Capítulo 3 REDES BAYESIANAS

3.1 Presentación intuitiva

Antes de presentar formalmente la teoría matemática de las redes bayesianas, explicaremos mediante ejemplos sencillos el significado intuitivo de los conceptos que después introduciremos, utilizando el mismo esquema seguido en (Díez, 1994). En una red bayesiana, cada nodo corresponde a una variable, que a su vez representa una entidad del mundo real. Por tanto, de aquí en adelante hablaremos indistintamente de nodos y variables, y los denotaremos con letras mayúsculas. Para referirnos a un valor cualquiera de una variable *X* utilizaremos la misma letra en minúscula *x*. Los arcos que unen los nodos indican relaciones de *influencia causal* entre ellas. Veamos unos ejemplos sencillos[7] en el contexto de esta tesis, es decir, de modelado del alumno.

Ejemplo 3.1

La red bayesiana no trivial más simple que podemos imaginar consta de dos variables, que llamaremos *C* y P_1, y un arco desde la primera hasta la segunda, como se muestra en la Figura 3.1.

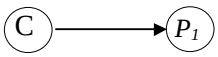


Figura 3.1 Red bayesiana con dos nodos.

Para concretar el ejemplo, supongamos que *C* representa el conocimiento del alumno sobre cierto concepto C y P_1 su capacidad de resolver correctamente cierta pregunta P_1 relativa a dicho concepto. Entonces, que el alumno sepa el concepto C

[7] Los ejemplos son adaptaciones de los presentados en (Díez, 1994).

tiene influencia causal en que sea capaz de responder bien a la pregunta P_1, lo cual se expresa mediante el arco dirigido que aparece en la Figura.

La notación que usaremos será la siguiente: si *X* es una variable binaria, denotaremos por $+x$ la presencia de aquello a lo que representa y por $\neg x$ a su ausencia. Así, por ejemplo en este caso $+c$ significará "el alumno conoce el concepto C" y $\neg c$ "el alumno no conoce el concepto C"; $+p_1$ significará "el alumno es capaz de resolver correctamente la pregunta P_1" y $\neg p_1$ "el alumno no es capaz de resolver correctamente la pregunta P_1".

La información cuantitativa de una red bayesiana viene dada por:

- La probabilidad a priori de los nodos que no tienen padres.
- La probabilidad condicionada de los nodos con padres.

Por tanto, en nuestro ejemplo, los datos que debemos conocer son $P(c)$ y $P(p_1 / c)$.

Así, la red bayesiana completa sería la que se muestra en la Figura 3.2.

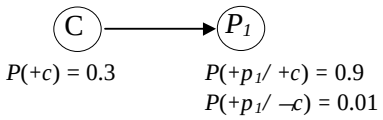
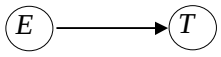


Figura 3.2 Red bayesiana con parámetros.

Veamos qué significado[8] tienen en este caso estos valores:

---

[8] En el campo de la medicina, estos parámetros tienen una interpretación muy sencilla: supongamos que tenemos una red que representa la relación entre padecer o no cierta enfermedad E y el resultado de un test T que se utiliza para el diagnóstico de la enfermedad E. La red bayesiana es:



Entonces:

- $P(+e)$ representa el tanto por ciento de la población en estudio que padece la enfermedad E, es decir, la *prevalencia* de E.
- $P(+t / +e)$ indica el tanto por ciento de pacientes que dan positivo en el test T entre los que padecen la enfermedad E. Esto se conoce como *sensibilidad* del test.
- $P(+t / \neg e)$ indica el tanto por ciento de pacientes que dan positivo en el test T entre los que no padecen la enfermedad E. A la probabilidad complementaria $P(\neg e / \neg t)$, es decir, a la proporción de pacientes que dan negativo en el test entre los que no padecen la enfermedad se le llama *especificidad* del test.

En medicina siempre se buscan los tests con mayor grado de sensibilidad y especificidad. Esta semántica puede extenderse al caso del modelado del alumno, así que a partir de ahora hablaremos también de la sensibilidad y especificidad de una pregunta para un concepto.

- $P(+c)$ = 0.3 indica que el 30% de los alumnos del grupo en estudio conocen el concepto.
- $P(+p_1 / +c)$ = 0.9 indica que el 90% de los alumnos que conocen el concepto C responden correctamente a la pregunta P_1. Esto quiere decir que incluso los alumnos que conocen el concepto pueden tener un despiste y contestar mal a la pregunta (en una proporción del 10%).
- $P(+p_1 / \neg c)$ = 0.01 significa que sólo el 1% de los alumnos que no conocen el concepto C son capaces de contestar correctamente a la pregunta P_1. Este parámetro indica por tanto qué alumnos que no conocen el concepto pueden adivinar la respuesta correcta a la pregunta P_1.

Conociendo estos datos, podemos calcular:

a) La probabilidad a priori de que un alumno cualquiera conteste correctamente a la pregunta P_1,

$$P(+p_1) = P(+p_1 / +c) \cdot P(+c) + P(+p_1 / \neg c) \cdot P(\neg c) = 0.277$$
$$P(\neg p_1) = P(\neg p_1 / +c) \cdot P(+c) + P(\neg p_1 / \neg c) \cdot P(\neg c) = 0.723$$

b) Las probabilidades a posteriori dada una evidencia observada ***e***, $P^*(c) = P(c / \mathbf{e})$.

Supongamos que la evidencia observada es que cierto alumno ha contestado correctamente a la pregunta P_1. ¿Qué probabilidad hay ahora de que conozca el concepto C?. Si no existiese posibilidad ninguna de que un alumno que no conozca el concepto C responda bien a la pregunta P_1, esa probabilidad sería 1, pero como no es así tenemos que calcular $P^*(+c) = P(+c / +p_1)$. Para ello aplicamos el teorema de Bayes y obtenemos que:

$$P^*(+c) = P(+c / +p_1) = \frac{P(+c) \cdot P(+p_1 / +c)}{P(+p_1)} = \frac{0.3 \cdot 0.9}{0.277} = 0.97473$$

De la misma forma podríamos calcular $P^*(\neg c)$:

$$P^*(\neg c) = P(\neg c / +p_1) = \frac{P(\neg c) \cdot P(+p_1 / \neg c)}{P(+p_1)} = \frac{0.7 \cdot 0.01}{0.277} = 0.02527$$

que, por supuesto, es la probabilidad complementaria.

La expresión general del teorema de Bayes que hemos utilizado es:

$$P^*(c) = P(c / p_1) = \frac{P(c) \cdot P(p_1 / c)}{P(p_1)}.$$

Por razones que quedarán claras más adelante, vamos a reescribirla como:

$$P^*(c) = \alpha \cdot P(c) \cdot \lambda_{P_1}(c),$$

donde $\alpha = [P(p)]^{-1}$ y $\lambda_{P_1}(c) = P(p / c)$.

Con la fórmula expresada de esta forma, queda claro que la probabilidad a posteriori de la variable *C* depende fundamentalmente de la probabilidad a priori de *C* y de las probabilidades condicionadas de *P* dado *C*, puesto que α juega simplemente el papel de una constante de normalización.

Utilizando esta nueva expresión, podemos repetir los cálculos:

$P^*(+c) = \alpha \cdot 0.3 \cdot 0.9 = 0.27 \cdot \alpha$
$P^*(\neg c) = \alpha \cdot 0.7 \cdot 0.01 = 0.007 \cdot \alpha$

Y normalizando obtenemos el mismo resultado que antes.

Para el caso en que el alumno respondiese incorrectamente, la probabilidad a posteriori de que conozca el concepto se calcula con un procedimiento totalmente análogo.

Ejemplo 3.2

Supongamos que ampliamos el modelo anterior añadiendo otra pregunta P_2. La red bayesiana se muestra en la Figura 3.3.

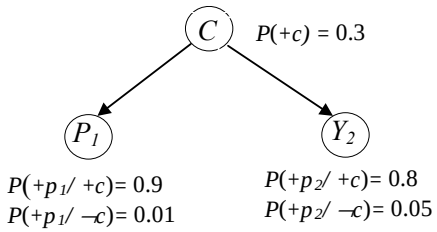


Figura 3.3 Red bayesiana con tres nodos.

Observamos que la pregunta P_2 es menos *sensible* y menos *específica* para el concepto C que la pregunta P_1.

Veamos qué tipo de conclusiones podemos extraer a partir de esta información.

a) Supongamos que el alumno responde correctamente a la pregunta P_2, es decir, que a la evidencia disponible es $e = \{+p_2\}$. Entonces, podemos calcular como antes la probabilidad a posteriori de que el alumno conozca el concepto C dado que ha respondido bien a la pregunta P_2:

$P^*(+c) = P(+c / +p_2) = \alpha \cdot 0.3 \cdot 0.8 = 0.8727.$
$P^*(\neg c) = P(\neg c / +p_2) = \alpha \cdot 0.7 \cdot 0.05 = 0.1272.$

Como podemos observar, la probabilidad de que un alumno que conteste bien a la pregunta P_2 conozca el concepto C es 0.8727, resultado algo más bajo que en el caso de que conteste bien a la pregunta P_1 (0.97473), lo cual se explica por la menor sensibilidad y especificidad de la pregunta P_2.

b) Supongamos que $e = \{+p_1, +p_2\}$. ¿Cuál es ahora $P^*(c) = P(c / +p_1, +p_2)$?

Para calcularla, usamos de nuevo el teorema de Bayes:

$$P^*(c) = P(c / p_1, p_2) = \frac{P(c) \cdot P(p_1, p_2 / c)}{P(p_1, p_2)}.$$

Pero ahora vemos que hay datos del problema que no conocemos, como $P(p_1,p_2)$ y $P(p_1,p_2 / c)$. Para poder seguir nuestros cálculos, necesitamos realizar unas hipótesis adicionales, que se denominan *hipótesis de independencia condicional*. En concreto, vamos a suponer que las variables P_1 y P_2 son independientes dados su padre común en la red (C) es decir:

$$P(p_1,p_2 / c) = P(p_1 / c) \cdot P(p_2 / c).$$

Si suponemos esto podremos continuar con los cálculos porque $P(p_1,p_2)$ se obtendrá como una constante de normalización.

¿Qué significa aceptar esta hipótesis?. Significa aceptar que, conocido que un alumno conoce el concepto C, el hecho de que responda bien o mal a la pregunta P_2 no depende de si responde bien o mal a la pregunta P_1, lo cual parece razonable.

Para continuar con la nueva formulación que introdujimos en el ejemplo 1, vamos a denotar por $\lambda(c)$ al producto $\lambda_{P_1}(c) \cdot \lambda_{P_2}(c)$. Entonces tendríamos que:

$$P^*(c) = \alpha \cdot P(c) \cdot \lambda(c).$$

En nuestro ejemplo, $\boldsymbol{e} = \{+p_1, +p_2\}$, así que:

$$\lambda(+c) = \lambda_{P_1}(+c) \cdot \lambda_{P_2}(+c) = 0.9 \cdot 0.8 = 0.72$$
$$\lambda(\neg c) = \lambda_{P_1}(\neg c) \cdot \lambda_{P_2}(\neg c) = 0.01 \cdot 0.05 = 0.0005$$

Por tanto:

$$P^*(+c) = 0.216 \cdot \alpha = 0.9984.$$
$$P^*(\neg c) = 0.00035 \cdot \alpha = 0.0016.$$

Como era de esperar, cuando tenemos dos evidencias en favor de que el alumno conozca el concepto, la probabilidad resultante es mayor que la correspondiente a cada una de ellas por separado.

c) Aún podemos extraer más información de este ejemplo. Supongamos ahora que hay un alumno que ha respondido correctamente a la pregunta P_2, es decir, que la evidencia disponible es $\boldsymbol{e} = \{+p_2\}$. ¿Cuál es la probabilidad de que si le planteamos la pregunta P_1 también la responda correctamente?. Para ello, debemos calcular ahora $P(p_1 / +p_2)$. Por teoría elemental de probabilidad, sabemos que:

$$P^*(p_1) = P(p_1 / p_2) = \sum_c P(p_1 / c, p_2)\, P(c / p_2) = \sum_c P(p_1 / c, p_2) \cdot \frac{P(c, p_2)}{P(p_2)}.$$

Aplicando la hipótesis de independencia condicional y definiendo

$\pi_{P_1}(c) = P(c, p_2) = P(c) \cdot P(p_2 / c)$.
$\alpha = [P(p_2)]^{-1}$.

la expresión anterior nos queda:

$$P^*(p_1) = \alpha \cdot \sum_c P(p_1 / c) \cdot \pi_{P_1}(c) .$$

Sustituyendo los valores numéricos de nuestro ejemplo, tenemos que:

$\pi_{P_1}(+c) = 0.3 \cdot 0.8 = 0.24$
$\pi_{P_1}(\neg c) = 0.7 \cdot 0.05 = 0.035$

Y, finalmente,

$P^*(+p_1) = \alpha\,(0.9 \cdot 0.24 + 0.1 \cdot 0.035) = 0.7867$
$P^*(\neg p_1) = \alpha\,(0.1 \cdot 0.24 + 0.99 \cdot 0.035) = 0.2133$

Resulta interesante comparar las expresiones utilizadas para calcular la probabilidad a priori $P(p_1)$ y la a posteriori $P^*(p_1)$. Para la primera, utilizábamos $P(c)$, ahora hemos utilizado $\pi_{P_1}(+c)$, que representa la probabilidad de *c* tras considerar la evidencia relativa a c *diferente* de P_1.

Vemos así cómo la información que aporta el nodo P_2 modifica la probabilidad de *C*, y, en consecuencia, también la de P_1. El carácter simultáneamente ascendente y descendente del mecanismo de propagación es lo que nos permite utilizar la red tanto para realizar inferencias abductivas (cuál es la combinación de valores de las variables que mejor explica la evidencia disponible) como predictivas (cuál es la probabilidad de obtener cierto resultado en el futuro). Un mismo nodo puede ser tanto fuente de información como objeto de predicción, dependiendo de cuáles sean los hallazgos disponibles y el objeto del proceso de inferencias.

Terminada ya esta presentación intuitiva, vamos a introducir formalmente las redes bayesianas.

3.2 Definición formal de red bayesiana

Antes de definir formalmente las redes bayesianas, vamos a definir algunos conceptos de teoría de grafos y teoría de la probabilidad:

Definiciones previas

- **Arco.** Es un par ordenado (X, Y). Esta definición de arco corresponde a lo que en otros lugares se denomina *arco dirigido*. En la representación gráfica, un arco (X,Y) viene dado por una flecha desde X hasta Y.

- **Grafo dirigido.** Es un par $G = (N, A)$ donde N es un conjunto de nodos y A un conjunto de arcos definidos sobre los nodos.

- **Grafo no dirigido.** Es un par $G = (N,A)$ donde N es un conjunto de nodos y A un conjunto de arcos no orientados (es decir, pares no ordenados (X,Y)) definidos sobre los nodos.

- **Camino.** Es una secuencia ordenada de nodos $(X_{i_1}, \ldots, X_{i_r})$ tal que $\forall\, j = 1, ..., r\text{-}1$, ó bien el arco $X_j \rightarrow X_{j+1} \in A$ o bien el arco $X_{j+1} \rightarrow X_j \in A$.

- **Camino dirigido.** Es una secuencia ordenada de nodos $(X_{i_1}, \ldots, X_{i_r})$ tal que para todo $j = 1, ..., r\text{-}1$ el arco $X_j \rightarrow X_{j+1} \in A$.

- **Ciclo.** Es un camino $(X_{i_1}, ..., X_{i_r})$ en el que $X_{i_1} = X_{i_r}$.

- **Ciclo dirigido.** Es un camino dirigido $(X_{i_1}, \ldots, X_{i_r})$ en el que $X_{i_1} = X_{i_r}$.

- **Padre.** X es un *padre* de Y si y sólo si existe un arco $X \rightarrow Y$. Se dice también que Y es **hijo** de X. Al conjunto de los padres de X se representa como $pa(X)$, y al de los hijos de X por $S(X)$.

- **Antepasado.** X es un *antepasado* de Z si y sólo si existe un camino dirigido de X a Z.

- **Conjunto ancestral** de un nodo X es un conjunto que contiene a X y a todos sus antepasados.

- **Descendiente.** Z es un *descendiente* de X si y sólo si X es un antepasado de Z. Al conjunto de los descendientes de X lo denotaremos por $de(X)$.

- **Adyacentes de *X*.** Es el conjunto resultante de la unión de los padres de X y de los hijos de X. Lo denotaremos por $ady(X)$.

- **Familia de un nodo *X*.** Es el conjunto de nodos formado por X y los padres de X, $pa(X)$, $F(X) = \{X\} \cup pa(X)$.

- **Familia de probabilidad de un nodo.** Llamamos *familia de probabilidad* de X a la probabilidad condicional $f_x = P(X \mid pa(X))$.

- **Grafo completo.** Un grafo no dirigido G se dice que es *completo* si existe una arista entre cada par de nodos.

- **Conjunto completo.** Sea $G = (N, A)$ un grafo no dirigido. Dado un subconjunto X_c de N, se dice que es *completo* si existe en A una arista entre cada par de nodos de X_c.

- **Grupo maximal.** Sea $G = (N, A)$ un grafo no dirigido. Decimos que un conjunto completo X_c es un *grupo maximal* (*clique* en inglés) si no es subconjunto propio de otro conjunto completo en G.

- **Orden.** Dado un conjunto de nodos $N = \{X_1, \ldots, X_n\}$, un *orden* σ es una biyección que asigna a cada número entre 1 y n un nodo de $\{X_1, \ldots, X_n\}$.

- **Variable proposicional** es una variable aleatoria que toma un conjunto exhaustivo y excluyente de valores. La denotaremos con letras mayúsculas, por ejemplo X, y a un valor cualquiera de la variable con la misma letra en minúscula, x.

- **Separación condicional**. Dadas tres variables proposicionales X, Y y Z, diremos que *Z separa condicionalmente* a X e Y si X e Y son independientes dado Z.

Definición (red bayesiana)

Una red bayesiana es:

- Un conjunto de variables proposicionales, V,
- un conjunto E de relaciones binarias definidas sobre las variables de V,
- una distribución de probabilidad conjunta P definida sobre las variables de V,

tales que:

- (V, E) es un grafo acíclico, conexo y dirigido G.
- (G, P) cumple las *hipótesis de independencia condicional*, también llamadas de *separación direccional*, que se enuncian a continuación:

***Hipótesis de independencia condicional* (*o de separación direccional*)**

Un grafo acíclico conexo y dirigido $G = (V, E)$ y una distribución de probabilidad conjunta P definida sobre las variables del grafo se dice que cumplen las hipótesis de independencia condicional si para toda variable X de V se tiene que el conjunto de los padres directos de X *separa condicionalmente* a X de todo subconjunto $\boldsymbol{Y}$ de V que no contenga a X ni a ninguno de sus descendientes. Es decir,

$$\forall\, X \in V \; y \; \forall\, \boldsymbol{Y} \subseteq V - \{X \cup de(X)\} \quad \text{se tiene que} \quad P(X / pa(X), Y) = P(X / pa(X))$$

En la definición de red bayesiana, hemos partido de una distribución de probabilidad conjunta para las variables. Si tenemos una red con N nodos y con variables binarias, haría falta conocer 2^N-1 valores. Sin embargo, las condiciones de independencia dadas por la separación direccional permiten que no sea necesario conocer todos estos valores, puesto que, como veremos en el siguiente Teorema, la distribución de probabilidad conjunta se puede expresar como producto de las distribuciones condicionadas de cada nodo dados sus padres.

Teorema (Factorización de la probabilidad)

Dada una red bayesiana, la distribución de probabilidad conjunta puede expresarse como:

$$P(x_1, ..., x_n) = \prod_i P(x_i / pa(x_i)).$$

Demostración

Es fácil construir una ordenación de las variables en la que los padres de cada nodo aparezcan siempre después de él. Supongamos por tanto que la ordenación $\{X_1, \ldots, X_n\}$ cumple dicha propiedad. Por tanto:

$$P(x_1, \ldots, x_n) = \prod_i P(x_i / x_{i+1}, \ldots, x_n).$$

Pero por la forma de escoger la ordenación, el conjunto $\{X_{i+1}, ..., X_n\}$ incluye a todos los padres de X, y, en consecuencia, la separación direccional nos dice que

$$P(x_i / x_{i+1}, \ldots, x_n) = P(x_i / pa(x_i))$$

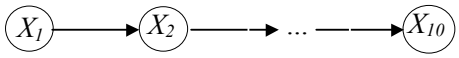
❑

La importancia de este teorema es que nos permite describir una red bayesiana a partir de la probabilidad condicionada de cada nodo (o la probabilidad a priori en el caso de nodos sin padres) en lugar de dar la probabilidad conjunta, que,

- requiere un número de parámetros exponencial en el número de nodos, y
- plantea el problema de verificar la separación direccional.

Sin embargo, el número de parámetros requerido para dar las probabilidades condicionadas es mucho menor (proporcional al número de nodos), nos permite reconstruir la distribución condicionada aplicando el teorema, y además, a la hora de pedirle estos valores al experto, son valores con pleno significado, como vimos en el ejemplo 1.

Por ejemplo, para la red bayesiana dada por:



Suponiendo que todas las variables fuesen binarias, serían necesarios 2^{10}-1 parámetros para dar la distribución conjunta, sin embargo, si construimos la distribución conjunta a partir de los 19 valores que especifican las condicionadas, tendremos además asegurado que se satisfacen las hipótesis de independencia condicional.

Terminada ya esta presentación intuitiva, vamos a presentar algunos algoritmos de propagación de probabilidades.

3.3 Algoritmos de propagación de probabilidades

Una vez que tenemos la red bayesiana nos interesará realizar consultas sobre las variables incluidas en la misma. En el campo de los sistemas expertos el principal interés se centra en ver cómo los valores que toman ciertas variables afectan a las probabilidades del resto. Si intentáramos afrontar estos cálculos aplicando el Teorema de Bayes, la ley de probabilidad total y las condiciones de independencia condicional necesitaríamos realizar un número de operaciones que crece exponencialmente con el número de variables de la red, y se convertiría en una tarea computacionalmente intratable. Los algoritmos de propagación de probabilidades utilizan las relaciones de independencia implícitas en la estructura de una red bayesiana para calcular las probabilidades de cada uno de los nodos dada la evidencia disponible de una forma más eficiente. Calculadas estas probabilidades, se pueden utilizar tanto para hacer inferencias de tipo abductivo como predictivo.

Para entender el funcionamiento de los algoritmos, empezaremos con el más simple que existe, que es el algoritmo para redes con forma de árbol. Posteriormente describiremos los llamados métodos de agrupamiento, y en particular el algoritmo que hemos implementado en esta tesis, que es el algoritmo HUGIN (Jensen, Olesen et al., 1990).

3.3.1 Algoritmo de propagación para redes en forma de árbol

El primer método de propagación para redes bayesianas que se desarrolló es el algoritmo de propagación en árboles (Pearl, 1982). La idea consiste en que cuando se modifica la información asociada a un nodo, éste traspasa la información a sus nodos vecinos mediante un conjunto de mensajes; estos nodos, a su vez, procesan la información recibida junto con la que ellos poseen y la pasan a sus nodos vecinos (aún no modificados) y así sucesivamente hasta que todos los nodos han actualizado su información. La ventaja de este algoritmo es que funciona en un orden de tiempo lineal respecto al número de nodos de la red, pero su principal limitación es que sólo se puede aplicar a redes con estructura de árbol, restricción demasiado fuerte, ya que, en la práctica, la presencia de otras estructuras resulta muy habitual. La descripción que hacemos del algoritmo se basa en la que aparece en (Neapolitan, 1990). El algoritmo consta de dos fases:

Fase de inicialización

En esta fase se obtienen las probabilidades a priori de todos los nodos de la red, obteniendo un estado inicial de la red que denotaremos por S_0.

Fase de actualización

Cuando una variable se instancia se actualiza el estado de la red, obteniéndose las probabilidades a posteriori de las variables de la red basadas en la evidencia considerada, adoptando la red un estado que denotaremos por S_1. Este paso se repite cada vez que una variable se instancia, obteniéndose los sucesivos estados de la red.

La idea principal en la que se basa el algoritmo es la siguiente:

Cada vez que una variable se instancia, informa a sus nodos vecinos mediante el paso de lo que llamaremos *mensajes*, de la siguiente forma:

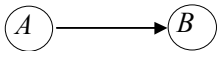
- La variable envía a su padre un mensaje, que llamaremos el λ-mensaje, para informarle de que ha cambiado su valor.

- La variable envía a todos sus hijos un mensaje, que llamaremos el π-mensaje, para informarles de que ha cambiado su valor.

Así, la información se va propagando por la red tanto en sentido ascendente como descendente.

Estos mensajes asignan a cada variable unos valores que llamaremos λ-valor y π-valor. Multiplicando estos valores obtendremos las probabilidades a posteriori de cada una de las variables de la red.

Tanto los valores como los mensajes son vectores de números. Por ejemplo, supongamos que tenemos el arco:



en el que la variable *A* toma tres valores posibles que denotaremos a_1, a_2, y a_3, y la variable *B* es binaria y toma valores b_1 y b_2. Tendríamos que:

- Si *B* se instancia, enviará un λ-mensaje a *A*, $\lambda_B(A) = (\lambda_B(a_1), \lambda_B(a_2), \lambda_B(a_3))$.
- Si *A* se instancia, enviará un π-mensaje a *B*, $\pi_B(A) = (\pi_B(a_1), \pi_B(a_2), \pi_B(a_3))$.

En función de esos mensajes, tendremos un λ-valor y π-valor para *A*,

$\lambda(A) = (\lambda(a_1), \lambda(a_2), \lambda(a_3))$ y $\pi(A) = (\pi(a_1), \pi(a_2), \pi(a_3))$.

Y también un λ-valor y un π-valor para B,

$\lambda(B) = (\lambda(b_1), \lambda(b_2))$ y $\pi(B) = (\pi(b_1), \pi(b_2))$.

Multiplicando los valores y normalizando, obtendremos las probabilidades asociadas a *A* o a *B*, según sea el caso.

Los ejemplos presentados en la sección 3.1 sirven para ilustrar el mecanismo descrito. A continuación, presentamos las fórmulas para el cálculo de los λ y π-mensajes, λ y π-valores y probabilidades *P** y el algoritmo.

3.3.1.1.1 Fórmulas para el cálculo de λ y π–mensajes, λ y π-valores y probabilidades P*:

1. Si *B* es un hijo de *A*, *B* tiene *k* valores posibles y *A* *m* valores posibles, entonces para $j=1,\ldots,m$ el λ-mensaje de *B* a *A* viene dado por:

$$\lambda_B(a_j) = \sum_{i=1}^{k} P(b_i/a_j) \cdot \lambda(b_i).$$

2. Si *B* es hijo de *A* y *A* tiene m valores posibles, entonces para $j=1,\ldots,m$, el π-mensaje de *A* a *B* viene dado por:

$$\pi_B(a_j) = \begin{cases} \pi(a_j) \cdot \prod_{\substack{c \in S(A) \\ c \neq B}} \lambda_c(a_j) & \text{si } A \text{ no ha sido instanciada} \\ 1 & \text{si } A = a_j \\ 0 & \text{si } A \neq a_j \end{cases}$$

3. Si *B* tiene *k* valores posibles entonces para $i=1,\ldots,k$ el λ-valor de *B* viene dado por:

$$\lambda(b_i) = \begin{cases} \prod_{C \in S(B)} \lambda_C(b_i) & \text{si } B \text{ no ha sido instanciada} \\ 1 & \text{si } B = b_i \\ 0 & \text{si } B \neq b_i. \end{cases}$$

4. Si A es padre de B, B tiene k valores posibles y A tiene m valores posibles, entonces, para i=1,...,k, el π-valor de B viene dado por;

$$\pi(b_i) = \sum_{j=1}^{m} P(b_i/a_j) \cdot \pi_B(a_j).$$

5. Si B es una variable con k posibles valores, entonces, para i = 1,...,k, la probabilidad a posteriori basada en las variables instanciadas se calcula como:

$$P^*(b_i) = \alpha \cdot \lambda(b_i) \cdot \pi(b_i)$$

A continuación presentamos el algoritmo:

Algoritmo 3.1. Algoritmo de propagación en redes con forma de árbol

1. *Inicialización.*

A. Inicializar todos los λ-mensajes y λ-valores a 1.
B. Si la raíz A tiene m posibles valores, entonces para $j = 1,...,m$, sea

$$\pi(a_j) = P(a_j).$$

C. Para todos los hijos B de la raíz A, hacer

Enviar un nuevo π-mensaje a B usando la fórmula 2.
(En ese momento comenzará un flujo de propagación debido al procedimiento de actualización 2.C).

2. *Actualización.*

Cuando una variable se instancia o una variable recibe un λ o π-mensaje, se usa uno de los siguientes procedimientos de actualización:

A. Si una variable B se instancia a un valor b_j, entonces:

A.1. Inicializar $P^*(b_j) = 1$ y $P^*(b_i) = 0$, para todo $i \neq j$.
A.2. Calcular $\lambda(B)$ usando la fórmula 3.
A.3. Enviar un nuevo λ-mensaje al padre de B usando la fórmula 1.
A.4. Enviar nuevos π–mensajes a los hijos de B usando la fórmula 2.

B. Si una variable B recibe un nuevo λ-mensaje de uno de sus hijos y la variable B no ha sido instanciada todavía, entonces:

B.1. Calcular el nuevo valor de $\lambda(B)$ usando la fórmula 3.
B.2. Calcular el nuevo valor de $P^*(B)$ usando la fórmula 5.
B.3. Enviar un nuevo λ-mensaje al padre de B usando la fórmula 1.
B.4. Enviar nuevos π–mensajes a los otros hijos de B usando la fórmula 2.

C. Si una variable B recibe un nuevo π–mensaje de su padre y la variable B no ha sido instanciada todavía, entonces:

C.1. Calcular el nuevo valor de $\pi(B)$ usando la fórmula 4.
C.2. Calcular el nuevo valor de $P^*(B)$ usando la fórmula 5.
C.3. Enviar nuevos π–mensajes a los hijos de B usando la fórmula 2.

3.3.2 Algoritmos de propagación exactos basados en técnicas de agrupamiento.

Para el caso general se han desarrollado otros algoritmos. El problema de la propagación en redes bayesianas es NP duro (Cooper, 1990), lo que significa que no es posible obtener un algoritmo de complejidad polinomial para el problema de la propagación en redes bayesianas con una topología general. Para intentar paliar esta complejidad se han desarrollado algoritmos de propagación aproximada, que también son NP-duros en la precisión de la estimación, pero que tienen tiempos de ejecución lineales en el número de variables si la precisión se mantiene fija (Dagum & Luby, 1993). Básicamente, podríamos agrupar los métodos exactos en dos categorías: los algoritmos de condicionamiento y los algoritmos de agrupamiento. Los algoritmos de agrupamiento han tenido un gran auge a partir del trabajo realizado por Lauritzen y Spiegelhalter (Lauritzen & Spiegelhalter, 1988), que fue mejorado posteriormente por Jensen, Olesen y Andersen en el llamado algoritmo HUGIN (Jensen, Olesen et al., 1990). El programa comercial HUGIN[9], desarrollado por investigadores vinculados a la Universidad de Aalborg y considerado como la herramienta más eficaz para el desarrollo y la computación de redes bayesianas, está basado en esta técnica. En sus primeras versiones utilizaba la búsqueda de máxima cardinalidad (Tarjan & Yannakakis, 1984) como algoritmo de triangulación, pero las versiones más modernas se basan en los algoritmos heurísticos investigados por Kjærulff (Kjærulff, 1990). Nosotros hemos optado por el uso de métodos exactos, y de entre ellos por los algoritmos de agrupamiento, que son los que más éxito han tenido, y que básicamente consisten en transformar la red en otra estructura gráfica acíclica y no dirigida cuyos nodos están formados por conjuntos de variables. Pasamos por tanto a describir dichos métodos[10].

Antes de explicar los algoritmos daremos una breve idea de en qué consisten. Los algoritmos de agrupamiento se desarrollan en dos fases. El objetivo de la primera fase es reducir el grafo a una estructura llamada *árbol de grupos maximales*. Esta estructura será utilizada en la segunda fase, de modo que los cálculos necesarios para el cómputo de las probabilidades puedan realizarse de manera local en cada grupo, y pasarse de un grupo a otro utilizando mensajes entre los grupos, de forma similar al caso de propagación en árboles. A continuación describimos brevemente las etapas más importantes de cada fase:

---

[9] HUGIN está disponible en http://www.hugin.dk, donde también podemos encontrar una versión de evaluación que ofrece todas las capacidades de la herramienta comercial limitando sólo el número de nodos de la red (200).

[10] Dado que en este trabajo hemos utilizado algoritmos exactos de agrupamiento, no describiremos ni los algoritmos de condicionamiento ni los algoritmos aproximados. Todos estos algoritmos pueden encontrarse descritos en detalle en (Castillo, Gutiérrez et al., 1997).

- En la fase 1, el primer paso del método consiste en añadir enlaces para *moralizar* el grafo (se dice que un grafo no dirigido es moral cuando todos los padres están "casados", es decir, relacionados). Luego suprimimos la dirección de los arcos, con lo cual nos queda un grafo no dirigido. El paso siguiente consiste en *triangular* el grafo. Decimos que un grafo está *triangulado* si para cada ciclo de longitud mayor o igual a cuatro (hablamos ya de ciclos no dirigidos) hay al menos un arco que conecta dos nodos no consecutivos. Por tanto, el proceso de triangular un grafo consiste en añadir los arcos necesarios para que no haya ciclos de cuatro o más nodos. Este paso es crucial en la eficiencia del algoritmo, como se verá más adelante.

  A continuación hay que formar los grupos maximales. Una vez obtenidos los grupos maximales, se ordenan y conforme a esta ordenación se les dota de una estructura que se denomina árbol de grupos maximales.

- En la fase 2 se parte del árbol de grupos maximales obtenido en la fase anterior. En primer lugar hay que inicializar la red asignando a cada grupo maximal una función de sus variables que se llama *función potencial*, que consiste en una distribución de probabilidad marginal obtenida a partir de las tablas de probabilidad condicional y de la evidencia disponible. Estas funciones serán posteriormente utilizadas para actualizar las probabilidades a posteriori a medida que se vaya adquiriendo información. La probabilidad correspondiente a una variable se calcula marginalizando y normalizando la tabla de probabilidad de uno de los grupos maximales que contienen dicha variable.

A modo de resumen, las etapas del algoritmo de agrupamiento son:

Fase 1. Obtención de un árbol de grupos maximales.
- Obtención del grafo moral G_M.
- Obtención del grafo triangular G_T.
- Construcción del árbol de grupos maximales a partir de G_T y de la lista de grupos maximales asociada.

Fase 2. Cálculo de las probabilidades.
- Cálculo de potenciales.
- Construcción de una factorización de la distribución de probabilidad conjunta.
- Fase de absorción de evidencias (si existen).
- Fase de propagación.

Utilizaremos este resumen para ir describiendo las etapas del algoritmo. Para una mejor comprensión del algoritmo iremos aplicando cada uno de estos pasos a un ejemplo de prueba conocido como la red Asia (Lauritzen & Spiegelhalter, 1988).

Ejemplo 3.3 (*Red Asia*)

La Figura 3.4 muestra el grafo acíclico dirigido correspondiente a la red causal Asia que modela el siguiente problema:

La tuberculosis (*T*) y el cáncer de pulmón (*L*) son causas de que un paciente esté enfermo del pulmón (*E*). Si una persona está enferma del pulmón, entonces la enfermedad puede provocarle disnea (*D*) y también pueden influir en el resultado de una prueba de rayos X en el pecho (*X*). Por otra parte, la bronquitis (*B*) es otra causa de disnea. Además, el hecho de haber visitado recientemente Asia (*A*) incrementa la probabilidad de padecer tuberculosis, mientras que el hecho de ser fumador (*S*) es una de las causas posibles de la bronquitis y del cáncer de pulmón.

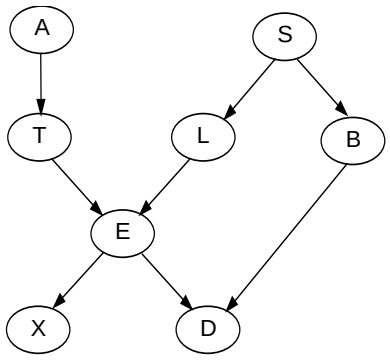


Figura 3.4 *Red causal Asia.*

Comenzaremos por describir cómo se obtiene un árbol de grupos a partir de una red bayesiana.

Fase 1: Obtención de un árbol de grupos maximales

En primer lugar es necesario construir el grafo triangular. Como hemos dicho anteriormente, un *grafo triangular* es aquel en el que para cada ciclo de longitud mayor o igual que cuatro existe al menos una arista entre dos nodos no consecutivos del ciclo.

Para obtener un grafo no dirigido triangular G_T a partir de un grafo dirigido G hay que obtener primero el *grafo moral* G_M. El procedimiento a seguir para obtener el grafo moral asociado a un grafo acíclico dirigido dado es muy sencillo y se detalla en el siguiente algoritmo.

Algoritmo 3.2. Obtención del grafo moral G_M

Entrada: Un grafo acíclico dirigido *G*
Salida: El grafo moral G_M

1. Para cada nodo X_i de *G*, añadir las aristas necesarias para que $pa(X_i)$ sea un conjunto completo.
2. Eliminar la direccionalidad de las aristas.

Como vemos, el nombre de grafo moral viene de que el procedimiento consiste en "casar" a los padres de los nodos y después eliminar la direccionalidad de las aristas. El grafo moral G_M representa un grafo no dirigido para el que se cumple que todas las independencias presentes en él lo están también en el grafo dirigido *G*, aunque no se cumple la implicación en el otro sentido. Al unir los padres de un nodo se consigue mantener las dependencias que se pierden al eliminar la dirección de los arcos.

En el ejemplo de la Red Asia, la aplicación de este algoritmo produce el grafo moral asociado que aparece en la Figura 3.5

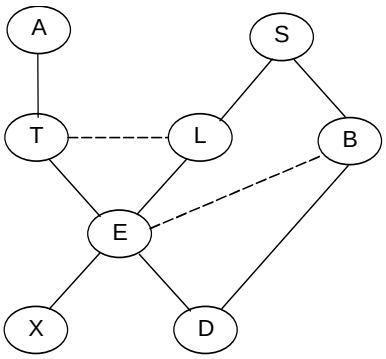


Figura 3.5 Grafo moral asociado a la red causal Asia.

Una vez que tenemos el grafo moral G_M, que consiste en el conjunto de nodos *U* y el conjunto de enlaces *E*, esto es, $G_M = (U, E)$, tendremos que añadir las aristas necesarias para romper los ciclos de longitud mayor o igual que cuatro y obtener así un grafo triangular. Este proceso se conoce como *rellenado de aristas*, pues partiendo de un grafo moral y de un orden de eliminación de las variables se van añadiendo las aristas pertinentes hasta obtener un grafo triangular que posteriormente utilizaremos para obtener el conjunto de grupos maximales. La idea es la siguiente: se selecciona el primer nodo según cierto orden especificado, y se añaden las aristas necesarias para hacer que el conjunto de adyacentes de ese nodo sea completo. Hecho esto, se elimina el nodo y las aristas que lo contengan, se elige el siguiente

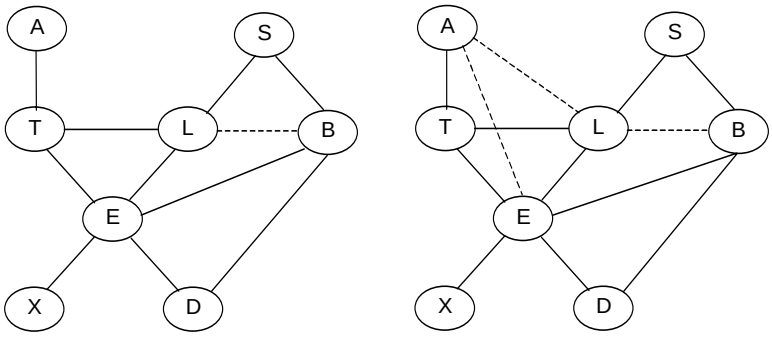
nodo en el orden, y se repite el proceso hasta que no queden nodos. Este procedimiento se detalla en el siguiente algoritmo:

Algoritmo 3.3. Rellenado de aristas

Entrada: Un grafo moral G_M =(U, E) y una secuencia de eliminación de nodos σ
Salida: El grafo triangulado G_T

1. E' ← ∅.
2. Para cada i = 1 hasta card(U), hacer:
   - X_i ← σ(i).
   - Sea E'' el conjunto de aristas que es necesario añadir al grafo para que *ady*(X_i) sea un conjunto completo.
   - Añadir al grafo el conjunto de aristas E''.
   - Eliminar del grafo el nodo X_i y todas sus aristas.
   - $E' \leftarrow E' \cup E''$.
3. Eliminar la direccionalidad de las aristas.
4. Devolver el grafo triangulado G_T = (U, $E \cup E^{\wedge}$).

Al proceso que consiste en dado un nodo X hacer el conjunto *ady*(X) completo y eliminar del grafo el nodo X y todas sus aristas se le llama *eliminación del nodo X*. La triangulación obtenida depende mucho del orden o secuencia de eliminación de las variables, ya que en función del mismo se añadirán más o menos aristas al grafo. En la Figura 3.6 vemos dos grafos triangulados distintos para la red Asia. El primero de ellos lo produce la secuencia σ_1=(A,X,T,D,E,L,S,B), y el segundo la secuencia σ_2 = (T,A,X,S,D,L,B,E).



Secuencia σ_1 *Secuencia* σ_2

Figura 3.6 Dos grafos triangulares distintos para la red Asia.

El grafo triangulado se usará para obtener una descomposición del grafo en un conjunto de grupos maximales. Por ejemplo, los grupos maximales obtenidos para cada una de las triangulaciones anteriores son los que se muestran en la Tabla 3.1.

Grupos maximales (σ_1)	Grupos maximales (σ_2)
{*A*,*T*}	{*A*,*T*,*L*,*E*}
{*T*,*L*,*E*}	{*X*,*E*}
{*X*,*E*}	{*S*,*L*,*B*}
{*S*,*L*,*B*}	{*L*,*B*,*E*}
{*L*,*B*,*E*}	{*E*,*B*,*D*}
{*E*,*B*,*D*}	

Tabla 3.1 Grupos maximales obtenidos con las secuencias de eliminación σ_1 y σ_2.

Es dentro de estos grupos maximales donde se efectuarán los cálculos de las propagaciones, así que la triangulación obtenida va a afectar en gran medida a la eficiencia. Por tanto, es crucial buscar un orden que produzca la mejor triangulación posible. La bondad de una triangulación depende del problema que se vaya a resolver utilizando el grafo triangulado. Vamos a definir diferentes medidas que ayudarán a determinar la bondad de una triangulación.

Definiciones (Tamaño y Peso de un conjunto de variables)

Sea X un conjunto de variables. Se define:

- **Tamaño** de X es el número de variables de X,
- **Peso** de X como peso(X) = $\prod_{X_i \in X} \text{card}(\Omega_{X_i})$,

donde Ω_{X_i} representa el conjunto de posibles valores que puede tomar la variable X_i. Así, si por ejemplo tenemos un conjunto X con tres variables X_1, X_2 y X_3, donde X_1 y X_2 son binarias y X_3 es una variable discreta con cinco valores distintos, el peso del conjunto X sería $2 \cdot 2 \cdot 5 = 20$.

El **peso** de una triangulación se define como la suma de los pesos de cada uno de los grupos maximales que la forman. Para la propagación en redes bayesianas, no sólo es importante el número de aristas que se añaden al grafo para obtener el grafo triangulado, sino también el número de estados posibles de cada nodo. Veamos esto en un ejemplo: consideremos de nuevo el grafo moral asociado a la red Asia, donde junto a cada nodo aparece el número de estados posibles de la variable correspondiente. Dicho grafo se muestra en la Figura 3.7.

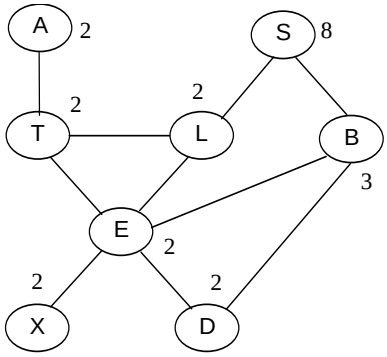


Figura 3.7 Grafo moral con número de estados posibles de cada variable.

Hay dos triangulaciones posibles que añaden sólo un enlace: la que añade el enlace *LB* y la que añade el enlace *SE*. Los grupos maximales obtenidos en cada caso se muestran en la Tabla 3.2.

Añadiendo *LB*		Añadiendo *SE*	
Grupos maximales	Peso grupo	Grupos maximales	Peso grupo
{*A*,*T*}	4	{*A*,*T*}	4
{*T*,*L*,*E*}	8	{*T*,*L*,*E*}	8
{*X*,*E*}	4	{*X*,*E*}	4
{*S*,*L*,*B*}	48	{*S*,*L*,*E*}	32
{*L*,*B*,*E*}	12	{*S*,*B*,*E*}	48
{*E*,*B*,*D*}	12	{*E*,*B*,*D*}	12

Tabla 3.2 Grupos maximales y pesos.

Atendiendo a estos resultados observamos que es preferible una secuencia de eliminación que añada la arista *LB*, ya que el peso de la triangulación sería 88, frente al peso de la obtenida añadiendo SE que es 108.

Sin embargo, el problema de la obtención de una secuencia de eliminación óptima es NP-duro (Wen, 1991). A pesar de eso, existen heurísticos que, en la mayoría de los casos, producen triangulaciones óptimas, según el estudio comparativo realizado por Kjærulff (Kjærulff, 1990). Los heurísticos analizados en este trabajo consisten en eliminar primero aquellos nodos cuyo conjunto de adyacentes es ya completo, con objeto de no añadir aristas innecesarias. Una vez eliminados estos nodos, hay varios criterios diferentes para escoger el siguiente nodo a eliminar:

- *Criterio de relleno mínimo*, que es aquel que selecciona para ser eliminado el nodo que necesita el número mínimo de enlaces para hacer completo el conjunto de sus padres, y por tanto genera la triangulación que añade menos enlaces.

- *Criterio de peso mínimo*, en el que se selecciona para ser eliminado el nodo que produce el grupo maximal de peso mínimo, y que por tanto genera grafos triangulados de peso mínimo.
- *Criterio de tamaño mínimo*, que selecciona para ser eliminado el nodo que produce el grupo maximal de tamaño mínimo, y que por tanto genera grafos triangulados para los cuales la suma de los tamaños de los grupos maximales es mínima.

Dependiendo de para qué vaya a ser utilizado el grafo triangular, el concepto de optimalidad se puede definir de forma diferente. Para el caso de la propagación en redes bayesianas lo que va a influir en la eficiencia de la propagación es el peso de los grupos maximales, y por tanto el heurístico seleccionado será el de peso mínimo, que es el que tiene por defecto la herramienta HUGIN (aunque se ofrece la posibilidad de elegir cualquiera de los otros heurísticos).

En nuestra aplicación hemos implementado, además del heurístico de peso mínimo, un nuevo heurístico que consiste simplemente en ir eliminando aquellos nodos con menor número de adyacentes. La razón es que la estructura de las redes que vamos a utilizar hace que con este heurístico se obtengan siempre triangulaciones óptimas a un coste menor que los heurísticos anteriores, ya que no hay que comprobar si los conjuntos de adyacentes de todos los nodos son o no completos. En caso de empate se elige el nodo X_i tal que $ady(X_i)$ tiene peso mínimo, ya que en el caso de que sea necesario añadir una arista es preferible hacerlo en el grupo de menor peso[11].

A continuación presentamos el algoritmo que, dado un orden (que puede ser el obtenido aplicando el heurístico de peso mínimo o el nuevo heurístico presentado), realiza la triangulación del grafo y obtiene el conjunto de grupos maximales.

[11] Sin embargo, dado que para otras estructuras de la red el heurístico de peso mínimo genera triangulaciones mejores, hemos implementado ambos algoritmos de forma que pueda seleccionarse el más apropiado para cada caso.

Algoritmo 3.4. Algoritmo de triangulación y obtención de grupos maximales

Entrada: Un grafo moral $G_M = (U,E)$ y una secuencia de eliminación σ.
Salida: El grafo triangulado G_T y el conjunto C de grupos maximales.

1. E'←∅.
2. C ←∅.
3. Para i =1 hasta card(U) hacer:
   - $X_i \leftarrow \sigma(i)$.
   - Sea E'' el conjunto de aristas que es necesario añadir a G_M para que $ady(X_i)$ sea un conjunto completo.
   - Añadir a G_M el conjunto de aristas E''.
   - Eliminar en G_M el nodo X_i y todas sus aristas.
   - $E' \leftarrow E' \cup E''$.
   - $G_i \leftarrow ady(X_i) \cup X_i$.
   - Si no existe G_j en L tal que $G_i \subset G_j$.
     - $C \leftarrow C \cup G_i$
4. $G_T \leftarrow (U, E \cup E')$
5. Devolver el conjunto de grupos maximales C y el grafo triangulado G_T como salida.

Una vez obtenido el grafo triangulado G_T y el conjunto C de grupos maximales, el siguiente paso es dotarlos de una estructura en forma de árbol. Para ello es necesario seguir los siguientes pasos:

1. Numerar los nodos del grafo. Para ello se puede usar el *algoritmo de búsqueda de máxima cardinalidad* (Tarjan & Yannakakis, 1984) (Algoritmo 3.5.).

Algoritmo 3.5. Algoritmo de búsqueda de máxima cardinalidad

Entrada: Un grafo no dirigido $G = (U,E)$.
Salida: Un orden σ para las variables de U.

1. Elegir un nodo cualquiera X_i de U, y hacer $\sigma(1) = X_i$.
2. $j \leftarrow 2$.
3. Mientras $j \leq \text{card}(U)$ hacer:
   - Entre los nodos no numerados todavía, seleccionar el nodo X_i que tenga mayor número de vecinos ya numerados (romper empates arbitrariamente).
   - $\sigma(j) \leftarrow X_i$.
   - $j \leftarrow j + 1$.
4. Devolver σ como salida.

Para ilustrar el funcionamiento del algoritmo, vamos a aplicarlo a uno de los grafos triangulados obtenidos para nuestro ejemplo (el obtenido con la secuencia σ_1). Si empezamos por el nodo A, uno de los posibles órdenes obtenidos con el algoritmo de búsqueda de máxima cardinalidad (hay varios empates) es el que aparece en la Figura 3.8:

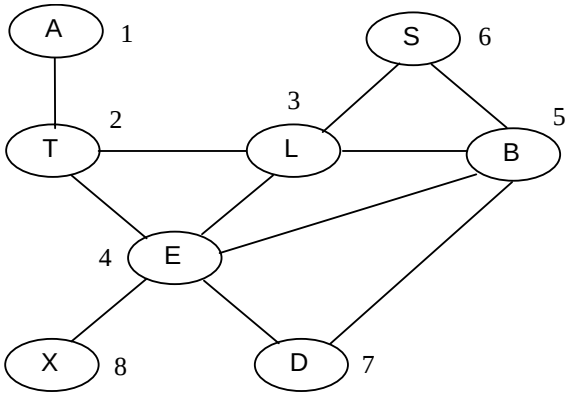


Figura 3.8 Una posible ordenación para los nodos del grafo moral de la Red Asia utilizando el algoritmo 3.5.

2. A continuación se numeran los grupos maximales de acuerdo con el nodo de menor orden que contienen. Los empates se deshacen considerando el siguiente nodo contenido en cada grupo. En nuestro ejemplo, para el orden obtenido en la etapa anterior la numeración de los grupos maximales sería la que se muestra en la Tabla 3.3.

Grupos maximales	Orden
{*A*,*T*}	1
{*T*,*L*,*E*}	2
{*L*,*B*,*E*}	3
{*S*,*L*,*B*}	4
{*E*,*B*,*D*}	5
{*X*,*E*}	6

Tabla 3.3 Ordenación de los grupos.

3. Supongamos ahora que los grupos maximales están ya ordenados de acuerdo al orden obtenido en la etapa anterior. Ahora debemos obtener los conjuntos residuales y separadores. Para ello, cada grupo G_i se divide en dos conjuntos disjuntos, llamados *residual* R_i y *separador* S_i y definidos por:

$$R_i = \begin{cases} G_i \setminus S_i & \text{si } i > 1, \\ G_i & \text{si } i = 1. \end{cases}$$

$$S_i = \begin{cases} G_i \cap (G_1 \cup G_2 \cup \ldots G_{i-1}) & \text{si } i > 1, \\ \varnothing & \text{si } i = 1. \end{cases}$$

Una vez obtenidos los conjuntos residual y separador asociados a cada grupo maximal, dotamos al conjunto de grupos maximales de estructura de árbol, teniendo en cuenta la siguiente consideración: cualquier grupo G_j que contenga al separador S_i con $j < i$ será un posible padre del grupo G_i. En consecuencia, la raíz del árbol será G_1. Para nodos con más de un posible padre se elige aquel cuya intersección con el grupo en cuestión es máxima (en caso de empate elegiremos el de peso mínimo, y si aún hay empate, arbitrariamente). El árbol construido de esta forma cumple las siguientes condiciones[12]:

- Toda familia $F(X_i)$ de la red original se encuentra en al menos un grupo del árbol construido.

- Se verifica la propiedad de *intersección dinámica*, es decir, para cada par de grupos G y G' del árbol cuya intersección $I = G \cap G'$ sea distinta del vacío, se verifica que I está incluido en todos los grupos que hay en el camino que une G con G'.

[12] La satisfacción de estas condiciones está garantizada gracias a la aplicación el algoritmo de búsqueda por máxima cardinalidad. La demostración puede encontrarse en el capítulo 4 de (Castillo, Gutiérrez et al., 1997).

En la Tabla 3.4. mostramos el conjunto de separadores y residuales obtenidos a partir del orden determinado en la etapa 2, y el conjunto de posibles padres para cada nodo:

Grupo	Nodos	Residual	Separador	Padres
1	{*A*,*T*}	*A*,*T*	∅	
2	{*T*,*L*,*E*}	*L*,*E*	*T*	1
3	{*L*,*B*,*E*}	*B*	*L*,*E*	2
4	{*S*,*L*,*B*}	*S*	*L*,*B*	3
5	{*E*,*B*,*D*}	*D*	*E*,*B*	3
6	{*X*,*E*}	*X*	*E*	2,3,5

Tabla 3.4 Conjuntos residual y separador y posibles padres.

El árbol de grupos maximales se representa como un árbol en el que dentro de cada grupo se ponen entre llaves los nodos del conjunto separador. En nuestro ejemplo, el árbol construido es el que aparece en la Figura 3.9 (donde se ha optado por elegir como padre del grupo 6 al grupo 2).

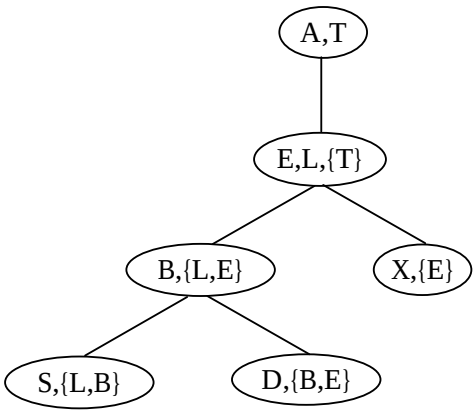


Figura 3.9 Árbol de grupos maximales para la red Asia.

El algoritmo implementado para la obtención del árbol realiza los pasos anteriores de forma simultánea (ordenación, obtención residuales y separadores y construcción del árbol), buscando una estructura para el árbol que mejore la eficiencia en la propagación. Para ello, en lugar de utilizar el algoritmo de máxima cardinalidad hemos preferido dar prioridad en el orden a los grupos de menor peso, ya que de esta forma se reduce el número de operaciones durante la etapa de propagación. Elegimos por tanto como raíz el grupo maximal de menos peso. Este grupo de nodos lo añadimos a un conjunto unión, que en principio estará vacío, y en el que mantendremos el conjunto de todos los grupos maximales considerados en cada momento del algoritmo. El siguiente grupo seleccionado será aquel cuya intersección con el conjunto *U* sea mayor. En caso de empate elegimos de nuevo el grupo de menor peso. El algoritmo es:

Algoritmo 3.6. Algoritmo para la ordenación de los grupos maximales, obtención de residuales y separadores y construcción del árbol

Entrada: un conjunto C de grupos maximales.
Salida: un árbol A de grupos maximales.

1. Elegir el grupo maximal G_i de C de peso mínimo.
2. $U \leftarrow G_i$.
3. $C \leftarrow C \,/\, \{G_i\}$.
4. $S_G \leftarrow \varnothing$.
5. $R_G \leftarrow G_i$.
6. $A \leftarrow \varnothing$.
7. Añadir G_i al árbol A como grupo raíz.
8. Mientras C sea no vacío hacer:
   - Elegir el grupo G_i cuya intersección con U sea la de mayor tamaño (en caso de empate elegir el grupo de menor peso).
   - $U \leftarrow U \cup G_i$.
   - $C \leftarrow C \,/\, \{G_i\}$.
   - $S_{Gi} \leftarrow G_i \cap U$.
   - $R_{G/} \leftarrow G_i \,/\, S_{Gi}$.
   - Elegir el grupo maximal G_j de A tal que $S_{Gi} \subset G_j$ (en caso de haber más de uno se toma aquel cuya intersección con G_i es mayor)
   - Añadir G_i al árbol A con G_j como padre.
9. Devolver el árbol de grupos maximales A.

Una vez obtenido el árbol de grupos maximales empieza la fase 2.

Fase 2. Cálculo de probabilidades y propagación de evidencias

El primer paso en la fase 2 consiste en obtener una factorización de la distribución de probabilidad conjunta a partir del árbol de grupos maximales. Para ello, asociaremos a cada grupo G_i del árbol una función $\psi_{G_i} : \Omega_{G_i} \rightarrow IR^+$, llamada *función de potencial* o simplemente *potencial*. Los potenciales de cada grupo se obtienen de la siguiente forma:

- Se inicializan todos los potenciales a 1.
- Para cada variable X_i de la red bayesiana original, se identifica un grupo G_j que contenga a la familia de X_i (si hay varios, se elige uno arbitrariamente)[13], y se multiplica ψ_{G_i} por la familia de probabilidad f_{X_i}.

[13] Cada nodo de la red ha de quedar asociado a uno y sólo uno de los grupos maximales que contienen a su familia.

Bajo estas condiciones se dice que el par $(\{G_1,...,G_t\},\{\psi_{G_1},..., \psi_{G_t}\})$ es una factorización de la distribución de probabilidad conjunta definida por la red, ya que se verifica que la probabilidad conjunta admite una factorización como producto de las familias de probabilidad, es decir:

$$P(x) = \prod_{i=1}^{t} \Psi_{G_i}(x^{\downarrow G_i}), \qquad \forall x \in \Omega_u ,$$

donde Ω_u es el conjunto de posibles estados que pueden tomar las variables, y $x^{\downarrow G_i}$ denota la proyección del vector x sobre el conjunto G_i.

Vamos a suponer que en el ejemplo de la red Asia asignamos los nodos A y T al grupo 1, el nodo E al grupo 2, ningún nodo al grupo 3, los nodos L, S y B al grupo 4, el nodo D al grupo 5 y el nodo X al grupo 6. Los potenciales son entonces:

- $\psi_{G_1}(x^{\downarrow G_1}) = f_A(x^{\downarrow Fa(A)}) \cdot f_T(x^{\downarrow Fa(T)}) = P(T/A) \cdot P(A)$.
- $\psi_{G_2}(x^{\downarrow G_2}) = P(E/T,L)$.
- $\psi_{G_3}(x^{\downarrow G_3}) = 1$.
- $\psi_{G_4}(x^{\downarrow G_4}) = P(B/S) \cdot P(L/S) \cdot P(S)$.
- $\psi_{G_5}(x^{\downarrow G_5}) = P(D/E,B)$.
- $\psi_{G_6}(x^{\downarrow G_6}) = P(X/E)$.

Con lo cual, la distribución conjunta puede factorizarse como producto de las seis funciones potenciales, es decir:

$$P(A,T,E,L,S,B,D,X) = P(T/A)\cdot P(A)\cdot P(E/T,L)\cdot P(B/S)\cdot P(L/S)\cdot P(S)\cdot P(D/E,B)\cdot P(X/E).$$

Como el árbol construido es independiente de la evidencia observada, puede ser utilizado para realizar cualquier propagación.

Ahora vamos a explicar cómo se realiza el cálculo de las probabilidades de los nodos de la red. Si existen evidencias debemos en primer lugar cambiar las funciones potenciales de acuerdo a dichas evidencias (si no existen seguimos con el mismo conjunto de funciones potenciales). Esta etapa se conoce con el nombre de *absorción de evidencias*. Supongamos que disponemos de la evidencia $E^* = e$, donde E^* es un conjunto de variables y e el conjunto de valores que toman dichas variables. En ese caso tenemos que actualizar los potenciales, y este proceso se puede llevar a cabo de dos formas distintas:

1. Mantenemos el mismo conjunto de nodos X y de grupos maximales $C = \{G_1,\ldots,G_m\}$. En este caso sólo es necesario modificar las funciones potenciales que contengan nodos evidencia de la forma siguiente: para cada grupo maximal G_i con algún nodo evidencia definimos la función ψ^*_i mediante:

$$\psi_i^*(x^{\downarrow G_i}) = \begin{cases} 0 & \text{si algún valor de } x^{\downarrow G_i} \text{ no es consistente con } e, \\ \psi_i(x^{\downarrow G_i}) & \text{en otro caso.} \end{cases} \quad (*)$$

Las funciones potenciales del resto de los grupos maximales no se modifican. Entonces, tenemos que:

$$P(x \ / \ e) \propto \prod_{i=1}^{m} \psi_i^*(x^{\downarrow G_i}).$$

2. Eliminando de G los nodos evidencia. Esta opción implica cambiar también el conjunto de grupos maximales y las representaciones de las funciones potenciales. La nueva representación potencial (C^*,ψ^*) se define en G^*, donde $G^* = G \setminus E$, C^* es la nueva lista de grupos maximales y ψ^* son los nuevos potenciales, que contienen la evidencia y se obtienen de la siguiente forma: para cada grupo maximal G_i en C tal que $G_i \cap E \neq \varnothing$ incluimos $C_i \setminus E$ en C^* y obtenemos el nuevo potencial de la siguiente forma:

$$\psi_i^*(x^{\downarrow G_i^*}) = \psi_i(x^{\downarrow G_i} \setminus e, E = e). \quad (**)$$

Para el resto de los grupos maximales que no tienen nodos evidencia no se necesitan cambios ni en el grupo maximal ni en la función potencial. Con ello, tendremos que:

$$P(x \,/\, e) \propto \prod_{i=1}^{m} \psi_i^*(x^{\downarrow G_i}).$$

Por tanto, el método puede aplicarse en ambos casos para calcular las probabilidades a posteriori cuando se observan evidencias. En el primer caso se continúa con la misma estructura pero se usan recursos de memoria innecesariamente. En el segundo, obtenemos una reducción en los recursos de memoria necesarios, pero necesitamos cambiar los datos y las estructuras de almacenamiento. Nosotros hemos preferido implementar el primer método por las razones expuestas anteriormente. De esta forma, mantenemos la misma estructura del árbol durante todo el proceso de absorción y propagación.

En el algoritmo HUGIN se toma la primera opción, y en el de Lauritzen-Spiegelhalter, la segunda. Una vez calculados los nuevos potenciales, los dos algoritmos proceden de igual manera para propagar la evidencia disponible.

En la llamada fase de propagación de la evidencia, se utiliza el árbol de grupos maximales junto con los potenciales asociados a cada uno de ellos (en los que habremos absorbido la evidencia si es que la había) para calcular las probabilidades de cada nodo. La propiedad de intersección consecutiva asegura que la distribución de probabilidad conjunta puede expresarse como:

$$P(x) = P(x^{\downarrow R_t} / x^{\downarrow S_t}) \cdot P(x^{\downarrow R_{t-1}} / x^{\downarrow S_{t-1}}) \cdot \ldots \cdot P(x^{\downarrow R_2} / x^{\downarrow S_2}) \cdot P(x^{\downarrow R_1}), \qquad \forall x \in \Omega_u$$

En nuestro ejemplo, esta expresión se convierte en:

$$P(A,T,E,L,S,B,D,X) = P(A,T) \cdot P(E, L / T) \cdot P(B / L,E) \cdot P(S / L,B) \cdot P(D / E,B) \cdot P(X / E).$$

El problema ahora está en pasar de la representación de potenciales a esta representación. Esto se lleva a cabo en dos fases:

- En la etapa ascendente, el objetivo es calcular para cada grupo maximal G_i la probabilidad de su residual condicionado a su separador, esto es, $P(x^{\downarrow R_i} \mid x^{\downarrow S_i})$. Para ello, realizamos las siguientes operaciones:

  - Para i = m hasta i =1 hacer:

    - Calcular $m_i(x^{\downarrow S_i}) = \sum_{R_i} \psi_i(x^{\downarrow G_i})$ (Si S_i es vacío, $m_i(\varnothing) = \sum_{G_i} P(x^{\downarrow G_i})$)

    - Asignar $P(x^{\downarrow R_i} \mid x^{\downarrow S_i}) = \dfrac{\psi_i(x^{\downarrow G_i})}{m_i(x^{\downarrow S_i})}$.

    - Si G_j padre de G_i, reemplazar la función potencial de G_j por $\psi_j(x^{\downarrow G_j}) \cdot m_i(x^{\downarrow S_i})$.

  Tras finalizar esta etapa tendremos calculado para cada grupo maximal G_i, con i =1,...,m, las funciones de probabilidad $P(x^{\downarrow R_i} \mid x^{\downarrow S_i})$.

- En la etapa descendente, utilizamos las distribuciones de probabilidad $P(x^{\downarrow R_i} \mid x^{\downarrow S_i})$. para ir obteniendo sucesivamente para cada grupo marginal la distribución conjunta de sus variables. Esto se hace siguiendo los pasos que se detallan a continuación:

  - Asignar $P(x^{\downarrow G_1}) = P(x^{\downarrow R_1} \mid x^{\downarrow S_1})$
  - Para i = 2 hasta i = m hacer:
    - Calcular $P(x^{\downarrow S_i}) = \sum_{s_j} P(x^{\downarrow G_j})$.
    - Asignar $P(x^{\downarrow G_i}) = P(x^{\downarrow R_i} \mid x^{\downarrow S_i})\, P(x^{\downarrow S_i})$.

Una vez obtenidas estas distribuciones conjuntas, para calcular la probabilidad de cualquier nodo bastará con identificar el grupo maximal de menor tamaño que lo contenga y marginalizar la distribución conjunta, es decir, debemos realizar las siguientes operaciones:

- Para i = 1 hasta n, hacer:
  - Elegir el grupo maximal G_j de menor tamaño que contenga al nodo X_i.
  - Asignar $P(X_i / E^*) \propto \sum_{G_j \setminus X_i} P(x^{\downarrow G_j})$

A modo de resumen de todo el proceso de absorción y propagación de evidencias, presentamos el algoritmo 3.7.

Algoritmo 3.7. Absorción y propagación de evidencias

Entrada
- Una red bayesiana con n nodos, $X_1, \ldots, X_n$.
- Un árbol de grupos maximales A asociado a la red, $G_1, \ldots, G_m$, junto con las funciones potenciales de cada grupo maximal $\psi_j(x^{\downarrow G_j})$.
- Un conjunto de evidencias E^*.

Salida: Las funciones de probabilidad condicionadas de cada nodo X_i de la red dada la evidencia disponible, $P(X_i / E^*)$.

Etapa 1: Absorción de evidencias

1. Absorber la evidencia $E = e$ en las funciones potenciales, utilizando (*) ó (**).

Etapa 2: Propagación de evidencias

Fase ascendente

2. Para $i = m$ hasta $i = 1$, hacer:
   - Calcular $m_i(x^{\downarrow S_i}) = \sum_{R_i} \psi_i(x^{\downarrow G_i})$.
   - Asignar $P(x^{\downarrow R_i} \mid x^{\downarrow S_i}) = \dfrac{\psi_i(x^{\downarrow G_i})}{m_i(x^{\downarrow S_i})}$.
   - Si G_j es padre de G_i, $\psi_j(x^{\downarrow G_j}) \leftarrow \psi_j(x^{\downarrow G_j})\, m_i(x^{\downarrow S_i})$.

Fase descendente

✦ *Cálculo de probabilidades conjuntas para cada grupo:*

3. Asignamos $P(x^{\downarrow G_1}) = P(x^{\downarrow R_1} \mid x^{\downarrow S_1})$
4. Para $i = 2$ hasta $i = m$, hacer:
   - Calcular $P(x^{\downarrow S_i}) = \sum_{G_j \setminus S_i} P(x^{\downarrow G_j})$, donde el grupo G_j es padre de G_i en el árbol.
   - Asignar $P(x^{\downarrow G_i}) = P(x^{\downarrow R_i} \mid x^{\downarrow S_i})\, P(x^{\downarrow S_i})$.

✦ *Cálculo de la probabilidad de cada variable:*

5. Para $i = 1$ hasta $i = n$, hacer:
   - Elegir el grupo maximal G_j de menor tamaño que contenga al nodo X_i.
   - Asignar $P(X_i / E) \propto \sum_{G_j \setminus X_i} P(x^{\downarrow G_j})$.
   - Normalizar los valores obtenidos.

Apliquemos el algoritmo a nuestro ejemplo, suponiendo el árbol de grupos maximales es el representado en Figura 3.9 (que también se obtiene aplicando nuestro algoritmo, tras eligiendo como nodo raíz el grupo {A,T} y romper los empates convenientemente). Supongamos que la evidencia disponible es que el paciente no tiene disnea y que ha dado positivo en la prueba de los rayos X, es decir, $E^* = \{D=\neg d, X=x\}$.

- En la fase de absorción de evidencias las únicas funciones potenciales que necesitamos cambiar son $\psi_5(D,B,E)$ y $\psi_6(X,E)$, ya que son las únicas que se ven afectadas por las evidencias. Las nuevas funciones potenciales son:

  $\psi^*_5(d,B,E) = 0$; $\psi^*_5(\neg d,B,E) = \psi^*_5(\neg d,B,E)=P(\neg d / E,B)$.
  $\psi^*_6(\neg x,E) = 0$; $\psi^*_6(x,E)= \psi_6(x,E)$.

- Empezamos ahora con la *fase ascendente*:

  Para el grupo maximal G_6:

  - $m_6(E) = \sum_E \psi^*_6(X,E)$
  - $P(R_6 / S_6) = P(X / E) = \dfrac{\psi^*_6(X,E)}{m_6(E)}$.
  - Reemplazamos $\psi_2(E,L,T)$ por $\psi^*_2(E,L,T) = \psi_2(E,L,T) \cdot m_6(E)$

  Para el grupo maximal G_5:

  - $m_5(B,E) = \sum_D \psi^*_5(B,E,D)$.
  - $P(R_5 / S_5) = P(D / B,E) = \dfrac{\psi^*_5(X,E)}{m_5(E)}$.
  - Reemplazamos $\psi_3(B,L,E)$ por $\psi^*_3(B,L,E) = 1 \cdot m_5(B,E)$

  Para el grupo maximal G_4:

  - $m_4(L,B) = \sum_S \psi^*_4(S,L,B)$.
  - $P(R_4 / S_4) = P(S / L,B) = \dfrac{\psi^*_4(X,E)}{m_4(E)}$.
  - Reemplazamos de nuevo $\psi_3(B,L,E)$ por $\psi^*_3(B,L,E)= 1 \cdot m_5(B,E) \cdot m_4(L,B)$.

  Para el grupo maximal G_3:

  - $m_3(L,E) = \sum_B \psi^*_3(B,L,E)$.
  - $P(R_3 / S_3) = P(B / L,E) = \dfrac{\psi^*_3(X,E)}{m_3(E)}$.
  - Reemplazamos de nuevo $\psi_2(E,T,L)$ por $\psi^*_2(E,T,L) = \psi_2(E,T,L) \cdot m_6(E) \cdot m_3(L,E)$

Para el grupo maximal G_2:

- $m_2(T) = \sum_{E,L} \psi^*_2(E,L,T)$.
- $P(R_2 / S_2) = P(E,L / T) = \frac{\psi^*_2(X,E)}{m_2(E)}$.
- Reemplazamos $\psi_1(A,T)$ por $\psi^*_1(A,T) = \psi_1(A,T) \cdot m_2(T)$.

Para el grupo maximal G_1:

- $m_1(\varnothing) = \sum_{A,T} \psi^*_1(A,T)$.
- $P(R_1 / S_1) = P(A,T / \varnothing) = \frac{\psi^*_1(X,E)}{m_1(E)}$.

- Tras este último paso empieza la *etapa descendente*:

  En primer lugar, asignamos $P(A,T) = P(R_1 / S_1)$

  Para el grupo maximal G_2:

  - $P(S_2) = P(T) = \sum_A P(A,T)$.
  - $P(E,L,T) = P(R_2 \mid S_2) \cdot P(S_2) = P(E,L / T) \cdot P(T)$.

  Para el grupo maximal G_3:

  - $P(S_3) = P(L,E) = \sum_T P(E,L,T)$.
  - $P(B,L,E) = P(R_3 \mid S_3) \cdot P(S_3) = P(B / L,E) \cdot P(L,E)$.

  Para el grupo maximal G_4:

  - $P(S_4) = P(L,B) = \sum_E P(B,L,E)$.
  - $P(S,L,B) = P(R_4 \mid S_4) \cdot P(S_4) = P(S / L,B) \cdot P(L,B)$.

  Para el grupo maximal G_5:

  - $P(S_5) = P(B,E) = \sum_L P(B,L,E)$.
  - $P(D,B,E) = P(R_5 \mid S_5) \cdot P(S_5) = P(D / B,E) \cdot P(B,E)$.

  Por último, para el grupo maximal G_6:

  - $P(S_6) = P(E) = \sum_{L,T} P(E,L,T)$.
  - $P(X,E) = P(R_6 \mid S_6) \cdot P(S_6) = P(X / E) \cdot P(E)$.

Una vez que tenemos las distribuciones conjuntas de todos los grupos de la red podemos calcular la distribución de probabilidad de cada variable eligiendo el grupo de menor tamaño que la contiene y marginalizando la distribución, según se

detalla en la Tabla 3.5, donde si es necesario habrá que normalizar las probabilidades:

Variable	Grupo	Probabilidad	Variable	Grupo	Probabilidad
A	G_1	$P(A) = \sum_{T} P(A,T)$	B	G_4	$P(B) = \sum_{S,L} P(S,L,B)$
T	G_1	$P(T) = \sum_{A} P(A,T)$	D	G_5	$P(D) = \sum_{B,E} P(D,B,E)$
S	G_4	$P(S) = \sum_{L,B} P(S,L,B)$	X	G_6	$P(X) = \sum_{E} P(X,E)$
L	G_4	$P(L) = \sum_{S,B} P(S,L,B)$	E	G_6	$P(E) = \sum_{X} P(X,E)$

Tabla 3.5 Obtención de las probabilidades de cada nodo.

Para mejorar la eficiencia hemos utilizado *el algoritmo orientado a un objetivo* (Castillo, Gutiérrez et al., 1997), que permite dado el conjunto de evidencias disponibles y el conjunto de nodos cuya probabilidad interesa conocer, identificar un subconjunto de la red en el que están los nodos relevantes para dicha operación y realizar la propagación en dicho subconjunto. En la siguiente sección se presenta dicho algoritmo.

3.3.3 Algoritmo orientado a un objetivo

El objetivo de los algoritmos descritos en la sección anterior es obtener la probabilidad de los nodos de una red una vez que se observa cierto conjunto de evidencias E. Sin embargo, en algunos casos sólo nos interesa cierto conjunto de variables Y, y nuestro objetivo es obtener la función de distribución condicionada de estas variables dada la evidencia observada. En esta situación, algunas de las variables en la red pueden no ser relevantes en los cálculos de las funciones de distribución condicionadas, y por tanto podemos evitar cálculos innecesarios si determinamos este conjunto de variables irrelevantes. La idea es eliminar estas variables del grafo y llevar a cabo la propagación en un subgrafo de menor tamaño que el inicial.

Supongamos por tanto que tenemos una red bayesiana $R = (X, A)$ de la que nos interesa conocer las probabilidades a posteriori de un subconjunto $Y \subset X$ dado un conjunto de evidencias observadas E^*. Las variables del conjunto Y se denominan *variables objetivo.*

Para describir el algoritmo necesitamos definir primero el concepto de d-separación:

Definición

Sea R una red bayesiana, y sean X, Y y Z tres subconjuntos disjuntos de nodos de R. Se dice que el conjunto Z *d-separa* a X e Y si y sólo sí a lo largo de cada camino dirigido desde cada nodo de X a cada nodo de Y hay un nodo intermedio A tal que o bien:

a) A es un nodo en el que las flechas convergen *cabeza con cabeza*, es decir, el camino es (... $U \rightarrow A \leftarrow V$ ...), y ni A ni sus descendientes están en Z, o

b) A no es un nodo en el que las flechas convergen cabeza con cabeza y A está en Z.

En estas condiciones dan a entender que las causas (padres) de cualquier mecanismo causal se vuelven dependientes una vez que el efecto (hijo) común se produce, puesto que un aumento de la creencia en una de ellas significaría un descenso en la creencia de la otra. Este mecanismo de explicación se conoce con el nombre de *explaining away*, que podría traducirse como *descartar causas*.

Para calcular si cierto conjunto Z d-separa a dos conjuntos dados X e Y podemos utilizar el Algoritmo 3.8.

Algoritmo 3.8. D-separación

Entrada: Tres conjuntos X, Y, Z.

Salida: Verdad o falsedad de la afirmación "Z d-separa a X e Y".

1. Identificar el menor subgrafo que contiene a X, Y Z y a sus conjuntos ancestrales.
2. Moralizar el subgrafo obtenido.
3. Si cada camino que existe en el subgrafo entre un nodo de X y un nodo de Y contiene al menos un nodo de Z, entonces la afirmación "Z d-separa a X e Y" es cierta.

Una vez definido el concepto de d-separación y visto el algoritmo para comprobarlo, pasamos a describir el algoritmo para obtener los nodos relevantes:

Algoritmo 3.9. Identificación de nodos relevantes

Entrada: Una red bayesiana, un conjunto de nodos de interés Y y un conjunto de evidencias E.

Salida: El conjunto de nodos R relevantes para el cálculo de la probabilidad a posteriori de las variables de Y a la vista de la evidencia E.

1. Construir un nuevo grafo dirigido R', añadiendo un nodo auxiliar Φ_i y una arista $\Phi_i \rightarrow X_i$ para cada nodo X_i de la red.
2. Identificar el conjunto Φ de nodos auxiliares que no estén d-separados de Y por E en R'.
3. Asignar a R los nodos X_i cuyos nodos auxiliares Φ_i están contenidos en Φ.

Como veremos, este algoritmo nos será de gran utilidad para reducir la complejidad computacional en ciertos casos.

Apliquemos el algoritmo a nuestro ejemplo de la red Asia. Supongamos que tenemos un paciente que sabemos que está enfermo (E=e), y que nos interesa conocer la probabilidad de que padezca bronquitis (B). En este caso, E^*={E=e} y el conjunto de nodos de interés es Y ={B}. La aplicación del algoritmo produciría la siguiente red (con los nodos ficticios añadidos):

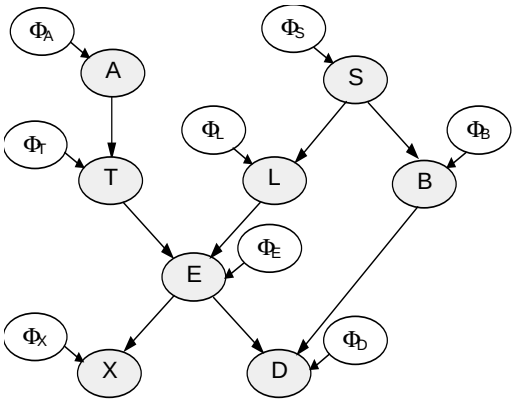


Figura 3.10 Red Asia con nodos ficticios.

Para esta red vemos que, por ejemplo, el nodo Φ_S no está d-separado de *B* por *E*, ya que en el grafo ancestral moralizado asociado a dichos nodos (que se muestra en la Figura 3.11) existe un camino entre los nodos Φ_S y B que no incluye al nodo E.

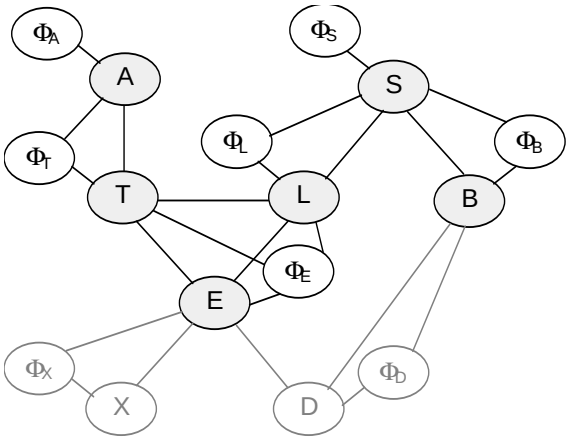


Figura 3.11 Grafo ancestral moralizado correspondiente a los nodos Φ_S *,B y E.*

Construyendo los grafos ancestrales moralizados correspondientes, puede observarse que tampoco están d-separados los nodos Φ_A, Φ_T y Φ_L. Sin embargo, el nodo Φ_D sí está d-separado del nodo *B* por *E*, ya que en el grafo ancestral moralizado correspondiente existe un camino de Φ_D a *B* que no incluye al nodo *E*. Por tanto, en este caso los nodos relevantes para el cálculo que queremos realizar son {*A*,*T*,*E*,*L*,*S*,*B*}, y el grafo reducido para calcular P(*B* / *E*) se muestra en la Figura 3.12:

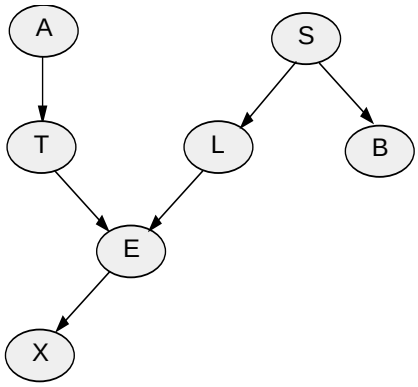


Figura 3.12 Grafo reducido para calcular P(*B* / *E*).

En la Figura 3.13 se muestran los resultados obtenidos al propagar la evidencia[14] (la persona padece la enfermedad *E*). Como se puede ver, la probabilidad de que padezca bronquitis es idéntica en ambos casos es (0.432).

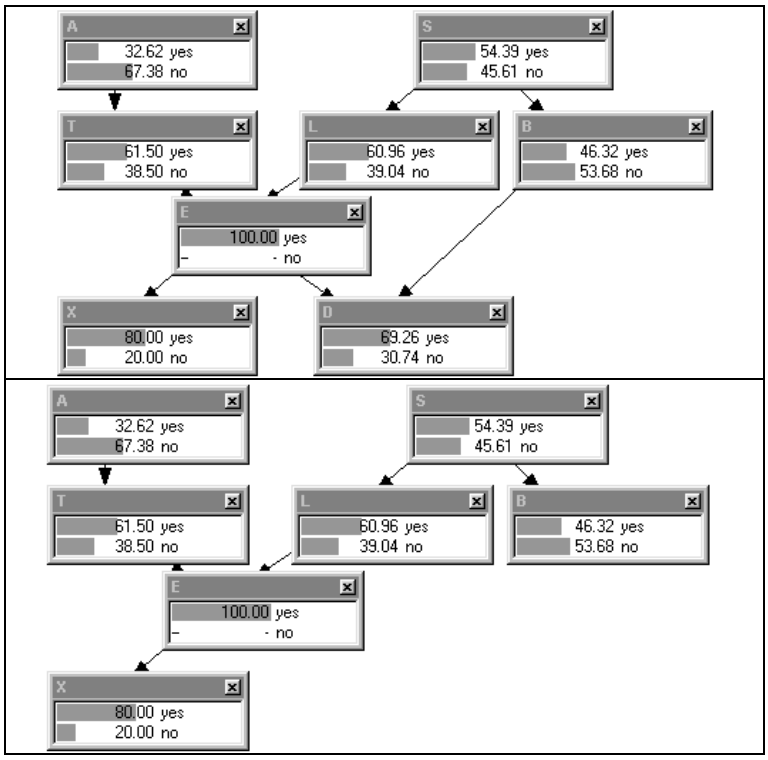


Figura 3.13 Cálculo de P(B / E=1) en la estructura completa y en la red simplificada.

Si bien en este ejemplo de prueba la reducción en el número de nodos no ha sido muy grande, dependiendo de la estructura de la red se alcanzan reducciones significativas no sólo en el número de nodos sino en el tipo de estructura, pudiéndose pasar incluso de un grafo acíclico dirigido a estructuras más simples como árboles o poliárboles, mejorando de esta forma la eficiencia de la propagación.

3.4 Conclusiones

En este capítulo hemos presentado los conceptos básicos en redes bayesianas, junto con los algoritmos que hemos utilizado en la implementación de nuestro sistema. Como se ha destacado a lo largo de la presentación, en la implementación se han

[14] La Figura 3.13 y los cálculos que en ella aparecen han sido realizados con HUGIN, versión de demostración 5.1.

hecho algunas modificaciones de los algoritmos, buscando siempre una mejora en la eficiencia. En concreto, dichas modificaciones consisten en un nuevo heurístico para la determinación del orden de eliminación de las variables en el proceso de triangulación y también una nueva ordenación de los grupos maximales en la construcción del árbol de grupos.