras, (Navarro et al.2009). La función h también será redefinida más adelante para el problema de estudio.

4.2.2. Contracción de la malla

La contracción es el proceso que lleva a cabo VMO para mantener la diversidad en la población. Para ello, se seleccionan los nodos más representativos o diversos de la iteración actual (existentes y creados) a través de un mecanismo conocido como limpieza adaptativa (Puris et al.2012). Este mecanismo elimina de la población los nodos que se encuentren cerca de otro con mejor calidad. A continuación se supone definida una función de distancia entre dos nodos que en general, son una función decreciente de la similaridad.

- Se ordenan todos los nodos seleccionados como malla inicial en función de su calidad.
- De forma secuencial, se compara cada nodo de la malla con sus sucesores, eliminando aquellos cuya similaridad sea mayor que una cota calculada dinámicamente. (Puris et al.2012). Este valor de la similitud debe permitir que el proceso sea decreciente; de manera que se obtenga mayor separabilidad entre los nodos al inicio que al final de la ejecución del método.
- Luego, se completa la malla inicial (de ser necesario) con nodos generados de forma aleatoria.

El valor adaptativo de la cota de distancia (ξ) le permite al método comenzar con exploraciones más generales y luego ir disminuyendo su influencia hasta centrarse en una zona más pequeña del espacio de búsqueda. Este elemento aumenta el nivel de explotación del método y lo hace más robusto. La ecuación propuesta en (Puris et al.2012) para calcular ξ se describe en la ecuación 4.5.

$$\xi = \begin{cases} \frac{longitud(min,max)}{4}, & si \quad c < 15\,\%C \\ \frac{longitud(min,max)}{8}, & si \quad 15\,\%C \leq c \leq 30\,\%C \\ \frac{longitud(min,max)}{16}, & si \quad 30\,\%C \leq c \leq 60\,\%C \\ \frac{longitud(min,max)}{50}, & si \quad 60\,\%C \leq c \leq 80\,\%C \\ \frac{longitud(min,max)}{100}, & si \quad c \geq 80\,\%C \end{cases} \tag{4.5}$$

Donde C representa el total de evaluaciones de la función objetivo y c el valor actual de este parámetro.

4.2.3. Parámetros y funcionamiento general del método

Luego de haber definido los pasos de cada uno de los procesos utilizados por VMO, se presenta en el algoritmo 4.1 la estructura general del mismo y en la tabla 4.1 se describen los parámetros utilizados por el algoritmo.

Tabla 4.1 Parámetros del algoritmo VMO

Parámetro	Descripción
p	Cantidad de nodos que integran la malla inicial en cada iteración
k	Cantidad de nodos que determinan la vecindad de cada nodo
C	Condición de parada del algoritmos

```
1  Generación de la malla inicial (M) de forma aleatoria con p nodos
2  Evaluar los nodos de la malla inicial, seleccionar el mejor ng
3  Crear un malla temporal M^temp <-- M
4  Repetir
5        Para cada nodo n_i ∈ M hacer
6              Encontrar sus k nodos más cercanos
7              Determinar el mejor de los vecinos nl^(i)
8              Si nl^(i) es mejor que n_i, entonces
9                    Aplicar ecuación 4.1 con nl^(i) y n_i
10             M^temp =<-- n*
11             Fin Si
12       Fin Para
13       Para cada nodo n_i ∈ M hacer
14             Generar un nuevo nodo usando la ecuación 4.3 con ng y n_i
15             M^temp =<-- n*
16       Fin Para
17             Seleccionar los dos nodos fronteras de la malla nf_1 y nf_2
18             Generar un nuevo para cada frontera usando la ecuación 4.4
19             M^temp =<-- n*_1, M^temp =<-- n*_2
20       Ordenar los nodos de M^temp según su calidad
21       Aplicar operador de limpieza adaptativo a M^temp
22       Construir M con los p primeros nodos de M^temp
23       En caso que no se alcance los p nodos en M, completar con soluciones aleatorias.
24 Hasta C evaluaciones
```

Algoritmo 4.1: Funcionamiento general VMO

4.3. VMO aplicado al aprendizaje de clasificadores bayesianos

El aprendizaje estructural de una red bayesiana representa uno de los elementos fundamentales en el desempeño de un clasificador bayesiano. Este problema cae en la categoría NP-difícil (Cooper y Herskovits1992), por lo que ha sido objeto de estudio de un sin número de investigaciones en el área. Dentro de los algoritmos aproximados, las metaheurísticas poblacionales se han utilizado en numerosas ocasiones mejorando en gran medida el rendimiento de los clasificadores.

En esta sección aplicaremos el algoritmos VMO al problema de aprendizaje estructural de un clasificador bayesiano. Para ello, se redefinen algunos operadores del algoritmo ajustándolos a las características del problema. Luego se presenta un estudio de los parámetros del modelo y por último se realiza un análisis comparativo con otros clasificadores bayesianos del estado del

arte, utilizando un conjunto de bases de conocimiento del repositorio de la UCI.

4.3.1. Modificaciones del algoritmo VMO

El primer paso en el proceso de adaptación del algoritmo VMO al aprendizaje estructural de un clasificador bayesiano lo determina la **representación de los nodos** o soluciones. En este sentido tomaremos el nodo como un conjunto de arcos $(X_i, X_j), i \neq j$ dirigidos, que representan las relaciones de dependencia entre las variables X_i y X_j de la red. La figura 4.1 presenta un ejemplo de un nodo para una red de 4 variables.

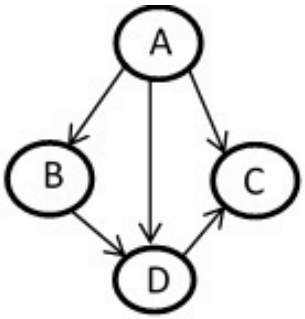

Figura 4.1 Representación de una BN como conjunto de arcos

Otro elemento que definiremos es la **similitud entre dos redes**, ya que se utiliza por VMO para determinar los nodos fronteras y para realizar el proceso de contracción de la malla. Nuestra propuesta radica en utilizar la operación de intersección entre dos conjuntos para obtener los arcos que comparten dos redes o nodos $(BN_1 \cap BN_2)$. Luego la cardinalidad del conjunto solución, lo tomaremos como la similitud entre ambas redes (ver la ecuación 4.6).

$$Similitud_{BN_1,BN_2} = Card(BN_1 \cap BN_2) \tag{4.6}$$

Otro de los elementos que se definen en la propuesta es la forma de **generar nuevas redes** para el proceso de expansión de la malla determinada por las funciones f, g y h (ver ecuaciones 4.1, 4.3 y 4.4). Es importante señalar que la redes que se generen tienen que cumplir con la restricción que establecen las redes bayesianas de ser grafos acíclicos.

A continuación se definirán las formas de generar los nuevos nodos dependiendo de la operación:

- *Generación de nodos al azar*: este tipo de nodo se construye con arcos generados al azar y añadidos, si no forman un ciclo y no están dirigidos a la variable clase. El proceso se repite hasta que todas las variables estén conectadas.

- *Generación a partir de un nodo*: esta generación se lleva a cabo en la función h del modelo, con el objetivo de introducir un pequeño cambio a la red. Para ello, se selecciona aleatoriamente un arco del nodo y se procede a cambiar su orientación. El nuevo arco es incluido en la red sólo si no introduce ciclo.

$$BN^* = (BN_1 \setminus a_{ij}) \cup a_{ji}, con\ a_{ij}\ seleccionado\ aleatoriamente \tag{4.7}$$

- *Generación a partir de dos nodos*: para este caso, se utilizan dos operaciones entre conjuntos; la unión y la unión de las diferencias. La operación unión obtiene un nuevo nodo con todos los arcos de las dos redes operadas, esto se puede ver en la ecuación 4.8. Por otro lado, la unión de las diferencias genera una nueva red con los arcos presentes en una red, pero no en la otra, tal como se visualiza en la ecuación 4.9. Las dos operaciones son presentadas en las figuras 4.2 y 4.3 y utilizadas como las funciones f y g en el proceso de expansión. Para este caso específico no utilizaremos el factor Pr presente en la propuesta original (Puris et al.2012) de VMO.

$$BN^* = (BN_1 \cup BN_2) \tag{4.8}$$

$$BN^* = (BN_1 \setminus BN_2) \cup (BN_2 \setminus BN_1) \tag{4.9}$$

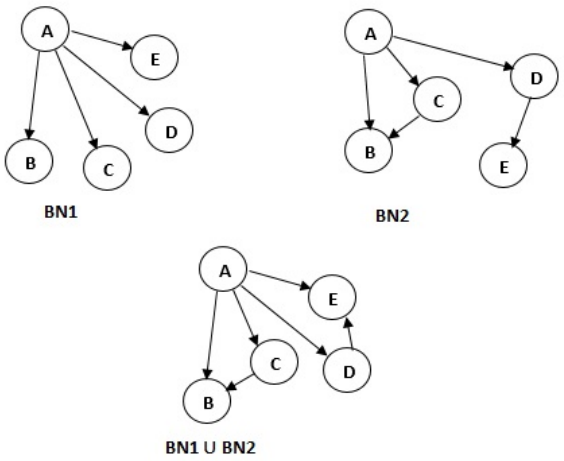

Figura 4.2 Operación unión entre BN_1 y BN_2

En el proceso de creación de las nuevas redes, se puede dar el caso que se presenten redes con ciclos, cuando esto sucede se aplica un algoritmo de eliminación de ciclos que selecciona un

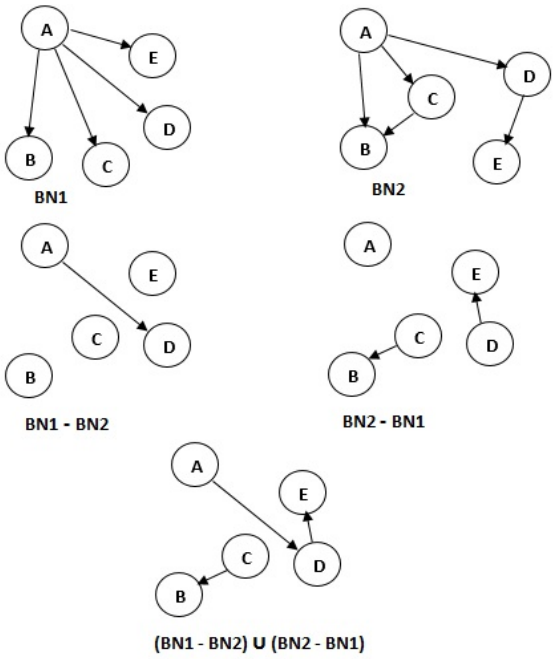

Figura 4.3 Operación unión de las diferencias entre BN_1 y BN_2

arco de forma aleatoria entre, el conjunto de arcos que forman el ciclo, y se invierte el sentido del arco. Este proceso se repite hasta que se eliminen todos los ciclos de la red.

La **cota de distancia** o valor de separabilidad entre soluciones, es otro elemento que se redefinió en la propuesta (ver ecuación 4.10), donde A representa el número de variables o características del problema, y al igual que la propuesta original, c y C representan las evaluaciones de la función objetivo (actual y total respectivamente) e incorporan la variabilidad de ξ. En este caso, es la siguiente:

$$\xi = \begin{cases} 0,8*A & si & c < \frac{1}{4}*C \\ 0,8*A & si & c < \frac{1}{4}*C \\ 0,7*A & si & c < \frac{2}{4}*C \\ 0,6*A & si & c < \frac{3}{4}*C \\ 0,5*A & Otros\,casos & \end{cases} \tag{4.10}$$

Donde A es el número de atributos del problema.

Por último, para obtener el valor de la función objetivo o **calidad de una red**, se utiliza la precisión de la clasificación con validación cruzada de 10 en el conjunto de entrenamiento. El algoritmo general de aplicación se presenta en 4.2.

1 Generación de la malla inicial M de forma aleatoria con p BN
2 Construir una clasificador con cada red y seleccionar la de mejor precisión BNg de M
3 Crear un malla temporal $M^{temp} \longleftarrow M$
4 **Repetir**
5 **Para cada** nodo $BN_i \in M$ **hacer**
6 Aplicar ecuación 4.6 para encontrar sus k vecinos más similares.
7 Determinar el clasificador de mejor precisión entre los vecinos $BNl^{(i)}$
8 **Si** $BNl^{(i)}$ tiene mejor precisión que BN_i, **entonces**
9 Aplicar ecuación 4.8 con $BNl^{(i)}$ y BN_i
10 $M^{temp} \longleftarrow BN^*$
11 **Fin Si**
12 **Fin Para**
13 **Para cada** nodo $BN_i \in M$ **hacer**
14 Generar un nuevo nodo usando la ecuación 4.9 con BNg y BN_i
15 $M^{temp} = \longleftarrow BN^*$
16 **Fin Para**
17 Seleccionar los dos nodos fronteras de la malla BN_{f1} y BN_{f2}
18 Generar un nuevo para cada frontera usando la ecuación 4.7
19 $M^{temp} \longleftarrow BN_1^*$, $M^{temp} \longleftarrow BN_2^*$
20 Ordenar las BN de M^{temp} según su precisión
21 Aplicar operador de limpieza adaptativo a M^{temp}
22 Construir M con las p primeras BN de M^{temp}
23 En caso que no se cuente con las p BN en M, completar con soluciones aleatorias.
24 **Hasta** C evaluaciones

Algoritmo 4.2: Funcionamiento general VMO

4.3.2. Complejidad computacional

La complejidad del algoritmo BayesVMO dependerá de dos procesos, los métodos de métrica (score) más búsqueda. La búsqueda es aplicada para la metaheurística VMO donde la operación más compleja es obtener la matriz de similitud de la población en cada una de las iteraciones del algoritmo. Esta complejidad está establecida como $O(CP^2A^4)$, donde C representa el número de iteraciones del algoritmo, P el tamaño de la población y A el número de atributos del problema. P^2 representa el número de comparaciones en la población (todos contra todos) y A^4 la operación intersección entre las estructuras (A^2 es el número máximo de arcos de un grafo y el cálculo de similitud, cada arco de un grafo debe ser buscado en otro grafo).

Con la métrica (score), la precisión de la clasificación por el método de validación cruzada es utilizada y ejecutada una vez por solución en cada iteración del algoritmo. Sin embargo, la complejidad del score depende de una propagación probabilística (Pr) y un tamaño de los datos (N). Esta complejidad está establecida como $O(CPPrN)$ con los que se incrementa la complejidad del algoritmo BayesVMO. La complejidad de este algoritmo es similar a la de los algoritmos ByNet, BayesChaid y BayesPSO presentados en la sección de experimentos 4.3.3 (conforme lo establecido por el autor) y es más complejo que otros clasificadores bayesianos simples como BN K2, BN TAN, NB, SNB presentados en la misma sección.

Tabla 4.2 Conjunto de datos y su descripción

Conjunto de datos	Atributos discretos	Atributos continuos	Clases	Instancias
Mammographic	4	1	2	961
Lung-cancer	56	0	3	32
Hepatitis	13	6	2	155
E colic	0	7	8	336
Breast-cancer-w	10	0	2	683
Contact-lenses	4	0	3	24
Hayes-roth-m	4	0	3	132
Monk	1	6	0	2
Vote	16	0	2	300
Balance-scale	4	0	3	625
Tic-tac-toe	9	0	2	958
Iris	0	4	3	150
Labor	8	8	2	57
Soybean	35	0	19	683

4.3.3. Puntos de referencia y estudio experimental

En esta sección se presenta el estudio experimental de la propuesta VMO al aprendizaje estructural (BayesVMO). Para ello, utilizamos 14 bases de conocimientos incluidas en el repositorio UCI ML (Lichman2013), cuya descripción se presenta en la tabla 4.2. La experimentación comienza realizando un estudio de los parámetros k (cantidad de nodos que determinan la vecindad de cada nodo) y p (cantidad de nodos que integran la malla inicial en cada iteración) del algoritmo. Luego la mejor configuración es utilizada para un estudio comparativo con otros clasificadores bayesianos del estados del arte.

Para el análisis de los resultados se aplica un conjunto de técnicas estadísticas no paramétricas presentes en (García et al.2009) para la comparación de resultados en problemas de clasificación. Entre ellos, utilizaremos el test de Iman-Davemport (Iman y James1980) para determinar si existen diferencias significativas en un grupo de resultados y en caso positivo, utilizaremos el test de comparaciones múltiples de Holm (Holm1979) utilizando como algoritmo de control el de mejor comportamiento.

En todos los experimentos se realiza un máximo de 10000 evaluaciones de la función objetivo (valor del parámetro C) y cada variante del algoritmo se ejecutó 50 veces de manera independiente para cada conjunto de datos, utilizándose el valor promedio para el análisis comparativo.

4.3.3.1. Estudio de parámetros

En esta sección, se realiza el ajuste de los parámetros k (cantidad de nodos que determinan la vecindad de cada nodo) y p (cantidad de nodos que integran la malla inicial en cada iteración) utilizando los valores $p = (12, 24, 48)$ y $k = (3, 5, 7)$ respectivamente. La tabla 4.3 presenta los resultados del estudio realizado para todas las combinaciones de parámetros representados como (p, k).

Tabla 4.3 Resultados del ajuste de parámetro de BayesVMO (EU,k,p)

Datos	**(12,3)**	**(12,5)**	**(12,7)**	**(24,3)**	**(24,5)**	**(24,7)**	**(48,3)**	**(48,5)**	**(48,7)**
Mammographic	84,39	84,29	84,50	84,39	84,50	84,18	83,87	83,77	83,77
Lung-cancer	78,13	78,13	75,00	75,00	78,13	75,00	68,75	68,75	75,00
Hepatitis	93,55	92,90	92,90	92,90	93,55	92,26	90,97	90,97	90,97
E colic	76,49	77,08	76,19	81,85	76,79	76,79	76,68	76,19	76,79
Breast-cancer-w	97,85	98,14	98,00	97,85	98,00	97,57	97,57	97,57	97,71
Contact-lenses	95,83	95,83	95,83	91,67	95,83	87,50	87,50	87,50	95,83
Hayes-roth-m	84,09	84,09	83,33	83,33	84,09	75,00	79,55	80,30	83,33
Monk	97,58	89,52	89,52	90,80	89,52	91,94	87,90	94,35	95,97
Vote	93,79	93,33	92,87	93,10	93,33	92,41	92,87	92,00	92,87
Balance-scale	86,72	86,08	86,88	86,08	86,40	84,96	84,48	85,76	84,64
Tic-tac-toe	75,68	75,68	75,99	74,84	75,47	73,49	72,76	74,01	74,01
Iris	98,00	98,67	97,33	98,67	98,00	96,67	96,67	96,67	96,67
Labor	100,00	100,00	100,00	100,00	100,00	95,00	97,50	95,00	97,50
Soybean	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00

El test de Iman-Davemport con 8 y 104 grados de libertad aplicado a la tabla 4.3 obtiene un valor $p = 2,92E^{-10}$ menor que el valor de $\alpha = 0,05$ utilizado para el test. Con este resultado se rechaza la hipótesis de semejanza, determinando que existen diferencias significativas en los resultados computados. A continuación aplicaremos el test de Holm, tomando como algoritmo de control la variante mejor clasificada BayesVMO(12,3)

La tabla 4.4 nos permite visualizar los resultados de la prueba, donde la hipótesis de semejanza ha sido rechazada para las 4 primeras variantes (Valor$p < \alpha/i$), determinándose que el algoritmo de control obtiene resultados significativamente mejores que todas ellas. Para el caso de las otras alternativas (BayesVMO(24,3), BayesVMO(12,7), BayesVMO(24,5) y BayesVMO(12,5)), los resultados dicen que no hay diferencias significativas.

De este análisis se puede determinar que el algoritmo BayesVMO obtiene los resultados más prometedores con los parámetros $p = 12$ y $k = 3$. A continuación utilizaremos esta variante para el análisis comparativo con otros clasificadores bayesianos.

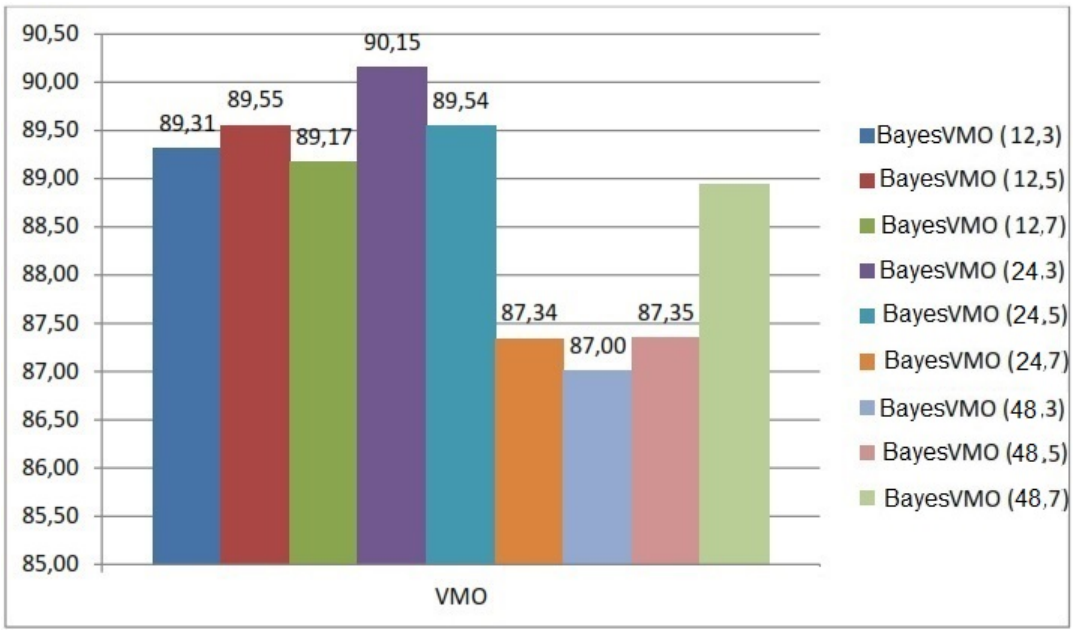

Figura 4.4 Resultados computados para el análisis comparativo con otros clasificadores bayesianos

Tabla 4.4 Resultados de la prueba de HOLM para BayesVMO(12,3) como algoritmo de control

Algoritmo	z	Valor P	α/i	Hipótesis
BayesVMO(48,5)	4,209	$2{,}560E^{-5}$	0,0062	Rechazada
BayesVMO(48,3)	4,174	$2{,}981E^{-5}$	0,0071	Rechazada
BayesVMO(24,7)	3,726	$1{,}942E^{-4}$	0,0083	Rechazada
BayesVMO(48,7)	2,863	0,0041	0,01	Rechazada
BayesVMO(24,3)	1,276	0,201	0,0125	Aceptada
BayesVMO(12,7)	1,173	0,240	0,0166	Aceptada
BayesVMO(24,5)	0,310	0,756	0,025	Aceptada
BayesVMO(12,5)	0,276	0,782	0,05	Aceptada

4.3.3.2. Análisis comparativo con otros clasificadores bayesianos

En esta sección aplicaremos algunos clasificadores bayesianos a los mismos problemas de estudio. Dentro de esto, tenemos algunos clásicos como: BN K2, BN TAN, NB, CBN (clasificador Naive Bayes) y otros como ByNet, BayesChaid, BayesPSO presentados en (Chávez2008). A continuación se realiza una breve descripción de estos últimos clasificadores:

- **ByNet**, Este algoritmo basa su proceso en la obtención de árboles de decisiones basados en la técnica CHAID. En esta técnica el usuario decide la cantidad de árboles que desea obtener y procede a la segmentación de la población a partir de la variable clase. Una vez construido el primer árbol el conjunto de variables que forman parte de él son descartadas, lo que ayuda a la reducción de variables (Chávez2008).
  En este caso, la técnica CHAID se aplica para conocer las variables que pueden ser elimi-

nadas. También para comprender el orden de importancia de los rasgos desde el punto de vista estadístico y de esta manera obtener un árbol en forma automática (Chávez2008).
En el algoritmo ByNet el primer árbol obtenido se dividirá por la variable más significativa acorde al test Chi-cuadrado. Este test nos ayuda a solucionar el problema de sobreajuste que se da ya que el algoritmo realiza una búsqueda incompleta. Las siguientes variables que formen parte de ese árbol estarán en niveles inferiores, lo que significa que su posición dentro de la red estará más alejada de la clase (Chávez2008).

- **BayesChaid**, está técnica se basa en ideas de la técnica de CHAID con adaptaciones. Éstas esencialmente están relacionadas en hacer la búsqueda de las dependencias entre las variables.
  El algoritmo BayesChaid se basa también en el criterio de la prueba Chi-cuadrado para obtener la estructura de la red. Al inicio trabaja de manera similar al algoritmo Naive Bayes, pero incluye un proceso de selección de rasgos (según el criterio del Chi-cuadrado). De esta manera garantiza que las variables más relacionadas con la clase, se encuentren en relación directa con ella. Los árboles son creados usando el algoritmo de búsqueda primero en profundidad (Depth First) y primero en amplitud (Breadth First) (Chávez2008).
- **BayesPSO**, este modelo utiliza la metaheurística de optimización de enjambre de partículas (Particle Swarm Optimization - PSO) como el enfoque de búsqueda y el score es la presición de la clasificación. El algoritmo BayesPSO difiere de manera notable con respecto a los otros dos anteriores. Es mucho más complejo en espacio, pues parte de estructuras de redes seleccionadas al azar y después de un proceso iterativo, se llega a la estructura final de la red (Chávez2008).

La tabla 4.5 muestra los resultados de computar cada uno de los clasificadores seleccionados. Es bueno aclarar, que los resultados de ByNet, BayesChaid y BayesPSO fueron tomados exactamente y como fueron presentados por el autor, incluso se utilizó la misma cantidad de evaluaciones de la función objetivo de BayesPSO, C=10000.
En el análisis de los resultados, al aplicar el test de Iman-Davemport se observan diferencias significativas en el grupo de algoritmos debido a que el valor $p = 4,316E^{-4}$ es estrictamente menor que $0,05$. Por sus parte el test de Holm es aplicado utilizando como algoritmo de control la propuesta BayesVMO(12,3) debido a que es la que mejor clasificación obtuvo. En los resultados mostrados en la tabla 4.6 se puede observar que la hipótesis de igualdad es rechazada en cada prueba independiente, concluyéndose que nuestra propuesta obtiene resultados significativamente superiores que todos los otros clasificadores.

Tabla 4.5 Resultados obtenidos para VMO - Otros

Datos	ByNet	BChaid	BPSO	RBK2	RBTAN	CBN	NB	TAN	BVMO
Mammographic	81,89	83,14	82,62	82,41	81,27	83,25	81,99	79,29	84,39
Lung-cancer	78,13	75,00	78,13	71,88	65,63	75,00	56,25	46,87	78,13
Hepatitis	85,16	86,45	83,87	84,52	85,16	85,16	87,09	87,74	93,55
E colic	67,26	85,12	84,52	85,12	84,82	85,42	77,38	62,79	76,49
Breast-cancer-w	90,78	97,51	97,36	97,36	95,31	97,36	97,13	87,83	97,85
Contact-lenses	87,50	83,33	83,33	70,83	66,67	70,83	83,33	66,66	95,83
Hayes-roth-m	57,58	81,06	74,24	72,73	67,42	80,30	66,66	57,57	84,09
Monk	72,58	70,97	99,19	79,03	95,97	77,42	73,38	57,25	97,58
Vote	94,33	91,67	92,67	91,33	93,67	89,67	89,65	94,03	93,79
Balance-scale	63,52	92,16	93,92	92,16	92,96	92,16	86,40	82,56	86,72
Tic-tac-toe	70,25	72,96	72,65	76,62	76,83	69,62	70,98	73,06	75,68
Iris	95,33	94,00	95,33	94,00	94,67	94,00	91,33	92,00	98,00
Labor	84,21	87,72	89,47	91,22	89,47	89,47	85,00	90,00	95,00
Soybean	68,08	93,11	89,16	94,58	94,58	92,97	100,00	99,87	100,00

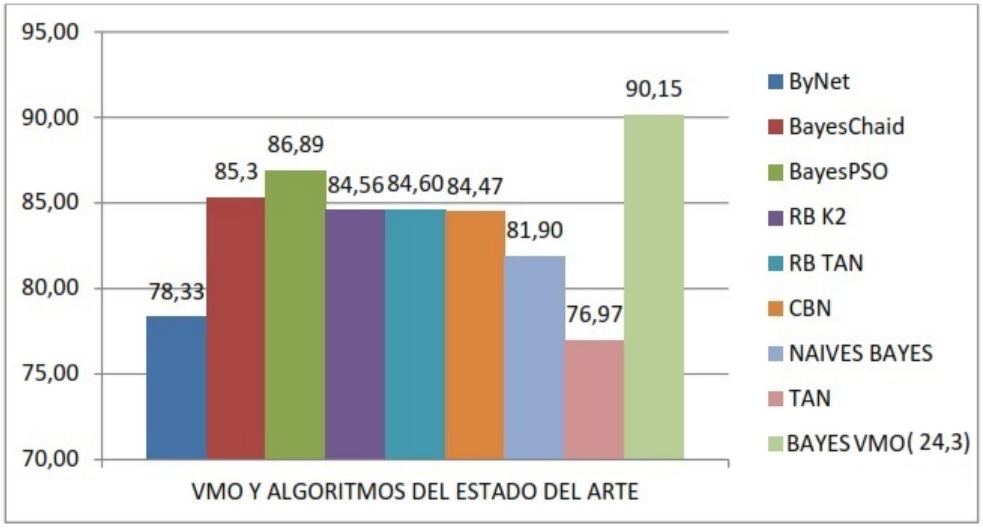

Figura 4.5 Resultados computados para el análisis comparativo con otros clasificadores bayesianos

Tabla 4.6 Resultados de la prueba de Holm's con BayesVMO como algoritmo de control

Algoritmo	Z	p-valores	α/i	Hipótesis
TAN	4.243	$2.19E^{-5}$	0.0062	**Rechazado**
NB	3.864	$1.11E^{-4}$	0.0071	**Rechazado**
ByNet	3.795	$1.47E^{-4}$	0.0083	**Rechazado**
CBN	2.725	0.0064	0.01	**Rechazado**
RB TAN	2.622	0.0087	0.0125	**Rechazado**
RB K2	2.415	0.0157	0.0166	**Rechazado**
BayesChaid	2.242	0.0243	0.025	**Rechazado**
BayesPSO	2.001	0.0453	0.05	**Rechazado**

4.4. Conclusiones parciales

Con la implementación de este algoritmo se logró determinar los operadores de exploración y contracción de la malla de soluciones, dentro de los cuales podemos contar con 'unión' y 'unión de diferencias' para el proceso de exploración y los operadores elitista y representativo para el de contracción de la malla. Se logró adaptar la metaheurística optimización de mallas variables (VMO) para el aprendizaje estructural de una red bayesiana BayesVMO(24,3). Donde utilizamos como métrica de score la precisión de la clasificación y las operaciones de VMO's también fueron adaptadas a este problema. Luego se llevaron a cabo experimentos utilizando 14 conjuntos de datos del repositorio de la UCI como punto de referencia. Los resultados de BayesVMO fueron comparados con 8 clasificadores del estado del arte y los análisis estadísticos presentan mejores resultados.

En un futuro cercano se pretende ampliar nuestra investigación para probar otros mecanismos que permitan crear nuevos nodos hacia el local y extremos globales (nuevos operadores). Además, se estudiará la forma de mejorar la limpieza del operador para regular los niveles de diversidad que introduce en la malla para saber de que manera influye en el comportamiento del algoritmo VMO.

5 Un nuevo clasificador bayesiano con aplicaciones al análisis de datos en problemas de educación

En este capítulo, en primer lugar se propone un nuevo clasificador bayesiano simple (CBS) que aprende forma rápida una frontera de Markov de la variable clase y una estructura de red que relaciona las variables de la clase y su frontera de Markov. Este modelo se compara con otros clasificadores bayesianos. A continuación, se considera el uso de modelos gráficos probabilísticos en el campo de la enseñanza para la realización del diagnóstico de estudiantes y poder determinar el problema de deserción estudiantil en las universidades, el mismo que ha sido ya estudiado por algunos investigadores: (Magaña, Montesinos, y Hernández2006) lo analiza haciendo uso de clúster, agrupando a individuos u objetos en conglomerados de acuerdo a sus semejanzas, maximizando la homogeneidad de los objetos dentro de los conglomerados a la vez que maximiza la heterogeneidad entre agregados. Otro caso de estudio para predecir la probabilidad de que un estudiante abandone la institución educativa se ha realizado utilizando técnicas de minería de datos; entre ellos tenemos a (García, Kuna, y Villatoro2010), quienes realizaron un trabajo basado en el uso del conocimiento, en reglas de asociación y en el enfoque TDIDT (*Top Down Induction of Decision Trees*) sobre la base de datos de la gestión académica del consorcio SIU de Argentina (que reúne 33 universidades de Argentina), lo cual permite un interesante análisis para encontrar las reglas de comportamiento.

(Lykourentzou et al.2009) usa un método de predicción de deserción en los cursos de e-learning, basado en tres técnicas populares de aprendizaje automático: redes neuronales feedforward, máquinas de soporte vectorial y métodos de ARTMAP difuso simplificado.

(Dekker, Pechenizkiy, y Vleeshouwers2009) comparan distintos modelos para predecir las tasas de abandono durante el primer semestre de los estudios de grado en la Universidad de Eindoven. Utilizan árboles de clasificación, naive Bayes, regresión logística, bosques de árboles, obteniendo unas tasas de acierto entre el 75 y el 80 por ciento.

También podemos citar el trabajo de (Porcel, Dapozo, y López2010), en el que se analiza la relación del rendimiento académico de los alumnos que ingresan a la Facultad de Ciencias Exactas y Naturales y Agrimensura de la Universidad Nacional del Nordeste (FACENA-UNNE) en

Corrientes, Argentina, durante el primer año de carrera con las características socioeducativas de los mismos. Se ajustó un modelo de regresión logística binaria, el cual clasificó adecuadamente el 75 por ciento de los datos.

Dentro de los problemas de diagnóstico, (García, Kuna, y Villatoro2010) proponen un modelo basado en redes bayesianas para determinar el estilo de aprendizaje de cada estudiante. La red se construye a partir de la información proporcionada por expertos (docentes). Dicho modelo es validado con alumnos obteniendo un alto grado de precisión.

Algunas aplicaciones van enfocadas al estudio de medidas de rendimiento colectivo, en lugar de centrarse en un estudiante. En esta línea podemos citar el trabajo de (Morales y Salmerón2003), en el que se propone una metodología para el análisis de relevancia de indicadores de rendimiento basada en el uso de redes bayesianas. En los últimos años ha cobrado gran importancia el uso de indicadores para describir el perfil de las universidades españolas en términos tanto académicos como investigadores y económicos. Estos indicadores son utilizados para tomar decisiones de gran importancia, llegando a afectar incluso a aspectos de financiación. Sin embargo, el número de indicadores a veces es excesivo, lo que aumenta el riesgo de redundancia y disfuncionalidad. Los modelos gráficos permiten obtener, de forma sencilla, las principales relaciones entre las variables a considerar. La metodología propuesta se aplica a un caso práctico, mostrando que es una herramienta útil para ayudar en la toma de decisiones en la elaboración de políticas basadas en indicadores de rendimiento. Esta tarea requiere el manejo de un alto número de variables de distintas naturalezas (cualitativas y cuantitativas), que pueden tener una compleja estructura de dependencias.

En este capítulo empezaremos con una sección 5.1 en la que introducimos un nuevo clasificador que llamaremos *clasificador bayesiano simple*, que será una red bayesiana genérica, pero aprendida con una técnica voraz.

En la sección 5.2 se realiza un análisis de deserción de los estudiantes legalmente matriculados en la Facultad de Ciencias de la Ingeniería de la Universidad Técnica Estatal de Quevedo en el periodo 2012-2013 en base a los datos socio económicos, (Oviedo et al.2015). En la subsección (5.2.1) se hace uso de la herramienta Weka para realizar la clasificación. Primero empezamos usando clasificadores bayesianos (5.2.1.1). En este caso se utiliza naive Bayes, BayesNet con K2 y un solo padre, BayesNet con K2 y un máximo de 5 padres, BayesNet con TAN, BayesNet con Hill Climber y un solo padre, BayesNet con Hill Climber y un máximo de 5 padres. Luego en la subsección (5.2.1.2), hacemos uso de los clasificadores de árboles, aquí se comparan los resultados obtenidos al hacer uso de árboles J48 y Random Forest. A continuación se analizan los datos con clasificadores que hacen uso de reglas de clasificación en la subsección (5.2.1.3). Con todos esos resultados realizamos un experimento con la herramienta Weka.

En la segunda parte de este capítulo; esto es la subsección (5.3) se analizan los datos de la UCI referente al rendimiento de los estudiantes secundarios que toman el curso de matemáticas en dos escuelas de Portugal y de acuerdo a algunos datos personales tratar de identificar a tiempo quienes aprueban el curso o quienes van a requerir ayuda para aprobarlo.

5.1. Un clasificador bayesiano simple

Supongamos un problema de clasificación supervisada con clase C y atributos $\mathbf{X} = (X_1, \ldots, X_n)$ y un conjunto de observaciones completas para la clase y los atributos $\mathscr{D}$. El objetivo es aprender una red bayesiana para un conjunto de variables $\mathbf{X}' \cup \{C\}$, donde $\mathbf{X}' \subseteq \mathbf{X}$. El método tendrá así un sistema implícito de selección de variables, descartando aquellas variables en $\mathbf{X} \setminus \mathbf{X}'$. Todas las variables seleccionadas estarán relacionadas con la variable clase C.

La idea es simple: se comienza introduciendo a C como nodo raíz en la red bayesiana $\mathscr{B}$ resultado. Se mantiene un conjunto de nodos $\mathbf{X}'$ de los atributos ya introducidos en la red (inicialmente vacía).

Se pueden usar distintas métricas $Score(X_i, \mathbf{A}|\mathscr{D})$ que miden la idoneidad de $\mathbf{A}$ como conjunto de padres de X_i (esta métrica puede ser BDEu, BIC, K2 o Akaike).

Se supone que tenemos un procedimiento PADRES(X_i,CANDIDATOS,Π_i) que calcula el mejor conjunto de padres Π_i de X_i con la métrica seleccionada y devuelve el valor de esa métrica óptima. La implementación actual es un algoritmo voraz que comienza con un conjunto Π_i vacío y va añadiendo o quitando de Π_i la variable que produce un mayor incremento de la métrica, hasta que ya no hay mejoras posibles.

En estas condiciones se calcula para cada variable $X_i \in \mathbf{X} \setminus \mathbf{X}'$ el valor:

$$Infor(X_i, C) = \text{PADRES}(X_i, \mathbf{X}' \cup \{C\}, \Pi_i) - \text{PADRES}(X_i, \mathbf{X}', \Pi_i')$$

Es decir $Infor(X_i, C)$ calcula la diferencia entre las mejores métricas de X_i con conjunto de padres elegido entre $\mathbf{X}'$ incluyendo C y sin incluir C entre los candidatos. Intuitivamente, es una medida de la dependencia condicional de X_i y C dadas las variables ya incluidas $\mathbf{X}'$. Este valor teóricamente siempre es mayor o igual a cero, pero podría ser negativo dado que el mejor conjunto de padres se calcula de forma aproximada.

Una vez calculado este valor para cada variable $X_i \in \mathbf{X} \setminus \mathbf{X}'$, se elige la variable $X_{max} = \arg\max_{X_i \in \mathbf{X} \setminus \mathbf{X}'} Infor(X_i, C)$. Esta sería la variable que más informa sobre la clase C condicionado a las variables ya introducidas. Si $Infor(X_{max}, C) > 0$, entonces esta variable aporta información adicional sobre C y se inserta en la red y en $\mathbf{X}'$. Su conjunto de padres es Π_i calculado con PADRES($X_{max}, \mathbf{X}' \cup \{C\}, \Pi_i$). En teoría, siempre $C \in \Pi_i$, ya que en otro caso $Infor(X_{max}, C) = 0$, aunque debido a la naturaleza voraz del procedimiento podría ser que C $notin \Pi_i$, aunque esta posibilidad sería rara.

El proceso continúa de forma iterativa tratando de añadir una nueva variable y termina cuando en un paso $Infor(X_{max}, C) \leq 0$.

Las principales características de este clasificador son:

- Aprende una red bayesiana arbitraria con un subconjunto de las variables iniciales que influyen directamente en esta variable. En este sentido se puede considerar como un algoritmo que calcula una frontera de Markov, ya que trata de conseguir un conjunto de

variables tales que dadas estas variables, el resto de las variables son independientes.

- La variable clase es siempre un nodo raíz. y hay enlaces desde este nodo al resto de los atributos /excepto en raras ocasiones debido a la naturaleza aproximada del cálculo de los padres). En este sentido se parece a los clasificadores bayesianos en los que siempre existe un enlace de la clase a cada uno de los atributos.
- El orden de los atributos se basa en elegir de forma voraz primero los atributos que más informan sobre la clase, dados los atributos ya seleccionados. De esta forma se introducen primero los atributos más relevantes. No se busca en el espacio de los órdenes de los atributos para obtener la red con mejor métrica, sino que se concentra sólo en obtener la máxima información para la clase. En este sentido puede haber alguna pérdida en la calidad de la red, pero el algoritmo gana en rapidez.

5.1.1. Experimentos con bases de datos UCI

Durante algún tiempo se ha considerado que los algoritmos para aprendizaje sin restricciones de redes bayesianas, especialmente los basados en el paradigma de métrica+búsqueda, no son adecuados para la construcción competitiva de clasificadores basados en redes bayesianas (Acid, De Campos, y Castellano2005). Actualmente esa percepción está cambiando debido al desarrollo de métodos de aprendizaje de redes genéricas que son muy competitivos (Acid, De Campos, y Castellano2005).

En esta sección se realizan pruebas experimentales para lo cual se han utilizado 31 bases de datos bien conocidas de la UCI (Lichman2013) y dos bases de variables artificiales. Las bases de datos se las puede visualizar en la tabla 5.1. Con estas bases de datos se compararan los resultados que obtienen algoritmos estudiados en el estado del arte como Naive Bayes, TAN, BAN, SBND y las combinaciones con diferentes métricas como K2, BIC, Akaike, BDe. También se comparan los métodos RPDAG y C-RPDAG ya analizados en la sección 2.3.5 de esta tesis. Estos métodos construyen clasificadores que son redes bayesianas genéricas equivalentes en independencia y equivalentes en clasificación. El trabajo experimental fue realizado en Elvira (Elvira2002).

La tabla 5.1 da una breve descripción de las características de cada base de datos, incluyendo el número de instancias, los atributos y los estados para la variable clase. Estos conjuntos de datos han sido preprocesados de la siguiente manera: las variables continuas se han discretizado usando el procedimiento propuesto por (Irani et al.1993), y las instancias con valores no definidos o perdidos fueron eliminados. Para esta etapa de preprocesamiento, se han utilizado los resultados de (Acid, De Campos, y Castellano2005).

Tabla 5.1 Descripción de las bases de datos

Database	Instancias	Atributos	Clases
adult-d-nm	45222	14	2
australian-d	690	14	2
breast-no-missing	682	10	2
car	1728	6	4
chess	3196	36	2
cleve-no-missing-d	296	13	2
corral-d	128	6	2
crx-no-missing-d	653	15	2
diabetes-d-nm	768	8	2
DNA-nominal	3186	60	3
flare-d	1066	10	2
german-d	1000	20	2
glass2-d	163	9	2
glass-d	214	9	7
heart-d	270	13	2
hepatitis-no-missing-d	80	19	2
iris-d	150	4	3
letter	20000	16	26
lymphography	148	18	4
mofn-3-7-10-d	1324	10	2
nursery	12960	8	5
mushroom	8124	22	2
pima-d	768	8	2
satimage-d	6435	36	6
segment-d	2310	19	7
shuttle-small-d	5800	9	7
soybean-large-no-missing-d	562	35	19
splice.dbc	3190	60	3
vehicle-d-nm	846	18	4
vote	435	16	2
waveform-21-d	5000	21	3

Los resultados obtenidos por cada uno de los clasificadores y sus combinaciones con las métricas estudiadas pueden ser observados en las tablas 5.2, 5.3, 5.4, 5.5 (por el tamaño de las tablas se han dividido en 4).

Tabla 5.2 Resultados con base de datos UCI

Database	SBND BDE	SBND BIC	SBND Ak	SBND K2	TAN	NBayes
adult-d-nm	85.255	85.213	85.405	85.963	85.295	83.090
australian-d	86.667	85.797	86.232	86.667	85.362	85.652
breast-no-missing	97.370	97.662	96.053	97.515	96.196	97.662
car	94.097	85.067	93.635	94.097	94.214	85.299
chess	97.496	96.214	97.653	97.121	91.989	87.765
cleve-no-missing-d	82.115	82.126	80.759	82.103	79.724	82.414
corral-d	99.231	100.0	99.167	99.231	99.231	85.962
crx-no-missing-d	85.916	86.375	86.671	87.140	86.974	86.678
diabetes-d-nm	78.780	79.040	78.908	79.429	77.997	77.341
DNA-nominal	96.171	96.203	93.943	95.983	94.822	95.418
flare-d	82.268	82.268	82.738	82.268	83.018	80.395
german-d	74.0	73.6	71.8	72.3	73.6	75.5
glass2-d	85.882	81.066	84.007	85.882	85.257	83.493
glass-d	73.355	64.545	71.494	69.134	73.852	73.853
heart-d	81.111	81.111	82.593	82.963	82.963	83.333
hepatitis-no-missing-d	90.0	90.0	90.0	86.25	86.25	85.0
iris-d	94.0	95.333	95.333	94.667	94.0	94.667
letter	81.565	74.015	85.655	85.36	86.320	73.6
letter-d	84.320	74.365	84.81	84.615	85.775	73.935
lymphography	77.571	81.620	80.381	80.381	79.048	81.762
mofn-3-7-10-d	92.522	90.790	100.0	93.501	91.237	85.425
nursery	91.890	91.705	97.469	94.537	92.261	90.332
mushroom	100.0	98.523	99.274	100.0	99.963	95.495
pima-d	79.166	79.560	78.259	79.429	79.038	77.994
satimage-d	85.144	82.316	86.807	85.812	88.252	82.440
segment-d	94.459	93.550	95.022	94.372	95.151	92.208
shuttle-small-d	99.741	99.552	99.534	99.759	99.069	99.052
soybean-large-no-missing-d	91.278	84.496	90.918	93.409	94.298	91.269
splice.dbc	96.238	96.270	91.944	96.364	94.796	95.454
vehicle-d-nm	65.849	65.013	70.451	64.892	69.986	61.829
vote	94.715	94.952	95.640	95.174	94.498	90.338
vote-no-missing	95.396	95.375	94.704	95.169	94.244	90.095
waveform-21-d	82.84	82.34	81.9	83.5	83.1	81.84
media	87.770	86.244	88.156	88.030	87.810	85.048

Luego se realizaron varios test no paramétricos sobre las diferencias entre los distintos métodos para determinar el algoritmo que mejor clasifica. Adicionalmente se ha incluido el valor de la media para cada uno de los algoritmos y se puede determinar que la mejor media la consigue CRPDAG-BDe con un valor de 88.354 seguido SBND Akaike con 88.156.

Dentro de los test no paramétricos básicos vamos a utilizar Friedman ya que se tiene más de 2 muestras asociadas.

La hipótesis nula (H_0) que se contrasta es que las respuestas asociadas a cada uno de los 'tratamientos' tienen la misma distribución de probabilidad o distribuciones con la misma mediana, frente a la hipótesis alternativa de que por lo menos la distribución de una de las respuestas

Tabla 5.3 Resultados con base de datos UCI

Database	BAN Learning BDe	BAN Learning BIC	BAN Learning K2
adult-d-nm	85.534	85.472	84.906
australian-d	84.638	86.812	84.203
breast-no-missing	97.662	97.662	97.664
car	93.517	85.414	94.213
chess	96.151	95.745	96.995
cleve-no-missing-d	81.069	79.713	78.701
corral-d	100.0	100.0	98.462
crx-no-missing-d	86.063	86.986	84.681
diabetes-d-nm	78.387	78.389	78.127
DNA-nominal	95.292	95.418	93.158
flare-d	82.830	82.831	83.298
german-d	75.3	75.3	74.1
glass2-d	85.882	85.221	85.846
glass-d	72.013	76.190	74.762
heart-d	82.963	82.222	82.963
hepatitis-no-missing-d	88.75	87.5	88.75
iris-d	94.0	94.0	94.0
letter	84.715	74.880	87.965
letter-d	85.56	77.34	86.945
lymphography	85.0	82.333	74.857
mofn-3-7-10-d	87.617	90.865	93.804
nursery	91.860	91.883	94.892
mushroom	100.0	100.0	100.0
pima-d	78.775	78.780	79.040
satimage-d	88.361	85.175	87.506
segment-d	95.281	92.900	95.195
shuttle-small-d	99.052	99.776	99.741
soybean-large-no-missing-d	93.424	93.418	89.860
splice	95.329	95.705	94.107
vehicle-d-nm	70.689	70.102	69.384
vote	94.493	93.811	92.659
vote-no-missing	93.092	93.552	92.875
waveform-21-d	82.94	82.62	83.5
media	88.090	87.211	87.791

difiere de las demás.

Los valores con los que se van a realizar estas pruebas se puede visualizar en la tabla 5.6. En dicha tabla se puede ver el orden promedio de los algoritmos. Aquí se puede determinar que el algoritmo con mejor comportamiento es SBND K2.

Los resultados del test de Friedman se encuentra en la tabla 5.7, donde visualizamos que valor es menor que $0,05$, por lo que la hipótesis nula es rechazada y se determina que las diferencias miden las distribuciones de los distintos métodos que son estadísticamente significativas.

Tabla 5.4 Resultados con base de datos UCI

Database	RPDag-BDe	RPDag-BIC	RPDag-K2
adult-d-nm	85.748	85.576	85.339
australian-d	85.797	85.362	85.797
breast-no-missing	97.662	97.662	97.662
car	93.228	85.878	94.040
chess	97.152	94.931	96.871
cleve-no-missing-d	82.115	81.770	80.425
corral-d	100.0	100.0	100.0
crx-no-missing-d	86.371	86.068	84.387
diabetes-d-nm	79.429	79.040	79.170
DNA-nominal	95.857	96.360	95.450
flare-d	82.268	82.268	82.268
german-d	74.4	74.2	73.8
glass2-d	84.632	84.044	82.169
glass-d	67.294	65.823	73.788
heart-d	80.370	81.481	82.963
hepatitis-no-missing-d	87.5	90.0	86.25
iris-d	96.0	95.333	94.667
letter	83.185	74.835	86.65
letter-d	86.085	74.87	86.325
lymphography	76.905	75.524	74.952
mofn-3-7-10-d	100.0	93.808	96.829
mushroom	100.0	100.0	100.0
nursery	93.465	91.312	94.792
pima-d	79.299	79.299	79.301
satimage-d	84.911	79.285	84.911
segment-d	94.199	94.589	95.325
shuttle-small-d	99.690	94.862	99.534
soybean-large-no-missing-d	89.148	86.096	93.064
splice	95.956	96.238	96.363
vehicle-d-nm	64.902	61.584	64.066
vote	94.720	94.947	95.185
vote-no-missing	94.709	95.153	93.092
waveform-21-d	79.98	81.06	83.320
media	87.666	86.038	87.841

Si las diferencias detectadas son significativas, se aplica la prueba de Holm´s para comparar el algoritmo de control (el mejor clasificado) con los restantes. Holm´s es una prueba de comparación múltiple mediante el cual vamos a confrontar el algoritmo SBND con K2 que es el de mejor valor de clasificación obtuvo con el resto de algoritmos.

En la tabla 5.8 se puede observar los resultados con el test de Holm para un nivel de significancia de 0.05 y en la tabla 5.9 con un nivel de significancia de 0.10

Tabla 5.5 Resultados con base de datos UCI

Database	CRPDag-BDe	CRPDag-BIC	CRPDag-K2
adult-d-nm	85.257	85.463	85.372
australian-d	86.667	86.232	84.348
breast-no-missing	97.662	97.662	97.664
car	93.228	85.878	94.040
chess	96.621	95.713	96.277
cleve-no-missing-d	81.747	81.057	78.724
corral-d	100.0	100.0	100.0
crx-no-missing-d	86.981	87.135	83.308
diabetes-d-nm	78.387	77.996	78.127
DNA-nominal	96.422	96.202	83.519
flare-d	83.020	82.830	82.738
german-d	73.4	74.1	74.0
glass2-d	86.471	85.221	85.882
glass-d	74.329	70.519	73.377
heart-d	82.593	82.222	82.963
hepatitis-no-missing-d	90.0	83.75	86.25
iris-d	94.0	94.0	94.0
letter	83.845	75.135	87.01
letter-d	86.55	77.34	87.195
lymphography	76.905	80.333	76.333
mofn-3-7-10-d	100.0	93.808	96.829
mushroom	100.0	100.0	100.0
nursery	93.465	91.312	94.792
pima-d	78.775	78.910	78.910
satimage-d	87.553	85.175	86.667
segment-d	95.325	92.900	94.978
shuttle-small-d	99.741	99.707	99.707
soybean-large-no-missing-d	89.859	90.025	93.590
splice	96.332	96.332	90.157
vehicle-d-nm	70.929	72.228	71.517
vote	93.811	93.346	93.351
vote-no-missing	92.854	92.860	93.330
waveform-21-d	82.94	82.74	83.46
media	88.354	86.913	87.528

En primer lugar se consideró $\alpha = 0.05$. El valor de P en la prueba de Holms es $P \leq 0.0045$. Con este valor comparamos con el resto de algoritmos basándonos en la columna de la derecha de la tabla 5.8. Se puede entonces observar que este algoritmo es significativamente mejor que Naive Bayes, SBND BIC, RPDag Leraning BIC y no hay diferencias significativas con el resto de algoritmos.

En segundo lugar, se considera un nivel de significación $\alpha = 0.10$. El valor de P en la prueba

Tabla 5.6 Puntuación promedio de los algoritmos

Algoritmo	Ranking
SBND BDE	7.676470588235294
SBND BIC	9.264705882352944
SBND Akaike	7.73529411764706
SBND K2	5.955882352941175
TAN	8.249999999999998
NaiveBayes	11.382352941176473
BAN Learning BDe	7.058823529411763
BAN Learning BIC	7.955882352941175
BAN Learning K2	7.823529411764705
RPDag Learning BDe	7.661764705882354
RPDag Learning BIC	9.20588235294118
RPDag Learning K2	7.191176470588233
CRPDag Learning BDe	6.2647058823529385
CRPDag Learning BIC	8.823529411764708
CRPDag Learning K2	7.749999999999998

Tabla 5.7 Resultados de la prueba de Friedman

Prueba	**Valor P**	**Hipótesis**
Friedman	$1{,}542E-4$	Rechazada

Tabla 5.8 Holm Tabla para $\alpha = 0.05$

i	algoritmo	$z = (R_0 - R_i)/SE$	p	Holm
14	NaiveBayes	5.0029586834427615	5.645704442401309E-7	0.0035714285714285718
13	SBND BIC	3.0505845630748563	0.0022839635380862903	0.0038461538461538464
12	RPDag Learning BIC	2.9963519486201924	0.0027323088004595057	0.004166666666666667
11	CRPDag Learning BIC	2.643839954664876	0.00819714047525547	0.004545454545454546
10	TAN	2.1150719637318978	0.03442381432538883	0.005
9	BAN Learning BIC	1.8439088914585775	0.06519641907813004	0.005555555555555556
8	BAN Learning K2	1.7218855089355838	0.08509026052283775	0.00625
7	CRPDag Learning K2	1.6540947408672533	0.09810826450210172	0.0071428571428571435
6	SBND Akaike	1.6405365872535898	0.10089364763218327	0.008333333333333333
5	SBND BDE	1.586303972798925	0.1126703715408176	0.01
4	RPDag Learning BDe	1.57274581918526	0.11577768575893127	0.0125
3	RPDag Learning K2	1.1388849035479442	0.2547511629904382	0.016666666666666666
2	BAN Learning BDe	1.0168615210249505	0.3092193106086886	0.025
1	CRPDag Learning BDe	0.28472122588698523	0.7758577275237244	0.05

de Holms es $P \leq 0.01$. Con este valor se realizan comparaciones múltiples con los valores de la tabla 5.9 basándonos en la columna de la derecha. Se puede determinar que nuestro algoritmo de control SBND con K2 clasifica mejor que los algoritmos Naive Bayes, SBND BIC, RPDag Learning BIC, CRPDag Learning BIC y no hay diferencias significativas con el resto de algoritmos.

Tabla 5.9 Holm tabla para $\alpha = 0.10$

i	algorithm	$z = (R_0 - R_i)/SE$	p	Holm
14	NaiveBayes	5.0029586834427615	5.645704442401309E-7	0.0071428571428571435
13	SBND BIC	3.0505845630748563	0.0022839635380862903	0.007692307692307693
12	RPDag Learning BIC	2.9963519486201924	0.0027323088004595057	0.008333333333333333
11	CRPDag Learning BIC	2.643839954664876	0.00819714047525547	0.009090909090909092
10	TAN	2.1150719637318978	0.03442381432538883	0.01
9	BAN Learning BIC	1.8439088914585775	0.06519641907813004	0.011111111111111112
8	BAN Learning K2	1.7218855089355838	0.08509026052283775	0.0125
7	CRPDag Learning K2	1.6540947408672533	0.09810826450210172	0.014285714285714287
6	SBND Akaike	1.6405365872535898	0.10089364763218327	0.016666666666666666
5	SBND BDE	1.586303972798925	0.1126703715408176	0.02
4	RPDag Learning BDe	1.57274581918526	0.11577768575893127	0.025
3	RPDag Learning K2	1.1388849035479442	0.2547511629904382	0.03333333333333333
2	BAN Learning BDe	1.0168615210249505	0.3092193106086886	0.05
1	CRPDag Learning BDe	0.28472122588698523	0.7758577275237244	0.1

5.2. Análisis de deserción en la FCI-UTEQ

En esta sección se desarrollan experimentos con una base de conocimientos de 773 estudiantes matriculados en el periodo 2012-2013 en la Facultad de Ciencias de la Ingeniería de la Universidad Técnica Estatal de Quevedo, de los cuales se han obtenido sus datos socio-económicos y académicos para poder ser usados en clasificación con la herramienta Weka (Garner1995). Las variables se ilustran en la tabla 3.2. Los diferentes valores que asumen cada una de esas variables, se muestran en las tablas 5.10, 5.11, 5.12,5.13.

Tabla 5.10 Valores y descripción de la variable carreras

Valor	Descripción
FI024	Ingeniería en Sistemas
FI025	Ingeniería en Diseño Gráfico
FI026	Ingeniería Mecánica
FI027	Ingeniería Industrial
FI028	Ingeniería en Telemática
FI029	Ingeniería Eléctrica
FI030	Ingeniería Agroindustrial
FI031	Ingeniería en Seguridad Industrial y Salud Ocupacional

Tabla 5.11 Valores y descripción de la variable cursos

Valor	Descripción
1	Primero
2	Segundo
3	Tercero
4	Cuarto
5	Quinto

Tabla 5.12 Variables y descripción de valores

Variable	Descripción
D	SI=1; NO=0
F	SI=1; NO=0
I	SI=1; NO=0
J	SI=1; NO=0
K	SI=1; NO=0
L	SI=1; NO=0
M	SI=1; NO=0
N	SI=1; NO=0
O	SI=1; NO=0
P	SI=1; NO=0
Q	SI=1; NO=0
R	SI=1; NO=0
S	SI=1; NO=0

Tabla 5.13 Variables y consideraciones a discretizar

Variable	Descripción
E	X<200=0;200>X<800=1;X>800=2
G	MEDIA AGUA=0;CASA/VILLA=1;DEPARTAMENTO=2; CUARTO DE INQUILINATO=3;OTRA=4;RANCHO=5
H	PADRE Y MADRE=0;PADRE=1;MADRE=2;OTRO PARIENTE=3;OTRO=4

5.2.1. Clasificación usando Weka

Weka es una colección de herramientas de visualización y algoritmos de inferencia y clasificación con librerías en JAVA que permite la extracción de conocimiento desde bases de datos. Permite trabajar con distintas herramientas de minería de datos como reglas de asociación, agrupación, clasificación y regresión. Podemos visualizar en las figuras 5.1, los resultados obtenidos por cada uno de los atributos en referencia a la clase *S* deserta. De forma visual, ya se puede observar que no hay variables que de forma individual den información significativa sobre la deserción. De acuerdo a la figura 5.1 se ha construido las tablas 5.14, 5.15 y 5.16, en el que podemos encontrar el análisis descriptivo de las variables donde se puede visualizar el porcentaje de cada uno de sus valores.

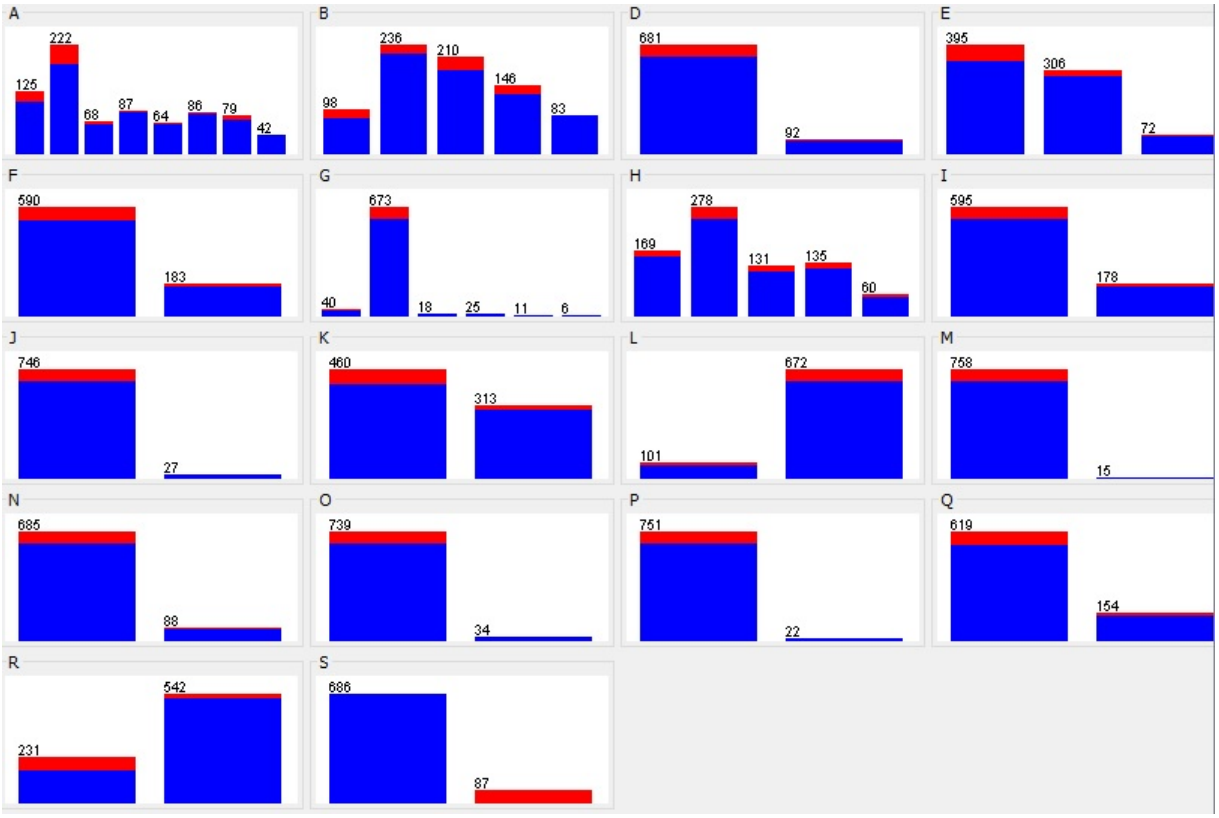

Figura 5.1 Resultados obtenidos por cada uno de los atributos en referencia a la clase

Tabla 5.14 Variables y análisis descriptivo

Variable	Descripción	Cantidad	Porcentaje
A	FCI024	125	16.17
	FCI025	222	28.72
	FCI026	68	8.80
	FCI027	87	11.25
	FCI028	64	8.28
	FCI029	86	11.13
	FCI030	79	10.22
	FCI031	42	5.43
B	1	98	12.68
	2	236	30.53
	3	210	27.17
	4	146	18.89
	5	83	10.74
D	no	681	88.10
	si	92	11.90

Tabla 5.15 Variables y análisis descriptivo

Variable	Descripción	Cantidad	Porcentaje
E	X<200	395	51.10
	200>X<800	306	39.59
	X>800	72	9.31
F	no	590	76.33
	si	183	23.67
G	media agua	40	5.17
	casa villa	673	87.06
	departamento	18	2.33
	cuarto inquilinato	25	3.23
	otro	11	1.42
	rancho	6	0.78
H	padre y madre	169	21.87
	padre	278	35.97
	madre	131	16.95
	otro pariente	135	17.46
	otro	60	7.76
I	no	595	76.97
	si	178	23.03
J	no	746	96.51
	si	27	3.49
K	no	460	59.51
	si	313	40.49
L	no	101	13.07
	si	672	86.93
M	no	758	98.06
	si	15	1.94
N	no	685	88.62
	si	88	11.38
O	no	739	95.60
	si	34	4.40
P	no	751	97.15
	si	22	2.75

Tabla 5.16 Variables y análisis descriptivo

Variable	Descripción	Cantidad	Porcentaje
Q	no	619	80,08
	si	154	19.92
R	no	231	29.89
	si	542	70.11
S	no	686	88.75
	si	87	11.25

5.2.1.1. Usando clasificadores bayesianos

Se obtuvieron resultados usando como clasificadores Naive Bayes y BayesNet con diferentes alternativas como K2, TAN, Hill Climber. con un padre y también con un máximo de 5 padres.

Tabla 5.17 Resultados obtenidos con los diferentes clasificadores

Clasificador	Clasificados correctamente	Tasa TN	Tasa TP
NaiveBayes	88.0983	0.965	0.218
BayesNet con K2-1 padre	87.9690	0.964	0.218
BayesNet con K2-5 padres	91.0737	0.985	0.322
BayesNet con TAN	89.9094	0.974	0.318
BayesNet con Hill Climber-1	88.6158	0.974	0.195
BayesNet con Hill Climber-5	89.6507	0.975	0.276

Para obtener estos valores en la herramienta Weka hemos clasificado usando una validación cruzada de 10. Como se puede observar en la tabla 5.17 se ha trabajado con 773 casos, de los cuáles BayesNet con K2 y máximo 5 padres es el que mejor ha clasificado correctamente (91.0737 por ciento) y adicionalmente nos indica la tasa de verdadero negativo (TN) y la tasa de verdadero positivo (TP). Podemos observar que hay un 32.20 por ciento de sensibilidad. Este es el porcentaje de estudiantes que se han clasificado correctamente entre aquellos que desertan. Éstos son los que deberían de recibir alguna atención y sobre los que habría que aplicar acciones especiales para disminuir este índice. Aunque no es una tasa muy alta, es importante señalar que es un problema difícil de predecir y por este procedimiento se detectan principalmente la tercera parte de los estudiantes que desertan. Además, el costo en términos de falsos positivos es muy bajo.

La tasa de verdaderos negativos o especifidad corresponde a la probabilidad de que un estudiante que esté bien en su proceso académico tenga un resultado negativo en la prueba. En este caso solo se llega a detectar como falsos positivos un 1.5 por ciento $(1 - 0.985)$.

En la figura 5.2 podemos visualizar que todas las variables están relacionadas directamente

con la clase deserción (S). La variable curso (B) depende también de la carrera (A) e influyen sobre el resultado académico (R). Por otro lado, se puede también considerar que la variable contar con servicio de tv cable (I) influye directamente sobre (J,K,M), servicio de plan celular (N), y éstas sobre servicio de vehículo propio (O) y trabajar actualmente (Q).

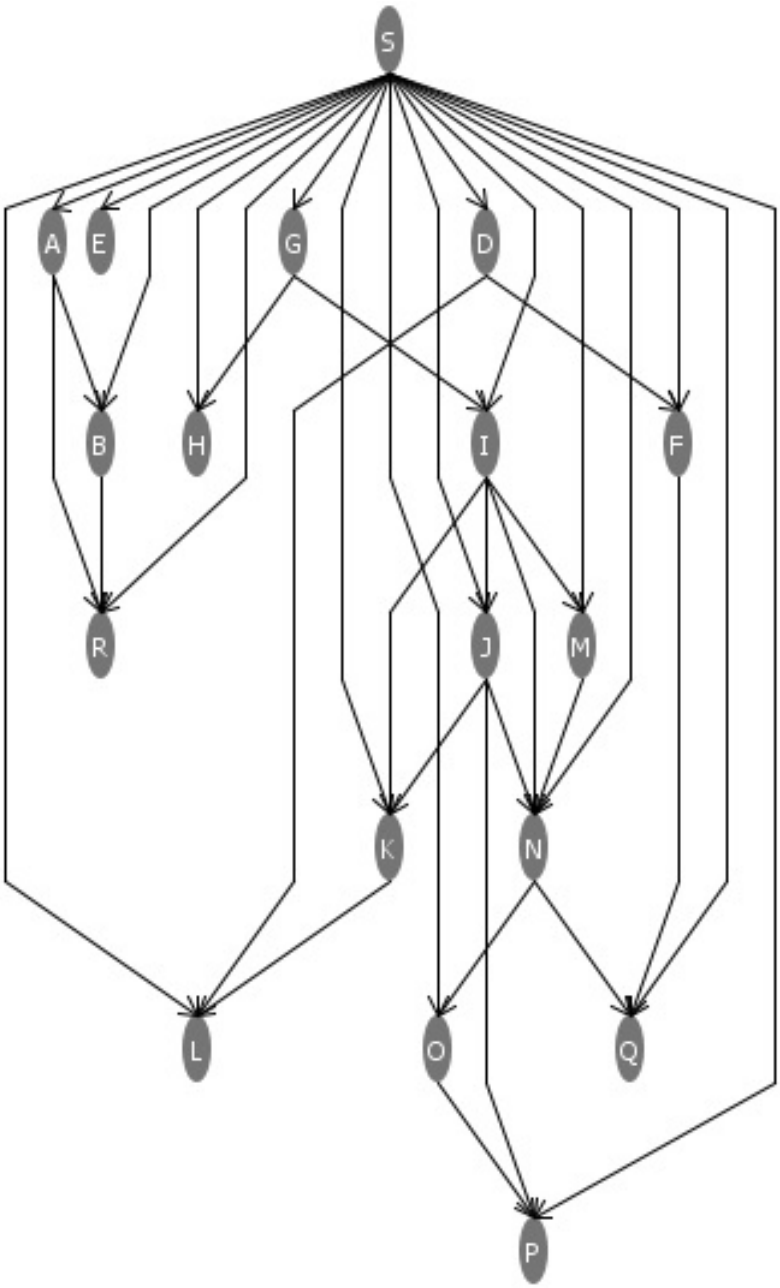

Figura 5.2 Red obtenida con clasificador BayesNet con K2 y un máximo de 5 padres

En la figura 5.3 podemos visualizar la dependencia de las variables con la clase S; así como también que el curso (B) y la variable trabaja actualmente (Q) van a estar dependiendo directamente de la variable carrera (A). El TAN considera sólo las relaciones más relevantes. Entre ellas la dependencia de la carrera para saber si se aprueba o no el curso (R), de igual manera que si el estudiante trabaja (Q) vivirá en un domicilio diferente al de la familia (F).

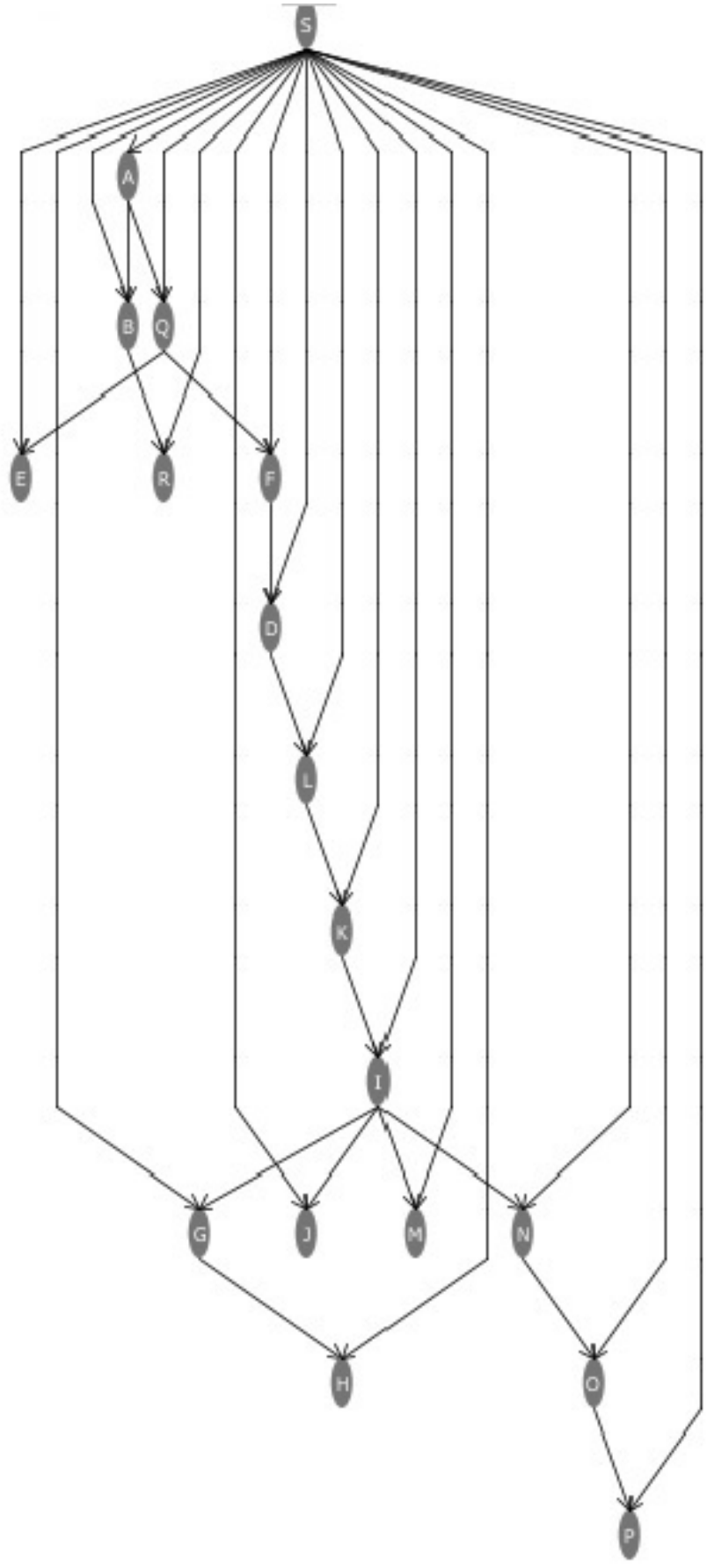

Figura 5.3 Red obtenida con clasificador BayesNet con TAN

En la figura 5.4 podemos visualizar la red obtenida con con BayesNet y un solo padre. Hay influencia de 5 variables carrera (A), curso (B), costo de educación (E), servicio de internet (K) y aprueba (R) de manera directa con la clase deserta (S); así como también la de la variable estudiante trabaja actualmente (Q) con carrera (A). También se demuestra una dependencia fuerte de la variable tener servicio de internet (K) con las variables servicios básicos (L) y con servicio de tv cable (I).

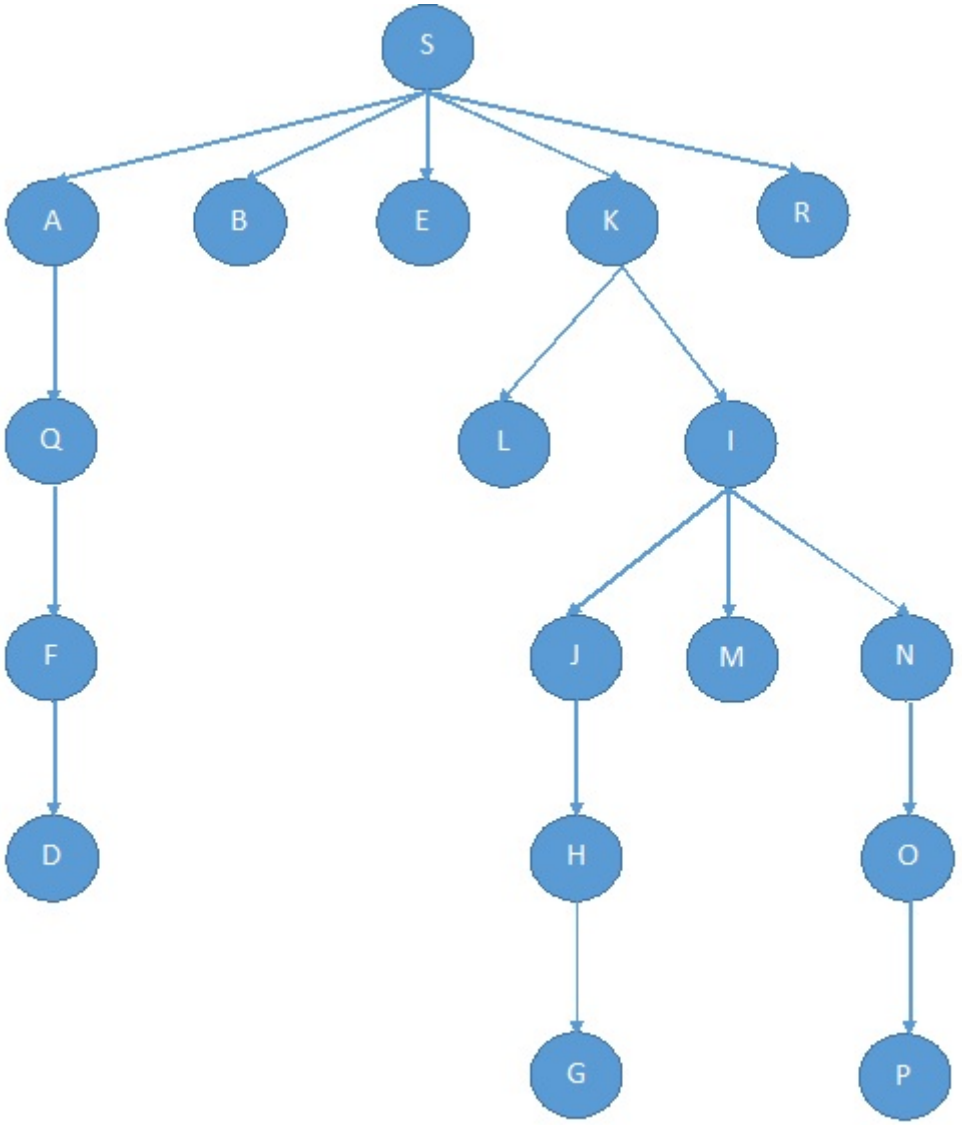

Figura 5.4 Red obtenida con clasificador BayesNet con HILL CLIMBER y un solo padre

En la figura 5.5 podemos visualizar la dependencia de las variables carrera (A), curso (B), costo de educación (E), servicio de internet (K) y aprueba (R) con la clase deserta (S); se sigue manteniendo la dependencia tanto de servicios básicos (L) como servicio de plan celular (N) de servicio de acceso a internet(K) y está de servicio de tarjeta de crédito (J).

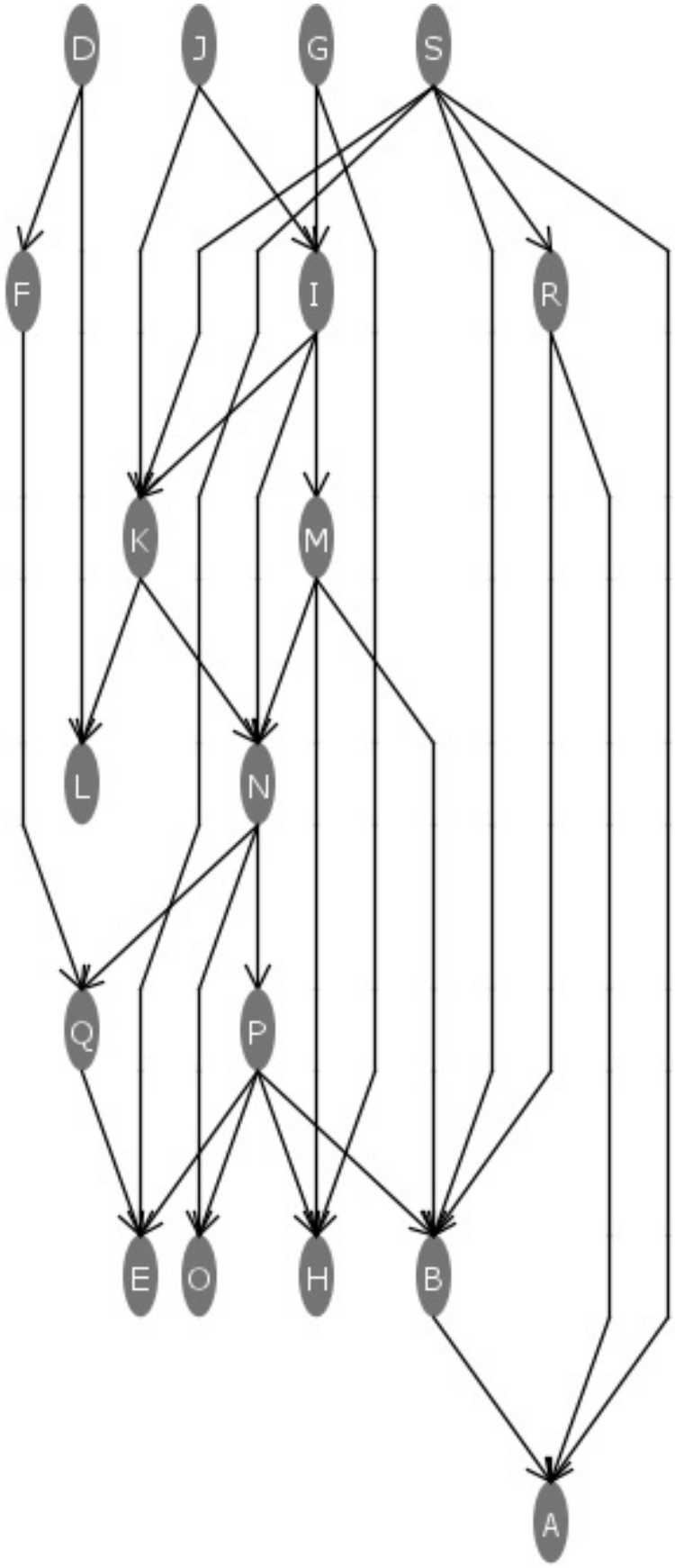

Figura 5.5 Red obtenida con clasificador BayesNet con HILL CLIMBER y un máximo de 5 hijos

5.2.1.2. Usando clasificadores de árboles

Los resultados que se obtuvieron usando como clasificadores de árboles J48 y Random Forest pueden ser visualizados en la tabla 5.18.

Tabla 5.18 Resultados obtenidos con clasificadores de árboles

Clasificador	Clasificados correctamente	Tasa TN	Tasa TP
J48	88.2277	0.993	0.011
RandomForest	89.6507	0.978	0.253

Como se puede observar trabajando con un clasificador de árbol J48 los casos clasificados correctamente equivalen al 88.2277 por ciento. Además se indica el porcentaje de sensibilidad y especifidad. Estos valores no mejoran al del clasificador BayesNet con K2 y 5 padres. De igual manera se puede observar que trabajando con un clasificador de árbol Random Forest, los casos clasificados correctamente mejoran en referencia a J48. Se debe indicar que es un bosque aleatorio de 100 árboles de los cuáles cada uno está construido con 5 características.

5.2.1.3. Usando reglas de clasificación

Los resultados usando como reglas de clasificación ZeroR y tablas de decisiones se pueden ver en la tabla 5.19.

Tabla 5.19 Resultados obtenidos con diferentes reglas de clasificación

Clasificador	Clasificados correctamente	Tasa TN	Tasa TP
ZeroR	88.7451	1	0
Tabla de decisiones	89.0039	0.981	0.172

Como se puede observar trabajando con reglas de clasificación ZeroR que corresponde a clasificadores triviales que siempre responden a la clase más frecuente ya sea negativa o positiva, los casos clasificados correctamente equivalen al 88.7451 por ciento. En el caso del trabajo con tablas de decisiones, los casos clasificados correctamente mejoran en referencia a ZeroR.

Al igual que en la sección 5.1.1 se realizará una comparación de los resultados obtenidos con los diferentes algoritmos con la base de datos de variables socio-económicas de los estudiantes de la Facultad de Ciencias de la Ingeniería de la Universidad Técnica Estatal de Quevedo tal como se indica en la tabla 5.20.

Tabla 5.20 Resultados con base de datos de estudiantes de la UTEQ

Datos	SBND1	SBND2	SBND3	SBND4	BAN BDe	BAN BIC
socioeconomico	88.232	88.745	88.357	89.910	87.581	87.711
Datos	BAN K2	RPDag BDe	RPDag BIC	RPDag K2	TAN	NaiveBayes
socioeconomico	87.723	87.584	87.972	89.657	88.743	87.456

Como se puede observar el algoritmo que entrega mejores resultados es SBND2 con mucha diferencia de los otros con los que se han comparado y que los de peor resultado son RPDag con métrica K2 y BAN con la métrica BIC.

En virtud de que valor que se obtiene con el test de Friedman es mayor que 0.05, la hipótesis nula no es rechazada y se determina que no hay diferencias significativas entre las distribuciones y por lo tanto no es necesario seguir realizando pruebas. Estos resultados se originan en virtud de que se han comparado pocas bases de datos.

5.3. Análisis con base de datos de la UCI

A continuación se analiza un conjuntos de datos de la UCI, que hace referencia al rendimiento de los estudiantes de educación secundaria en dos escuelas portuguesas (Cortez y Goncálves2008).

Por un lado, los atributos de los datos incluyen las calificaciones del estudiante, características, situación social y situación demográfica. Esta información fue obtenida mediante el uso de informes de la escuela y algunas encuestas tipo cuestionario.

Los conjuntos de datos proporcionados tienen referencia al rendimiento de dos temáticas diferentes, por un lado los que cursan matemáticas, y los que estudian portugués (Cortez y Goncálves2008). El conjunto de datos fue modelado en virtud de las tareas de clasificación y regresión binaria a cinco niveles.

El objetivo final $G3$ tiene una fuerte correlación con los atributos $G2$ y $G1$, ya que $G3$ es la nota final, mientras que G1 y G2 corresponden a las notas en 1er y 2do grado. Por este motivo, las variables $G1$ y $G2$ se han eliminado de nuestro estudio.

En las tablas 5.21, 5.22, 5.23 y 5.24 podemos visualizar las variables con sus propuestas de discretización, cumplimiento, temáticas e información de los atributos.

Luego se realizará el mismo entrenamiento que se hizo con los datos socio-económicos de los estudiantes de la Facultad de Ciencias de la Ingeniería de la Universidad Técnica Estatal de Quevedo; pero, ahora analizando el nivel educativo de la población portuguesa en las clases básicas de matemáticas y de portugués.

(Cortez y Goncálves2008) hace referencia a las mejoras del nivel educativo portugués, donde indica que Portugal aún se encuentra en el último puesto de Europa debido al alto número de fracasos y la deserción de los estudiantes. De ahí la importancia de este estudio.

Tabla 5.21 Variables y descripción de cumplimiento en UCI

Variable	Descripción
A	GP=1; MS=0
B	Femenino=1; Masculino=0
D	Urbano=1; Rural=0
F	SI=1; NO=0
L	Madre=1; Padre=0

Tabla 5.22 Grados relacionados con las temáticas de los cursos

Grado	Descripción	Tipo	Observación
G3	Grado final	Numérico	<=10 pierde; >10 cumple la meta

Tabla 5.23 Variables e información de los atributos

Variable	Nombre	Descripción
A	Escuela	Escuela del estudiante
B	Sexo	Sexo del estudiante
C	Edad	Edad del estudiante
D	Dirección	Tipo de dirección de la casa del estudiante
E	Famtam	Tamaño de la familia del estudiante
F	Pestado	Convive con los padres
G	Medu	Grado de educación de la madre
H	Pedu	Grado de educación del padre
I	Mtrabajo	Trabajo de la madre
J	Ftrabajo	Trabajo del padre
K	Razón	Razón por la que se cambia de escuela
L	Representante	Representante del estudiante
M	Ttraslado	Tiempo que se demora en ir del hogar a la escuela
N	Testudio	Tiempo que se dedica a estudiar en la semana
O	Fracasos	Número de fracasos en clases anteriores
P	Eapoyo	Apoyo extra educacional
Q	Fapoyo	Apoyo educacional familiar
R	Pago	Pago de clases adicionales fuera de la temática del curso
S	Actividades	Actividades extra curriculares
T	Enfermería	Uso de la enfermería de la escuela
U	EduSuperior	Desea tomar la educación superior
V	Internet	Tiene acceso a internet desde la casa
W	Rsentimental	Mantiene una relación sentimental
X	Rfamiliar	Calidad de la relación familiar
Y	Tiempolibre	Tiene mucho tiempo libre luego de la escuela
Z	Oscio	Sale con amigos
AA	dalcohol	Consume alcohol en los días laborables
AB	fsalcohol	Consume alcohol en los fines de semana
AC	Salud	Estado de salud actual
AD	Ausencia	Número de inasistencias a la escuela

Tabla 5.24 Variables y tipo de los atributos

Variable	Tipo	Propuesta discretización
A	Binario	GP=Gabriel Pereira=0, MS=Mousinho da Silveira=1
B	Binario	F=femenino=0 o M=masculino=1
C	Numérico	15 - 22
D	Binario	U=Urbano=0, R=rural=1
E	Binario	LE3=X<=3, GT3=X>3; LE3=0, GT3=1
F	Binario	T=viven juntos=1 o A=aparte=0
G	Numérico	0=nada, 1=primaria (4to grado), 2= de 5to a 9no, 3=secundaria o 4=superior
H	Numérico	0=nada, 1=primaria (4to grado), 2= de 5to a 9no, 3=secundaria o 4=superior
I	Nominal	profesor=0, relacionado a la salud=1, servicios civiles=2, en casa=3, otro=4
J	Nominal	profesor=0, relacionado a la salud=1, servicios civiles=2, en casa=3, otro=4
K	Nominal	cerca de casa=0, reputación de la escuela=1, preferencia de curso=2, otro=3
L	Nominal	madre=0, padre=1, otro=2
M	Numérico	1=<15 min, 2=de 15 a 30 min, 3= de 30 min a 1h, 4=>1h
N	Numérico	1=<2h, 2=de 2 a 5h, 3=de 5 a 10h, 4=>10h
O	Numérico	n si 1<=n<3, else 4
P	Binario	si=1 o no=0
Q	Binario	si=1 o no=0
R	Binario	si=1 o no=0
S	Binario	si=1 o no=0
T	Binario	si=1 o no=0
U	Binario	si=1 o no=0
V	Binario	si=1 o no=0
W	Binario	si=1 o no=0
X	Numérico	desde 1=muy mal a 5=excelente
Y	Numérico	desde 1=muy poco a 5=mucho
Z	Numérico	desde 1=muy poco a 5=mucho
AA	Numérico	desde 1=muy poco a 5=mucho
AB	Numérico	desde 1=muy poco a 5=mucho
AC	Numérico	desde 1=muy mal a 5=muy bien
AD	Numérico	si es 0=0; 1-15=1;16-30=2;31-45=3;46-60=4;61-75=5

5.3.1. Clasificación usando Weka con los datos de la UCI curso matemáticas

Podemos visualizar en las figuras 5.6 y 5.7, los resultados obtenidos por cada uno de los atributos en referencia a la clase G3 grado final. En base a esto se han construido las tablas 5.25, 5.26, 5.27 y 5.28, en el que podemos encontrar el análisis descriptivo de las variables.

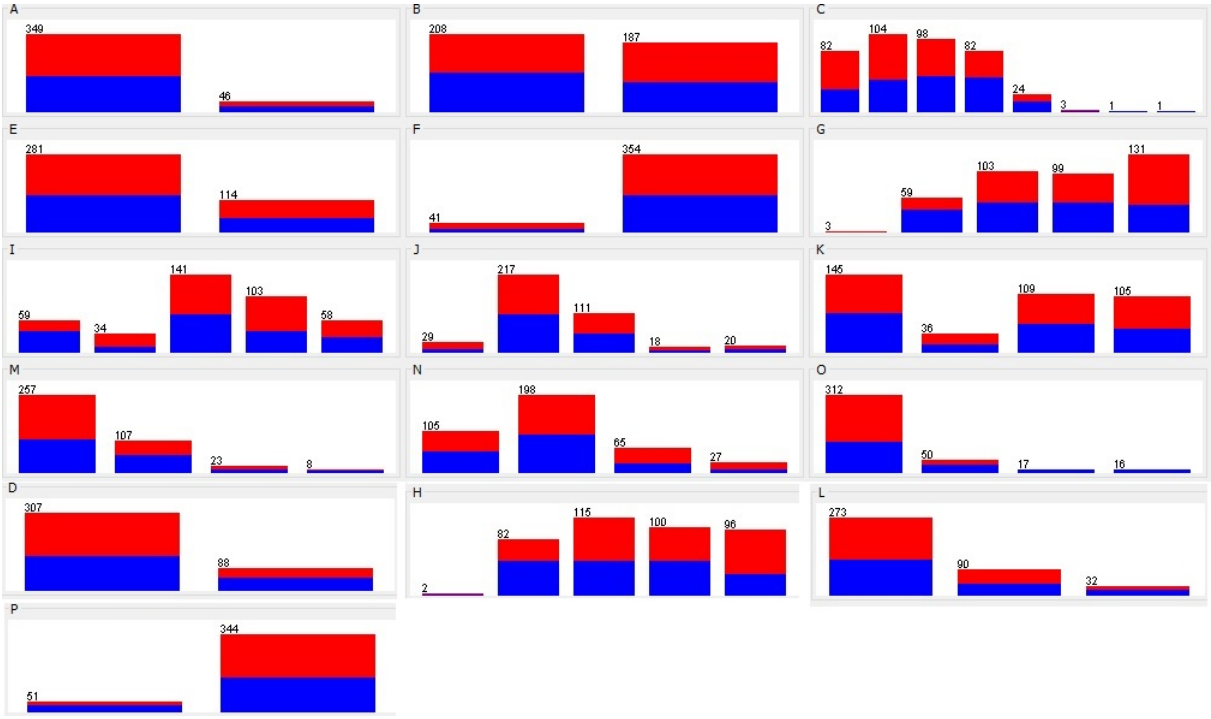

Figura 5.6 Resultados obtenidos por cada uno de los atributos en referencia a la clase

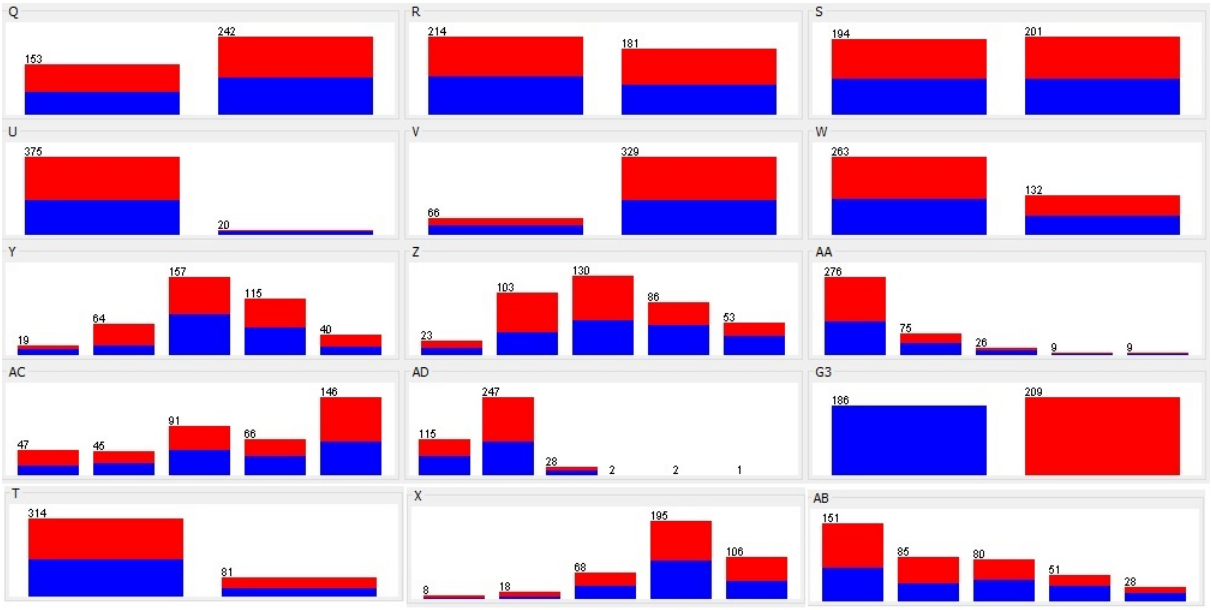

Figura 5.7 Resultados obtenidos por cada uno de los atributos en referencia a la clase

Tabla 5.25 Variables y análisis descriptivo

Variable	Descripción	Cantidad	Porcentaje
A	GP	349	88.35
	MS	46	11.65
B	F	208	52.66
	M	187	47.34

Tabla 5.26 Variables y análisis descriptivo

Variable	Descripción	Cantidad	Porcentaje
C	15	82	20.76
	16	104	26.33
	17	98	24.81
	18	82	20.76
	19	24	6.08
	20	3	0.76
	21	1	0.25
	22	1	0.25
D	U	307	77.72
	R	88	22.28
E	GT3	281	71.14
	LE3	114	28.86
F	A	41	10.38
	T	354	89.62
G	0	3	0.76
	1	59	14.94
	2	103	26.08
	3	99	25.06
	4	131	33.16
H	0	2	0.51
	1	82	20.76
	2	115	29.11
	3	100	25.32
	4	96	24.30
I	casa	59	14.94
	salud	34	8.61
	otro	141	35.70
	servicios	103	26.08
	profesor	58	14.69
J	casa	29	7.34
	salud	217	54.94
	otro	111	28.10
	servicios	18	4.56
	profesor	20	5.06

Tabla 5.27 Variables y análisis descriptivo

Variable	Descripción	Cantidad	Porcentaje
K	curso	145	36.71
	otro	36	9.11
	casa	109	27.60
	reputación	105	26.59
L	madre	273	69.11
	padre	90	22.78
	otro	32	8.10
M	1	257	65.06
	2	107	27.09
	3	23	5.82
	4	8	2.03
N	1	105	26.58
	2	198	50.13
	3	65	16.46
	4	27	6.84
O	0	312	78.99
	1	50	12.66
	2	17	4.30
	3	16	4.05
P	si	51	12.91
	no	344	87.09
Q	no	153	38.73
	si	242	61.27
R	no	214	54.18
	si	181	45.82
S	no	194	49.11
	si	201	50.89
T	si	314	79.49
	no	81	20.51
U	si	375	94.94
	no	20	5.06
V	no	66	16.71
	si	329	83.29
W	no	263	66.59
	si	132	33.41

Tabla 5.28 Variables y análisis descriptivo

Variable	Descripción	Cantidad	Porcentaje
X	1	8	2.03
	2	18	4.56
	3	68	17.22
	4	195	49.37
	5	106	26.84
Y	1	19	4.81
	2	64	16.20
	3	157	39.75
	4	115	29.11
	5	40	10.13
Z	1	23	5.82
	2	103	26.08
	3	130	32.91
	4	86	21.77
	5	53	13.42
AA	1	276	69.87
	2	75	18.99
	3	26	6.58
	4	9	2.28
	5	9	2.28
AB	1	151	38.23
	2	85	21.52
	3	80	20.25
	4	51	12.91
	5	28	7.09
AC	1	47	11.90
	2	45	11.39
	3	91	23.04
	4	66	16.71
	5	146	36.96
AD	0	115	29.11
	1	247	62.53
	2	28	7.09
	3	2	0.51
	4	2	0.51
	5	1	0.25
G3	0	186	47.09
	1	209	52.91

5.3.1.1. Usando clasificadores bayesianos para datos UCI curso de matemáticas

Ante la posibilidad de que un estudiante logre alcanzar la meta de terminar sus estudios o no, se los debe someter a pruebas, cuyo resultado final podrá ser positivo o negativo. Si el resultado es positivo se determina que el estudiante tendrá una posibilidad alta de aprobar. La sensibilidad y la especificidad son dos valores de probabilidad que cuantifican la fiabilidad diagnóstica de una prueba.

Se obtuvieron resultados usando como clasificadores Naive Bayes y BayesNet con diferentes alternativas como K2 con 1 y 5 padres, TAN, Hill Climber con 1 y 5 padres.

Tabla 5.29 Resultados obtenidos con clasificadores, datos UCI matemáticas

Clasificador	**Clasificados correctamente**	**Tasa TN**	**Tasa TP**
NaiveBayes	65.8228	0.575	0.732
BayesNet con K2-1 padre	66.0759	0.581	0.732
BayesNet con K2-5 padres	66.0759	0.597	0.718
BayesNet con TAN	67.0886	0.591	0.742
BayesNet con Hill Climber-1	61.7722	0.468	0.751
BayesNet con Hill Climber-5	58.9873	0.409	0.751

Como se puede observar en la tabla 5.29 se ha trabajado con 395 casos y que el clasificador que más correctamente ha clasificado es BayesNet con TAN con un 67.0886 por ciento, con una tasa de verdaderos negativos de 0.591, mientras que permite acertar de quiénes aprobarán el curso de matemáticas en un 74.2 por ciento.

Se puede determinar según la figura 5.8 que trabajando con BayesNet con Hill Climber la variable clase $(G3)$ está relacionada de manera directa sobre si el estudiante desea ir a la universidad (U) o no, así también influye directamente sobre el número de fracasos en clases anteriores (O). Por otro lado el grado de educación de los padres influye sobre el tipo de trabajo que tienen ellos; pero, no tienen relación directa con el fracaso del estudiante. Las variables sociales también se relacionan fuertemente entre ellas, como es el caso de que sale con amigos (Z) es porque tiene tiempo libre luego de la escuela (Y) o porque consume alcohol el fin de semana (AB).

5.3.1.2. Usando clasificadores de árboles para datos UCI matemáticas

Se obtuvo resultados usando como clasificadores de árboles J48 y Random Forest.

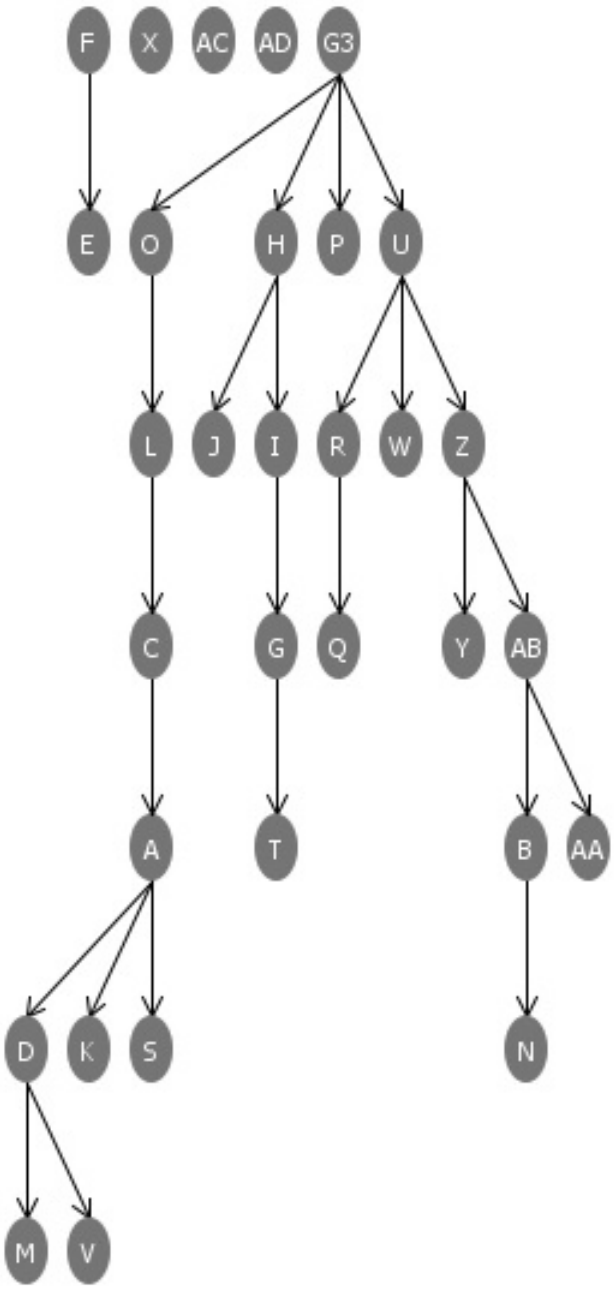

Figura 5.8 Red obtenida con clasificador BayesNet con Hill Climber con un solo padre

Tabla 5.30 Resultados obtenidos con clasificadores de árboles para UCI matemáticas

Clasificador	**Clasificados correctamente**	**Tasa TN**	**Tasa TP**
J48	59.4937	0.554	0.632
Random Forest	59.7468	0.527	0.66

Como se puede observar en la tabla 5.30 trabajando con un clasificador de árbol J48 los casos clasificados correctamente equivalen al 59.4937 por ciento de los 395 casos. En el caso de Random Forest los casos clasificados correctamente equivalen al 59.7468 por ciento, lo que

permite determinar que hay una certeza del 66 por ciento para determinar los aprobados del curso. Adicionalmente se debe indicar que es un bosque aleatorio de 100 árboles de los cuales cada uno está construido con 5 características aleatorias.

5.3.1.3. Usando reglas de clasificación para datos UCI matemáticas

Se obtuvieron resultados usando como reglas de clasificación ZeroR y tablas de decisiones.

Tabla 5.31 Resultados obtenidos con diferentes reglas de clasificación para UCI matemáticas

Clasificador	Clasificados correctamente	Tasa TN	Tasa TP
ZeroR	52.9114	0	1
Tabla de decisiones	63.7975	0.457	0.779

Como se puede observar en la tabla 5.31, trabajando con reglas de clasificación Zero los casos clasificados correctamente equivalen al 52.9114 por ciento de los 395 casos. Pero que al utilizar tablas de decisiones el porcentaje mejora a un 63.7975 por ciento. Se determina entonces que se tiene el 77.90 por ciento de certeza en determinar los alumnos que van aprobar el curso.

Al igual que en la sección 5.1.1 se realizará una comparación de los resultados obtenidos con los diferentes algoritmos con la base de datos de la UCI que toman el curso de matemáticas, tal como se indica en la tabla 5.32

Tabla 5.32 Resultados con base de datos de estudiantes de la UCI matemáticas

Datos	SBND1	SBND2	SBND3	SBND4	BAN BDe	BAN BIC
UCIMat	60.558	63.103	61.532	57.237	63.25	62.994
Datos	BAN K2	RPDag BDe	RPDag BIC	RPDag K2	TAN	NaiveBayes
UCIMat	63.256	63.859	63.859	59.25	61.551	66.564

5.3.2. Clasificación usando Weka con los datos de la UCI curso portugués

Podemos visualizar en las figuras 5.9 y 5.10, los resultados obtenidos por cada uno de los atributos en referencia a la clase G3 grado final. En base a esto se han construido las tablas 5.33, 5.34, 5.35 y 5.36, en el que podemos encontrar el análisis descriptivo de las variables. Se debe indicar que se trabaja para este caso del curso de portugués con un total de 649 estudiantes con la misma cantidad de atributos del caso del curso de matemáticas.

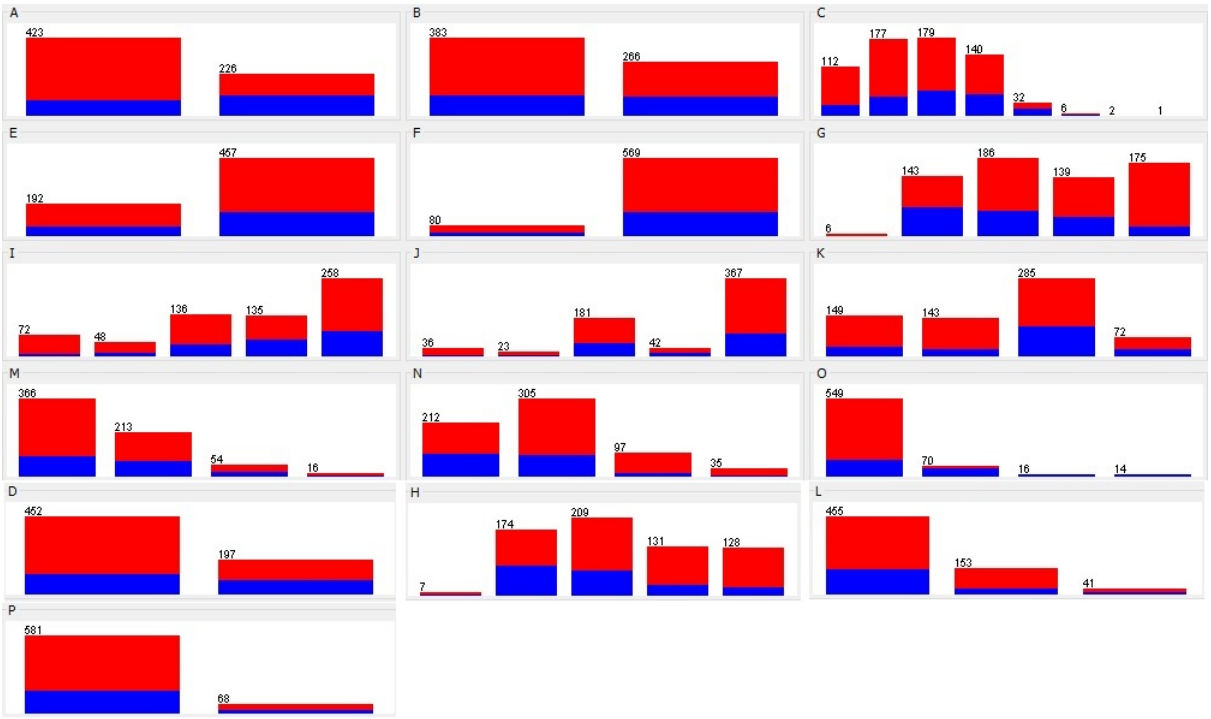

Figura 5.9 Resultados obtenidos por cada uno de los atributos en referencia a la clase

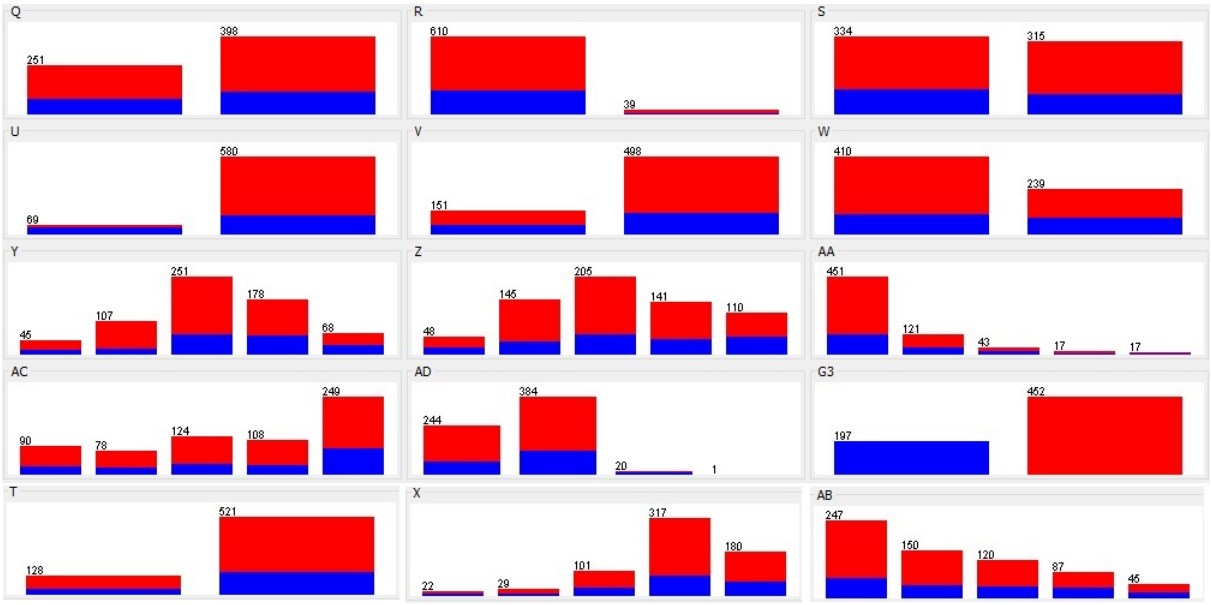

Figura 5.10 Resultados obtenidos por cada uno de los atributos en referencia a la clase

Tabla 5.33 Variables y análisis descriptivo

Variable	Descripción	Cantidad	Porcentaje
A	GP	423	65.18
	MS	226	34.82
B	F	383	59.01
	M	266	40.99

Tabla 5.34 Variables y análisis descriptivo

Variable	Descripción	Cantidad	Porcentaje
C	15	112	17.26
	16	177	27.27
	17	179	27.58
	18	140	21.57
	19	32	4.93
	20	6	0.92
	21	2	0.31
	22	1	0.15
D	U	452	69.65
	R	197	30.35
E	GT3	192	29.59
	LE3	457	70.41
F	A	80	12.33
	T	569	87.67
G	0	6	0.92
	1	143	22.03
	2	186	28.66
	3	139	21.42
	4	175	26.96
H	0	7	1.08
	1	174	26.81
	2	209	32.20
	3	131	20.18
	4	128	19.72
I	casa	72	11.09
	salud	48	7.40
	otro	136	20.96
	servicios	135	20.80
	profesor	258	39.75
J	casa	36	5.55
	salud	23	3.54
	otro	181	27.89
	servicios	42	6.47
	profesor	367	56.55

Tabla 5.35 Variables y análisis descriptivo

Variable	Descripción	Cantidad	Porcentaje
K	curso	149	22.96
	otro	143	22.03
	casa	285	43.1
	reputación	72	11.09
L	madre	455	70.11
	padre	153	23.57
	otro	41	6.32
M	1	366	56.39
	2	213	32.82
	3	54	8.32
	4	16	2.47
N	1	212	32,67
	2	305	47.00
	3	97	14.95
	4	35	5.39
O	0	549	84.59
	1	70	10.79
	2	16	2.47
	3	14	2.16
P	sí	581	89.52
	no	68	10.48
Q	no	251	38.67
	sí	398	61.33
R	no	610	93.99
	sí	39	6.01
S	no	334	51.46
	sí	315	48.54
T	sí	128	19.72
	no	521	80.28
U	sí	69	10.63
	no	580	89.37
V	no	151	23.27
	sí	498	76.73
W	no	410	63.17
	si	239	36.83

Tabla 5.36 Variables y análisis descriptivo

Variable	Descripción	Cantidad	Porcentaje
X	1	22	3.39
	2	29	4.47
	3	101	15.56
	4	317	48.84
	5	180	27.73
Y	1	45	6.93
	2	107	16.49
	3	251	38.67
	4	178	27.43
	5	68	10.48
Z	1	48	7.40
	2	145	22.34
	3	205	31.59
	4	141	21.73
	5	110	16.95
AA	1	451	69.49
	2	121	18.64
	3	43	6.63
	4	17	2.62
	5	17	2.62
AB	1	247	38.06
	2	150	23.11
	3	120	18.49
	4	87	13.41
	5	45	6.93
AC	1	90	13.87
	2	78	12.02
	3	124	19.11
	4	108	16.64
	5	249	38.37
AD	0	244	37.60
	1	384	59.17
	2	20	3.08
	3	1	0.15
	4	0	0.00
	5	0	0.00
G3	0	197	30.35
	1	452	69.65

5.3.2.1. Usando clasificadores bayesianos para datos UCI portugués

Ante la posibilidad de que un estudiante logre alcanzar la meta de terminar sus estudios o no, se los debe someter a pruebas, cuyo resultado final podrá ser positivo o negativo. Si el resultado es positivo se determina que el estudiante tendrá una posibilidad alta de aprobar.

Se obtuvo resultados usando como clasificadores Naive Bayes y BayesNet con diferentes alternativas como K2 con 1 y 5 padres, TAN, Hill Climber con 1 y 5 padres.

Tabla 5.37 Resultados obtenidos con clasificadores datos UCI portugués

Clasificador	Clasificados correctamente	Tasa TN	Tasa TP
NaiveBayes	78.1202	0.635	0.845
BayesNet con K2-1 padre	78.1202	0.64	0.843
BayesNet con K2-5 padres	77.6579	0.599	0.854
BayesNet con TAN	77.3498	0.574	0.861
BayesNet con Hill Climber-1	75.6549	0.614	0.819
BayesNet con Hill Climber-5	79.1988	0.569	0.889

Como se puede observar en la tabla 5.37 se ha trabajado con 649 casos y que el clasificador que más correctamente ha clasificado es BayesNet con Hill Climber con un máximo de 5 padres con un 79.1988 por ciento, con una tasa de verdaderos negativos de 0.569, lo que nos permite acertar quiénes aprobarán el curso de portugués en un 88.90 por ciento. Adicionalmente nos indica los valores de verdaderos positivos que es la sensibilidad y falsos positivos o verdaderos negativos que representa la especificidad.

En este caso las variables de número de inasistencias (AD) y si consume alcohol el estudiante en días laborables (AA) están relacionadas directamente con la variable clase $(G3)$. Las variables de índole social se siguen relacionando tal como en el caso del curso de matemáticas según la figura

5.3.2.2. Usando clasificadores de árboles para datos UCI portugués

Se obtuvieron resultados usando como clasificadores de árboles J48 y Random Forest.

Tabla 5.38 Resultados obtenidos con clasificadores de árboles para UCI portugués

Clasificador	Clasificados correctamente	Tasa TN	Tasa TP
J48	78.2743	0.553	0.883
Random Forest	77.0416	0.365	0.947

Como se puede observar en la tabla 5.38 trabajando con un clasificador de árbol J48 los casos clasificados correctamente equivalen al 78.2743 por ciento de los 649 casos. En el caso de

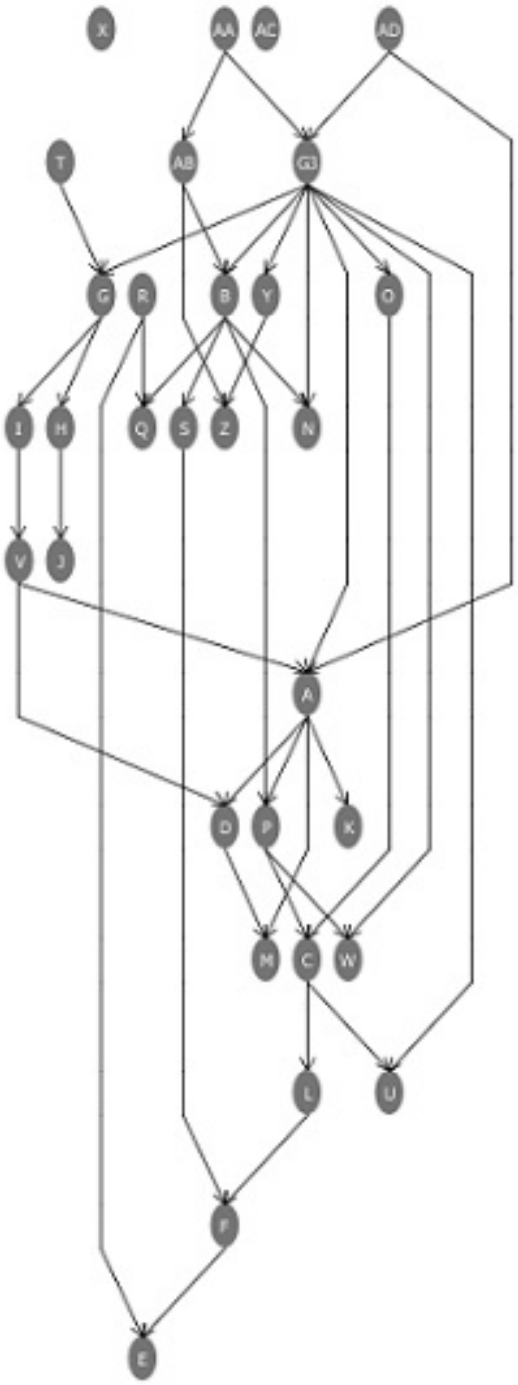

Figura 5.11 Red obtenida con clasificador BayesNet con Hill Climber con un máximo de 5 padres

Random Forest se trabajó con un bosque aleatorio de 100 árboles cada uno de ellos construidas con 5 características aleatorias y los casos clasificados correctamente equivalen al 77.0416 por ciento. En este caso clasifica mejor J48 con un 88.30 por ciento de efectividad para determinar los estudiantes que aprobarán el curso de portugués.

5.3.2.3. Usando reglas de clasificación para datos UCI portugués

Se obtuvo resultados usando como reglas de clasificación ZeroR y tablas de decisiones.

Tabla 5.39 Resultados obtenidos con diferentes reglas de clasificación para UCI portugués

Clasificador	Clasificados correctamente	Tasa TN	Tasa TP
ZeroR	69.6456	0	1
Tabla de decisiones	80.1233	0.513	0.927

Como se puede observar en la tabla 5.39 trabajando con reglas de clasificación ZeroR los casos clasificados correctamente equivalen al 69.6456 por ciento de los 649 casos. Pero al utilizar tablas de decisiones el porcentaje mejora a un 80.1233 por ciento llegando a tener un 92.70 por ciento de efectividad al determinar los estudiantes que aprobarán el curso.

Se realiza un experimento en el que se han cargado todos los algoritmos que se calcularon por separado con una significancia de 0.05, usando una validación cruzada de 10 tandas para comparar el campo de bien clasificados y que nos muestre la desviación estándar, obteniendo los datos de la tabla 5.40.

Tabla 5.40 Experimento con todos los algoritmos datos UCI portugués

Descripción	Valor
Analizando	Porcentaje correcto
Conjunto de datos	1
Conjunto de resultados	10

Datos	K2-5p	K2-1p	TAN	Hill-1p	Hill-5p	NBayes	Zero	Tabla	J48	RF
UCI port	76.86	77.80	76.61	76.61	78.79	77.89	69.65	79.94	77.92	77.57
desviación std	4.69	5.08	5.07	4.95	4.36	5.12	0.65	4.14	4.36	4.44

Al realizar el experimento ingresando todos los 10 algoritmos en el analizador de la herramienta Weka podemos determinar que el uso de las tablas de decisiones permite tener mayor cantidad de datos bien clasificados (79.94 por ciento) con una desviación estándar de 4.14.

Al igual que en la sección 5.1.1 se realizará una comparación de los resultados obtenidos con los diferentes algoritmos con la base de datos de la UCI que toman el curso de portugués, tal como se indica en la tabla 5.41

Tabla 5.41 Resultados con base de datos de estudiantes de la UCI portugués

Datos	SBND1	SBND2	SBND3	SBND4	BAN BDe	BAN BIC
UCIPort	79.661	79.815	76.269	79.351	78.572	77.493
Datos	BAN K2	RPDag BDe	RPDag BIC	RPDag K2	TAN	NaiveBayes
UCIPort	73.793	79.343	77.959	73.038	78.120	77.651

5.4. Conclusiones parciales

En este capítulo hemos estudiado un nuevo clasificador bayesiano simple, el mismo que fue aplicado al análisis de datos en problemas de educación. Este clasificador es rápido de aprender y muy competitivo en relación a los otros clasificadores del estado del arte. Se realizaron varios experimentos tanto con base de datos socio-económicos de estudiantes legalmente matriculados en la Facultad de Ciencias de la Ingeniería de la Universidad Técnica Estatal de Quevedo en el periodo 2012-2013, como también con base de datos de la UCI de dos escuelas portuguesas.

Se pudo determinar en esta investigación que la métrica BIC no da buenos resultados con pruebas no paramétricas, mientras que Akaike entrega un excelente resultado en referencia a la media, pero mal en el orden en términos de clasificación.

Una vez analizados los datos de clasificación obtenidos con la herramienta WEKA se puede determinar que hay variables que tienen mayor influencia que otras con respecto a la clase.

Se puede indicar que el problema de deserción estudiantil que se analiza es complejo y difícil y que ha sido necesario utilizar métodos que usan una combinación de factores (clasificadores bayesianos) para poder obtener algunas mejoras sobre el clasificador trivial que determina que ningún estudiante deserta.

Aunque la tasa de éxito no es muy alta se puede determinar que usando un clasificador bayesiano (bayesNet con K2 y un máximo de 5 padres) se puede detectar el 32.20 por ciento de los estudiantes que van a desertar y poder aplicar metodologías que ayuden a trabajar con este individuo de manera tal que se evite ese paso.

El coste medido como el porcentaje de alumnos que se consideran potenciales desertores entre los no desertores es muy bajo y equivale al 1.5 por ciento lo que no permite determinar con exactitud los desertores.

Como trabajo futuro, pensamos que se debe incluir los costes de las clasificaciones incorrectas en el problema, ya que no es lo mismo un falso positivo que un falso negativo. Si consideramos que el coste de un falso negativo es mejor que el de un falso positivo se podría detectar más alumnos que abandonarían, aunque se aumentaría el número de estudiantes que se consideran en peligro de desertar.

Parte III

Conclusiones

6 Conclusiones y trabajos futuros

6.1. Conclusiones

La permanencia o deserción de un estudiante en la universidad y la culminación con éxito de sus estudios están influenciadas por diferentes factores: individuales, académicos, socio-económicos e institucionales. En este sentido el fenómeno de la deserción seguirá aunque haya cambios en las Instituciones de Educación Superior. El investigar este problema permitió establecer soluciones que controlen de manera parcial los índices de deserción y se logre retener a los estudiantes.

Muchos problemas cotidianos pueden ser resueltos a través de la aplicación de las redes bayesianas que es un MGP muy popular basada en la estadística bayesiana y ha permitido modelizar en dominios complejos con incertidumbre intrínseca.

Se realizó una revisión de los modelos gráficos probabilísticos y los métodos de aprendizaje donde se han podido distinguir tres enfoques principales:

1. La estimación de la probabilidad conjunta: es encontrar una red genérica que mejor se ajuste a los datos.
2. La clasificación supervisada: cuando existe una variable distinguida llamada clase cuyos valores queremos predecir en función de los otros atributos.
3. La clasificación no supervisada: suponiendo la existencia de una variable oculta no observada, cuyos valores representan los distintos grupos que se supone que existen en la población.

También se realizó una breve descripción de otros modelos de clasificación supervisada como árboles de clasificación y reglas de asociación que se usaron en esta memoria.

Como contribuciones en el capítulo 3 hemos propuesto un nuevo procedimiento para inducir modelos gráficos probabilísticos con variables ocultas de los datos: 'un agrupamiento jerárquico'.

En lugar de considerar sólo una variable oculta como en AutoClass, se ha considerado un árbol de variables ocultas.

Se ha demostrado que este modelo tiene un buen rendimiento para estimar un conjunto de distribución de probabilidad de un conjunto de datos observados, tiene algunas ventajas adicionales en comparación con los procedimientos generales de aprendizaje de redes bayesianas. Nuestro enfoque produce árboles (o un bosque de árboles) que son fáciles de interpretar y rápidos para el cálculo.

El rendimiento de este modelo depende del caso particular donde está siendo aplicado. En general, es apropiado en problemas que hay variables relacionadas entre ellas y que esto es debido al hecho de que dependen de factores ocultos no observados. Este es el caso del conjunto de datos de los estudiantes de la Universidad Técnica Estatal de Quevedo (Ecuador) donde se tiene algunas variables que manifiestan dependencia del nivel económico de la familia de los estudiantes, otras variables están relacionadas con las características académicas de los estudiantes, etc.

En resumen, nuestro procedimiento produce una muy buena estimación de la distribución de probabilidades conjunta y al mismo tiempo genera un entorno natural y fácil de interpretar. Esto se debe a que se propone un agrupamiento de la estructura gráfica de las variables observables en una jerarquía.

Adicionalmente, se ha logrado adaptar la metaheurística optimización de mallas variables (VMO) en el capítulo 4 para el aprendizaje estructural de una red bayesiana para clasificación supervisada. Utilizamos como métrica de score la precisión de la clasificación. Las operaciones de VMO's también fueron adaptadas a este problema. Luego se llevaron a cabo experimentos utilizando 14 conjuntos de datos del repositorio de la UCI. Los resultados de BayesVMO fueron comparados con 8 clasificadores del estado del arte y los análisis estadísticos presentan mejores resultados.

En el capítulo 5, Se introduce un nuevo clasificador bayesiano simple (SBND) que aprende forma rápida una frontera de Markov de la variable clase y una estructura de red que relaciona las variables de la clase y su frontera de Markov. Se analizaron los datos de clasificación obtenidos con varios algoritmos y se los comparó con el propuesto SBND pudiendo determinar que este algoritmo con métrica K2 resultó ser el de mejor clasificación.

Hemos comprobado que el comportamiento depende de la métrica usada en la construcción del modelo. En esta investigación la métrica BIC no da buenos resultados con pruebas no paramétricas, mientras que Akaike entrega un excelente resultado en referencia a la media, pero mal en el orden en términos de clasificación.

Posteriormente se analizaron los datos de clasificación obtenidos con los diferentes algoritmos de la herramienta WEKA como naive Bayes, BayesNet con K2, BayesNet con HillClimber, TAN. Se puede determinar que las variables curso y carrera tienen mayor influencia que otras con respecto a la clase desertar.

Se puede indicar que el problema de deserción estudiantil que se analiza es complejo y difícil

y que ha sido necesario utilizar métodos que usan una combinación de factores (clasificadores bayesianos) para poder obtener algunas mejoras sobre el clasificador trivial que determina que ningún estudiante deserta.

Aunque la tasa de éxito no es muy alta se puede determinar que usando un clasificador bayesiano (bayesNet con K2 y un máximo de 5 padres) se puede detectar el 32.20 por ciento de los estudiantes que van a desertar y poder aplicar metodologías que ayuden a trabajar con este individuo de manera tal que se evite ese paso.

El coste medido como el porcentaje de alumnos que se consideran potenciales desertores entre los no desertores es muy bajo y equivale al 1.5 por ciento lo que hace un poco complicado determinar con exactitud los desertores.

Se ha podido determinar también que Naive Bayes es un algoritmo muy eficiente; pero que los clasificadores TAN y KDB generan redes con mejor precisión que NB, aunque estos son más complejos de desarrollar y requieren mayor tiempo de procesamiento para generar las redes.

De manera general en este trabajo se ha podido evidenciar que un factor de impacto para la deserción estudiantil es el curso que están tomando y la carrera, ya que de manera directa influyen en el desempeño académico del estudiante. Este es el caso de los estudiantes legalmente matriculados en las carreras de la Facultad de Ciencias de la Ingeniería de la Universidad Técnica Estatal de Quevedo durante el periodo lectivo 2012-2013. Una vez analizados los resultados obtenidos de manera descriptiva se ha podido determinar que el 30 por ciento de los estudiantes reprueban; pero, toman una segunda matrícula en la materia que reprueban. El 11 por ciento de los estudiantes que pierden una materia llegan a desertar. Estos datos son el resultado de la convergencia de las variables analizadas.

Se sugiere revisar los diferentes procesos de selección y admisión de los aspirantes al ingresar a la Universidad Técnica Estatal de Quevedo para poder detectar de entrada los posibles desertores. Muchas veces la elección inadecuada de la carrera por falta de orientación vocacional no permite un mejor desempeño académico por lo que se deberá hacer el respectivo seguimiento; de igual forma el factor económico influye de manera directa por lo que sería adecuado poder implementar políticas de becas y apoyo socio-económico.

6.2. Trabajos futuros

En el futuro, se ha planificado extender el estudio de deserción incluyendo más variables académicas que parecen tener más influencia en las variables de interés, de manera tal que permita tener una idea más clara de los motivos de la deserción estudiantil y la aprobación de los cursos, como es el caso de trabajar con variables relacionadas a la planta docente y a la infraestructura de la Institución de Educación Superior.

También se debe considerar el problema de deserción como un problema de clasificación no supervisada, adaptando nuestro procedimiento jerárquico para clasificación y comparando el

desarrollo de éste con otros modelos.

Se propone realizar un análisis del perfil de los estudiantes que llegan a desertar en las carreras de la Facultad de Ciencias de la Ingeniería teniendo cuidado con la protección de los datos tal como lo demuestra Carmen Lacave en el estudio realizado en la Universidad de Castilla-La Mancha.

Como trabajo futuro se propone también aplicar los métodos de esta investigación a los datos socio-económicos de los estudiantes que ingresan al pre universitario de la Universidad Técnica Estatal de Quevedo y luego comparar con los obtenidos en esta memoria.

6.3. Publicaciones derivadas de la investigación

Durante el desarrollo de la presente investigación se obtuvieron los resultados que se muestran a continuación.

Tesis de pregrado:

- Aprendizaje estructural de redes bayesianas utilizando la meta heurística optimización basada en mallas variables (VMO) - Autor: Luis Enrique Moreira Zamora - 2014.

Tesis de posgrado:

- Planificación de celdas en redes móviles GSM - Autor: Osmar Viera Carcache - 2015.

Artículos en revistas:

- A Hierarchical Clustering Method: Applications to Educational Data - Autores: Byron Oviedo, Serafín Moral, Amilkar Puris - Intelligent Data Analysis - An International Journal - 20 (2016) - páginas: 933-951 - doi 10.3233/IDA-160839

Trabajos enviados a congresos y/o eventos:

- Twelfth LACCEI Latin American and Caribbean Conference for Engineering and Technology (LACCEI 2014) Excellence in Engineering To Enhance a Countrys Productivity July 22 - 24, 2014 Guayaquil, Ecuador. Optimización basada en Mallas Variables: Caso de estudio Viajante de Comercio - Autores: Byron Oviedo, Amilkar Puris, Eduardo Díaz, Jorge Guanín
- Análisis de datos educativos utilizando redes bayesianas - Autores: Byron Oviedo, Amilkar Puris, Annabelle Villacís, Ana Moreno, Diana Delgado - LACCEI 2015, Latin American and Caribbean Consortium of Engineering Institutions

- Learning Bayesian Network by a Mesh of points: Autores: Byron Oviedo, Serafín Moral, Amilkar Puris, Luis Moreira - IEEE explorer - 2016

Bibliografía

[Abellán y Moral2003] Abellán, Joaquín, y Serafín Moral. 2003. Building classification trees using the total uncertainty criterion. International Journal of Intelligent Systems 18 (12): 1215–1225.

[Acid y De Campos2003] Acid, Silvia, y Luis De Campos. 2003. Searching for Bayesian network structures in the space of restricted acyclic partially directed graphs. Journal of Artificial Intelligence Research, pp. 445–490.

[Acid, De Campos, y Castellano2005] Acid, Silvia, Luis De Campos, y Javier Castellano. 2005. Learning Bayesian network classifiers: Searching in a space of partially directed acyclic graphs. Machine Learning 59 (3): 213–235.

[Acosta et al.2014] Acosta, Héctor, Rechy Fernando, Mezura Efraín, Cruz-Ramírez Nicandro, y Hernández Rodolfo. 2014. Application of time series discretization using evolutionary programming for classification of precancerous cervical lesions. Journal of Biomedical Informatics 49:73 – 83.

[Agrawal, Imieliñski, y Swami1993] Agrawal, Rakesh, Tomasz Imieliñski, y Arun Swami. 1993. Mining association rules between sets of items in large databases. Acm sigmod record, Volume 22. ACM, 207–216.

[Akaike1974] Akaike, Hirotugu. 1974. A new look at the statistical model identification. Automatic Control, IEEE Transactions on 19 (6): 716–723.

[Anderson et al.1986] Anderson, John, Michalski Ryszard, Carbonell Jaime, y Mitchell Tom. 1986. Machine learning: An artificial intelligence approach. Volume 2. Morgan Kaufmann.

[Aroztegui, Arraga, y Nesmachnow2003] Aroztegui, Miguel, Santiago Arraga, y Sergio Nesmachnow. 2003. Resolución del Problema de Steiner Generalizado utilizando un algoritmo genético paralelo. Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados. 387–394.

[Bhargava et al.2013] Bhargava, Neeraj, Sharma Girja, Bhargava Ritu, y Mathuria Manish. 2013. Decision tree analysis on j48 algorithm for data mining. Proceedings of International Journal of Advanced Research in Computer Science and Software Engineering 3, no. 6.

[Breiman2001] Breiman, Leo. 2001. Random forests. Machine learning 45 (1): 5–32.

[Carra2016] Carra, Pablo. 2016. Predictor en tiempo real de patrones armónicos.

[Castillo, Gutiérrez, y Hadi1997] Castillo, Enrique, José Gutiérrez, y Ali Hadi. 1997. Expert systems and probabilistic network models Springer. New York.

[Cheeseman et al.1993] Cheeseman, Peter, James Kelly, Self Mattew, Stutz John, Taylor Will, y Freeman Don. 1993. Autoclass: a bayesian classification system. Readings in Knowledge Acquisition and Learning. Morgan Kaufmann Publishers Inc., 431–441.

[Cheeseman et al.1996] Cheeseman, Peter, Self Mattew, Jim Kelly, y Stutz John. 1996. Bayesian Classification.

[Cheng y Russell1999] Cheng, Jie, y Greiner Russell. 1999. Comparing Bayesian network classifiers. Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., 101–108.

[Chickering1996] Chickering, David. 1996. Learning Bayesian Networks is NP-Complete. Learning from Data: Artificial Intelligence and Statistics V. Springer-Verlag, 121–130.

[Chow y Liu1968] Chow, C, y Cong Liu. 1968. Approximating discrete probability distributions with dependence trees. IEEE transactions on Information Theory 14 (3): 462–467.

[Chávez2008] Chávez, María. 2008. Modelos de redes bayesianas en el estudio de secuencias genómicas y otros problemas biomédicos. Tesis Doctoral, Universidad Central"Marta Abreu"de las Villas. Villa Clara, Cuba.

[Cooper y Herskovits1992] Cooper, Gregory, y Edward Herskovits. 1992. A Bayesian method for the induction of probabilistic networks from data. Machine learning 9 (4): 309–347.

[Corso2009] Corso, Cynthia. 2009. Aplicación de algoritmos de clasificación supervisada usando Weka. Córdoba: Universidad Tecnológica Nacional, Facultad Regional Córdoba.

[Cortez y Goncálves2008] Cortez, Paulo, y Silvia Goncálves. 2008. Using data mining to predict secondary school student performance, Univerity of Minho.

[De Campos y Castellano2007] De Campos, Luis, y Javier Castellano. 2007. Bayesian network learning algorithms using structural restrictions. International Journal of Approximate Reasoning 45 (2): 233–254.

[De Campos et al.2002] De Campos, Luis, Fernandez-Luna Juan, Gámez José, y Puerta José. 2002. Ant colony optimization for learning Bayesian networks. International Journal of Approximate Reasoning 31 (3): 291–311.

[De Campos y Puerta2001] De Campos, Luis, y Juan Puerta. 2001. Stochastic local and distributed search algorithms for learning belief networks. Proceedings of the III International Symposium on Adaptive Systems: Evolutionary Computation and Probabilistic Graphical Model. 109–115.

[Dekker, Pechenizkiy, y Vleeshouwers2009] Dekker, Gerben, Mykola Pechenizkiy, y Jan Vleeshouwers. 2009. Predicting Students Drop Out: A Case Study. International Working Group on Educational Data Mining.

[Dempster, Laird, y Rubin1977] Dempster, Arthur, Nan Laird, y Donald Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological), pp. 1–38.

[Duarte2007] Duarte, Abraham. 2007. Metaheurísticas. Volume 22. Librería-Editorial Dykinson.

[Edwards1998] Edwards, Ward. 1998. Hailfinder: tools for and experiences with Bayesian normative modeling. American Psychologist 53 (4): 416.

[Elvira2002] Elvira, Consortium. 2002. Elvira: An Environment for Probabilistic Graphical Models. Edited by J.A. Gámez y A. Salmerón, Proceedings of the 1st European Workshop on Probabilistic Graphical Models. 222–230.

[Filiberto et al.2011] Filiberto, Yaima, Bello Rafael, Caballero Yailé, y Frías Mabel. 2011. Algoritmo para el Aprendizaje de Reglas de Clasificación basado en la Teoría de los Conjuntos Aproximados Extendida. Dyna 169:63.

[Friedman, Geiger, y Goldszmidt1997] Friedman, Nir, Dan Geiger, y Moises Goldszmidt. 1997. Bayesian network classifiers. Machine learning 29 (2-3): 131–163.

[Friedman et al.2000] Friedman, Nir, Linial Michal, Nachman Iftach, y Peter Dana. 2000. Using Bayesian networks to analyze expression data. Computational Biology 7 (3-4): 601–620.

[Friedman et al.1997] Friedman, Nir, Goldszmidt Moises, Heckerman David, y Russell Stuart. 1997. Challenge: what is the impact of Bayesian Networks on learning? Proceedings of the 15th international joint conference on Artifical intelligence-Volume 1. Morgan Kaufmann Publishers Inc., 10–15.

[García2009] García, Francisco. 2009. Modelos bayesianos para la clasificación supervisada: aplicaciones al análisis de datos de expresión genética, Tesis Doctoral, Universidad de Granada.

[García et al.2006] García, Juan, López Jorge, Cano Jesús, Gea Ana, y De la Fuente Leticia. 2006. Aplicación de las redes bayesianas al modelado de las actitudes emprendedoras. Actas del IV Congreso de Metodología de Encuestas. 235–242.

[García, Kuna, y Villatoro2010] García, Ramón, Horacio Kuna, y Francisco Villatoro. 2010. Identificación de causales de abandono de estudios universitarios. TE & ET.

[García et al.2009] García, Salvador, Daniel Molina, Manuel Molina, y Herrera Francisco. 2009. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC2005 Special Session on Real Parameter Optimization. Journal of Heuristics 15:617–644.

[Garner1995] Garner, Stephen y others. 1995. Weka: The waikato environment for knowledge analysis. Proceedings of the New Zealand computer science research students conference. Citeseer, 57–64.

[Gunduz y Fokoue2013] Gunduz, Necla, y Ernest Fokoue. 2013. UCI Machine Learning Repository.

[Han et al.2004] Han, Jiawei, Pei Jian, Yin Yiwen, y Mao Runying. 2004. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data mining and knowledge discovery 8 (1): 53–87.

[Han, Pei, y Kamber2011] Han, Jiawei, Jian Pei, y Micheline Kamber. 2011. Data mining: concepts and techniques. Elsevier.

[Heckerman1998] Heckerman, David. 1998. A tutorial on learning with Bayesian networks. Springer.

[Heckerman, Geiger, y Chickering1995] Heckerman, David, Dan Geiger, y David Chickering. 1995. Learning Bayesian networks: The combination of knowledge and statistical data. Machine learning 20 (3): 197–243.

[Heckerman, Kadie, y Listgarten2007] Heckerman, David, Carl Kadie, y Jennifer Listgarten. 2007. Leveraging information across HLA alleles/supertypes improves epitope prediction. Journal of Computational Biology 14 (6): 736–746.

[Holm1979] Holm, Sture. 1979. A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics 6 (2): 65–70.

[Holte1993] Holte, Robert. 1993. Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. Machine Learning 11 (1): 63–90.

[Iman y James1980] Iman, Davenport, y Ronald James. 1980. Approximations of the critical region of the Friedman statistic. Commu. Stat. 18:571–595.

[Irani et al.1993] Irani, Keki, Cheng Jie, Fayyad Usama, y Qian Zhaogang. 1993. Applying machine learning to semiconductor manufacturing. iEEE Expert 8 (1): 41–47.

[Kohavi1995a] Kohavi, Ron. 1995a. The power of decision tables. European Conference on Machine Learning. Springer, 174–189.

[Kohavi1995b] Kohavi, Ron. 1995b. Wrappers for performance enhancement and oblivious decision graphs. Ph.D. diss., University of Citeseer.

[Koller y Friedman2009] Koller, Daphne, y Nir Friedman. 2009. Probabilistic graphical models: principles and techniques. MIT press.

[Langley y Sage1994] Langley, Pat, y Stephanie Sage. 1994. Induction of selective Bayesian classifiers. Proceedings of the Tenth international Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc., 399–406.

[Larrañaga, Inza, y Abdelmalik2000] Larrañaga, Pedro, Iñaki Inza, y Moujahid Abdelmalik. 2000. Árboles de clasificación. Universidad del País Vasco: Departamento de Ciencias de la Computación e Inteligencia Artificial.

[Larrañaga et al.2005] Larrañaga, Pedro, Lozano José, Peña José, y Inza Iñaki. 2005. Editorial of the special issue on probabilistic graphical models in classification. Machine Learning 59 (3): 211–212.

[Lichman2013] Lichman, Moshe. 2013. UCI Machine Learning Repository.

[Luna2014] Luna, José María. 2014. Nuevos Retos en Minería de Reglas de Asociación, Un Enfoque Basado en Programación Genética, Tesis Doctoral, Universidad de Granada.

[Lykourentzou et al.2009] Lykourentzou, Ioanna, Giannoukos Ioannis, Nikolopoulos Vassilis, Mpardis George, y Vassili Loumos. 2009. Dropout prediction in e-learning courses through the combination of machine learning techniques. Computers & Education 53 (3): 950–965.

[Magaña, Montesinos, y Hernández2006] Magaña, Martha, Osval Montesinos, y Carlos Hernández. 2006. Análisis de la evolución de los resultados obtenidos por los profesores en las evaluaciones ESDEPED y las realizadas por los estudiantes. Revista de la Educación Superior 35 (140): 29–48.

[Molina et al.2013] Molina, Daniel, Puris Amilkar, Bello Rafael, y Francisco Herrera. 2013. Variable mesh optimization for the 2013 CEC special session Niching methods for multimodal optimization. Evolutionary Computation (CEC), 2013 IEEE Congress on. IEEE, 87–94.

[Morales y Salmerón2003] Morales, María, y Antonio Salmerón. 2003. Análisis del alumnado de la Universidad de Almería mediante redes bayesianas.

[Mosier1951] Mosier, Charles. 1951. The need and means of cross validation. Problems and designs of cross-validation. Educational and Psychological Measurement.

[Murphy2012] Murphy, Kevin. 2012. Machine Learning: a Probabilistic Perspective. MIT press.

[Navarro et al.2009] Navarro, Ricardo, Puris Amilkar, Puris Rafael, y Francisco Herrera. 2009. Estudio del desempeño de la optimización basada en mallas variables en problemas con óptimos en las fronteras del espacio búsqueda. Revista Cubana de Ciencias Informáticas 3, no. 3-4.

[Neapolitan2004] Neapolitan, Richard. 2004. Learning Bayesian Networks. Prentice Hall Upper Saddle River.

[Oviedo et al.2015] Oviedo, Byron, Puris Amilkar, Villacís Annabelle, Delgado Diana, y Moreno Ana. 2015. Análisis de datos educativos utilizando Redes Bayesianas, Latin American and Caribbean Conference for Engineering and Technology LACCEI 2015.

[Oviedo et al.2016] Oviedo, Byron, Puris Amilkar, Moreira Luis, y Serafín Moral. 2016. Learning Bayesian network by a Mesh of points. IEEE CEC 2016.

[Oviedo et al.2014] Oviedo, Byron, Guanín Jorge, Díaz Eduardo, y Amilkar Puris. 2014. Optimización basada en Mallas Variables: Caso de estudio Viajante de Comercio, The 12th Latin American and Caribbean Conference for Engineering and Technology, LACCEI 2014.

[Oviedo, Moral, y Puris2016] Oviedo, Byron, Serafín Moral, y Amilkar Puris. 2016. A hierarchical clustering method: Applications to educational data. Intelligent Data Analysis 20 (4): 933–951.

[Pearl1988] Pearl, Judea. 1988. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann.

[Porcel, Dapozo, y López2010] Porcel, Eduardo, Gladys Dapozo, y María López. 2010. Predicción del rendimiento académico de alumnos de primer año de la FACENA (UNNE) en función de su caracterización socioeducativa. Revista Electrónica de Investigación Educativa 12 (2): 1–21.

[Puris2010] Puris, Amilkar. 2010. Desarrollo de metaheurısticas poblacionales para la solución de problemas complejos. Ph.D. diss., Universidad Central de Las Villas, Santa Clara, Cuba.

[Puris et al.2012] Puris, Amilkar, Bello Rafael, Molina Daniel, y Francisco Herrera. 2012. Variable mesh optimization for continuous optimization problems. Soft Computing 16 (3): 511–525.

[Quinlan1986] Quinlan, Ross. 1986. Induction of decision trees. Machine learning 1 (1): 81–106.

[Quinlan1993] Quinlan, Ross. 1993. C4.5: Programs for Machine Learning. Volume 16. San Mateo, CA: Morgan Kaufmann Publishers, 235–240.

[Rendell y Seshu1990] Rendell, Larry, y Raj Seshu. 1990. Learning hard concepts through constructive induction: Framework and rationale. Computational Intelligence 6 (4): 247–270.

[Sahami1996] Sahami, Mehran. 1996. Learning Limited Dependence Bayesian Classifiers. KDD, Volume 96. 335–338.

[Schwarz1978] Schwarz, G. 1978. Estimating the dimension of a model. The Annals of Statistics 6 (2): 461–464.

[Serrano et al.1998] Serrano, Luis, Batanero Carmen, Ortíz Jesús, y Jesús Cañizares. 1998. Heurísticas y sesgos en el razonamiento probabilístico de los estudiantes de secundaria. Educación Matemática 10 (1): 7–25.

[Servente y García2002] Servente, Magdalena, y Martínez García. 2002. Algoritmos TDIDT Aplicados a la Minería Inteligente. Revista del Instituto Tecnológico de Buenos Aires 26:39–57.

[Sierra2006] Sierra, Basilio. 2006. Aprendizaje automático: conceptos básicos y avanzados. Prentice-Hall.

[Spirtes, Glymour, y Scheines2000] Spirtes, Peter, Clark Glymour, y Richard Scheines. 2000. Causation, prediction, and search. MIT press.

[Srikant y Agrawal1996] Srikant, Ramakrishnan, y Rakesh Agrawal. 1996. Mining quantitative association rules in large relational tables. Acm Sigmod Record, Volume 25. ACM, 1–12.

[Sucar2008] Sucar, Luis. 2008. Clasificadores Bayesianos: De Datos a Conceptos, Instituto Nacional de Astrofsica, Óptica y Electrónica, México.

[Sucar y Martínez-Arroyo2011] Sucar, Luis, y Miriam Martínez-Arroyo. 2011. Aprendizaje de Clasificadores Bayesianos Dinámicos.

[Utgoff, Berkman, y Clouse1997] Utgoff, Paul, Neil Berkman, y Jeffery Clouse. 1997. Decision tree induction based on efficient tree restructuring. Machine Learning 29 (1): 5–44.

[Webb y Pazzani1998] Webb, Geoffrey, y Michael Pazzani. 1998. Adjusted probability naive Bayesian induction. Proceedings of the Eleventh Australian Joint Conference on Artificial Intelligence. Springer-Verlag, 285–295.

[Zaki2000] Zaki, Mohammed. 2000. Scalable algorithms for association mining. IEEE Transactions on Knowledge and Data Engineering 12 (3): 372–390.