Universidad Nacional Autónoma de México

Facultad de Ciencias

Redes bayesianas aplicadas a la detección de diabetes mellitus tipo 2.

T E S I S.

Que para obtener el título de:

Actuario

Presenta:

Ricardo Cruz Sánchez

Tutor:

Ricardo Ramirez Aldana

Ciudad Universitaria, Cd. Mx.,2020

1. **Datos del alumno**
   Cruz
   Sánchez
   Ricardo
   75 75 13 86
   Universidad Nacional Autónoma de México
   Facultad de Ciencias
   Actuaría
   311045305

2. **Datos del tutor**
   Dr.
   Ricardo
   Ramírez
   Aldana

3. **Datos del sinodal 1**
   M. en C.
   Antonio
   Soriano
   Flores

4. **Datos del sinodal 2**
   Act.
   Miguel Ángel
   Chong
   Rodríguez

5. **Datos del sinodal 3**
   Act.
   Yolanda
   Martínez
   Guerrero

6. **Datos del sinodal 4**
   Act.
   Sergio Daniel
   Raya
   Rios

7. **Datos del trabajo escrito**
   Redes bayesianas aplicadas a la detección de diabetes mellitus tipo 2
   100 páginas
   2020

Índice

Agradecimientos

En primer lugar, quisiera agradecer a mis padres, ya que sin ellos no hubiese podido lograr desarrollar esta tesis para la defensa de grado, pues me han apoyado siempre, no solo en lo académico, sino en cada aspecto de mi vida, este logro en mi vida es, en su totalidad, de ellos. Por proporcionarme todo lo que esta a su alcance todos los días.

De igual forma a mis hermanos y abuelos que me han brindado sus mejores consejos y compañía para poder seguir superándome día a día.

Posteriormente, a los amigos que he formado hasta el día de hoy, en particular, a mis compañeros de universidad, Sergio y Dalina. Gracias a ellos pude aprovechar de la mejor manera mi estancia en la faculta de ciencias.

A mis profesores, comenzando por Antonio Soriano, Yolanda Mártinez y Edgar Díaz. A partir de sus clases nació mi gusto por la estadística y mi carrera cobró un sentido totalmente nuevo.

A mi asesor y profesor, el Dr. Ricardo Ramírez, por el apoyo para sacar adelante este proyecto, además de las clases que me brindó las cuales disfruté y hoy en día me siguen siendo útiles.

A mis compañeros de trabajo que encontré en SAS, por ayudarme a crecer en el ámbito profesional y personal, además de la amistad que generamos.

Al profesor Miguel Chong, por ayudarme en la revisión de mi tesis y ofrecerme la oportunidad de dar clases como ayudante, lo cual resulto en una gran experiencia.

A todas las personas que conocí en Monterrey en una nueva etapa de mi vida y me motivaron a concluir este proyecto.

A la universidad y a todo el personal que hizo posible que tuviese la oportunidad de cursar una carrera y concluirla.

1. Introducción.

A lo largo de la historia, una de las preocupaciones de la humanidad ha sido determinar la ocurrencia de eventos futuros, ya sea para tomar ventaja de ellos o simplemente implementar acciones para reducir el daño que pudiese ocasionar como consecuencia. Dado que no todos los eventos pueden predecirse a través de una fórmula determinista, se recurre a medidas que puedan expresar la incertidumbre que se tiene de los eventos desconocidos, como lo es la probabilidad. Esta medida asocia un número entre 0 y 1, el cual representa la certeza en que un evento específico suceda.

Los eventos de interés pueden depender de diversos factores, es decir, la ocurrencia de otros eventos puede determinar si el evento de interés es más o menos probable. Por ejemplo, si nos interesa saber si el día de hoy lloverá, un evento que podría aumentar o disminuir la probabilidad asociada es conocer si es un día nublado, en este caso, solo dos eventos son considerados para medir la probabilidad. Sin embargo, existen problemas en los cuales se deben considerar múltiples eventos, tal es el caso cuando se quiere conocer la presencia de una enfermedad, se deben considerar los hábitos y características del paciente, además de sus síntomas, o en caso de querer determinar si se le debería otorgar un crédito a un cliente, se debe considerar las características socioeconómicas y financieras del cliente.

En la actualidad, existen diversos modelos para calcular probabilidades asociadas a problemas específicos, la elección del modelo dependerá en el contexto del problema. Si este contexto incluye un gran número de variables (eventos), entonces las probabilidades asociadas son difíciles de calcular, por lo tanto, puede recurrirse a los modelos gráficos probabilísticos, los cuales simplifican la asignación de probabilidades, además de mostrar la influencia que existe entre variables.

1.1. Sistemas expertos y modelos gráficos probabilísticos.

En la década de los 60's, se originaron los sistemas expertos como un área de estudio de la inteligencia artificial. Su objetivo es replicar, mediante un sistema informático, las acciones que un humano pudiese hacer utilizando su conocimiento en cierto tema en el que esté especializado. Por lo tanto, la creación de un sistema experto es un trabajo multidisciplinario, ya que implica tener conocimientos sólidos en el tema a desarrollar, en sistemas computacionales y en algunas ocasiones se recurre a la probabilidad y teoría de gráficas.

Para la construcción de un sistema experto se deben considerar dos elementos básicos a implementar, el primero de ellos es la base de conocimiento, que corresponde a las reglas de negocio, relaciones entre variables y probabilidades, que solo un especialista en el tema podría transmitir. El segundo elemento es el motor de inferencia, el cual se constituye de los algoritmos encargados de aplicar el razonamiento especificado con base en el conocimiento de los datos que son proporcionados al sistema. En otras palabras, el motor de inferencia se encarga de aplicar las reglas de negocio para obtener conclusiones sobre los datos disponibles. La calidad del sistema experto dependerá de la correcta definición de la base de conocimiento y motor de inferencia.

Los primeros sistemas expertos pretendían resolver problemas de carácter determinístico, por lo que se basaban en una serie de preguntas o reglas lógicas. Al conocer el resultado podía realizarse otra pregunta o regla, continuando de manera sucesiva hasta llegar a la resolución que un experto pudiera tomar. Las desventajas de este tipo de sistemas se presentan cuando no se cuenta con los datos requeridos por el sistema, debido a que siguen un flujo, se necesita tener los datos referentes a una regla para continuar con la siguiente. Las posibles soluciones son esperar a tener el valor del dato o asignar un valor por defecto, aunque pudiese ser diferente al real, lo cual tal vez llevaría a conclusiones que no necesariamente reflejen una solución al problema. Gracias a esto, surge la necesidad de involucrar una medida de incertidumbre asociada a las variables del sistema. Las opciones para esta medida fueron los factores de certeza, funciones de creencia, la lógica difusa, la probabilidad, entre otros.

Si bien la probabilidad pareciera ser la medida más adecuada para representar incertidumbre, esta no era consistente con las reglas lógicas, pues existen axiomas lógicos que pueden violar los axiomas probabilísticos. Aunado a esto, la asignación de una función de probabilidad a un conjunto considerable de variables no es una tarea sencilla, ya que el número de valores que se deben especificar para cada combinación de variables crece de manera exponencial de acuerdo al número de variables y rangos de las mismas.

Posteriormente, surgieron los modelos gráficos probabilísticos, los cuales son sistemas expertos que se basan en la teoría de grafos y probabilidad. Estos modelos logran representar de manera compacta a una probabilidad conjunta por medio de un grafo, simplificando la especificación de parámetros mediante las independencias marginales y condicionales entre variables. El aprovechamiento de las independencias mencionadas posicionaron a este tipo de modelos como uno de los más utilizados por la inteligencia artificial, además del crecimiento tecnológico que impulsó el desarrollo de algoritmos más sofisticados para la propagación de la incertidumbre de las variables en el modelo.

En general, los sistemas expertos proveen una forma eficaz de obtener conclusiones similares a las de un experto en un tema en específico. Además, cuando la demanda de especialistas es muy elevada como para satisfacerla, los sistemas expertos ofrecen la solución al automatizar el proceso realizado por el experto. Sin embargo, el costo en tiempo y recursos puede ser elevado, ya que la elaboración de un sistema experto involucra la participación de múltiples elementos humanos y tecnológicos, en los cuales no se puede escatimar si se desea un sistema con la mayor precisión posible.

El proceso para diseñar un sistema experto involucra a tres entes esencialmente, el experto en el tema, el experto en la implementación del sistema y el usuario final. Estas son las personalidades primordiales al desarrollar el sistema, sin embargo, se pueden incorporar más personajes si es que así lo requiere el proyecto. Las actividades que desempeñan varían de acuerdo al rol y la fase en la que se encuentre el proyecto, a continuación se detallan dichas fases en el proceso de construcción de un sistema experto:

1. **Definición del problema:** Se comienza conociendo el problema a solucionar. En esta fase se determina qué se espera del sistema experto por parte de los usuarios finales, el experto en el tema explica de forma general la manera en que se solucionan los problemas que se esperan resolver y el experto en sistemas indica el o los tipos de modelos que se implementarán, así como el software y hardware que utilizará.

2. **Transferencia del conocimiento:** A través de entrevistas y ejemplos prácticos para resolver el problema, el experto en el tema mostrará las reglas del negocio a quien implementará el sistema experto. El entendimiento del negocio será plasmado en el modelo generado en la siguiente fase.

3. **Implementación de un sistema experto prototipo:** El experto en sistemas crea un modelo de acuerdo a lo aprendido en la fase anterior, lo ideal es haber detectado todas las reglas del negocio, en caso contrario se puede regresar a la fase 2 para resolver dudas del negocio.

4. **Pruebas del prototipo:** Se evalúa la calidad del modelo ejecutándolo en múltiples casos y se mide su desempeño. Si el sistema experto no cumple con los requerimientos o puede presentar mejoras, entonces la fase 3 o la fase 2 deben volver a ser realizadas según sean las circunstancias.

5. **Puesta en producción:** En esta fase, el sistema experto se pone a disposición del usuario final.

6. **Retroalimentación y mantenimiento:** El usuario final comunica su experiencia al utilizar el modelo, detallando las fallas que pudiesen presentarse. En caso de necesitarlo, el experto en sistemas deberá realizar las modificaciones pertinentes al modelo.

1.2. Tipos de modelos gráficos probabilísiticos y sus aplicaciones.

Existen diversos tipos de modelos gráficos probabilísticos (PGM por sus siglas en inglés), los cuales varían en los problemas que son capaces de resolver, es decir, existen ventajas y desventajas entre diferentes tipos de PGM's y usualmente esto depende de la estructura del grafo (dirigido, no dirigido o parcialmente dirigido) y supuestos que se realicen sobre el modelo. A continuación se presenta una lista con los PGM's más comunes:

- **Clasificador naive Bayes:** Modelo que a través de un grafo dirigido codifica la función de distribución conjunta de una variable de clasificación y distintas variables que representan múltiples características del objeto de estudio. Este modelo supone independencias que comúnmente no se presentan en los problemas reales.

- **Modelo oculto de Markov:** Determina como un proceso de Markov evoluciona en el tiempo por medio de un grafo dirigido. Se basa en el supuesto de la propiedad de Markov y el estado del proceso no siempre es observable. Es un tipo de red bayesiana dinámica.

- **Campo aleatorio de Markov:** Determina la interacción entre un conjunto de variables que satisfacen la propiedad de Markov en un grafo no dirigido. El estado de cada

variable es independiente al resto de las variables si se conoce el estado de sus vecinos en el grafo.

- **Red Bayesiana:** Grafo acíclico dirigido asociado a un conjunto de variables, este grafo representa de manera compacta la distribución conjunta de las variables y las relaciones de independencia entre variables. El estado de una variable es independiente a sus no descendientes en el grafo si se conoce el estado de sus padres en el grafo.

- **Red Bayesiana dinámica:** Es una red bayesiana cuyo estado evoluciona a lo largo del tiempo.

- **Modelos de gráficas de cadena:** Corresponden a un modelo híbrido entre redes bayesianas y redes de Markov (caso particular de un campo aleatorio de Markov). Por lo tanto, el grafo asociado al modelo es parcialmente dirigido.

- **Modelos de decisión:** Grafos en los cuales se incorpora la teoría de decisiones, el objetivo principal de estos modelos es maximizar una función de utilidad y determinar el conjunto de acciones a tomar (decisión).

Actualmente la inteligencia artificial ha desarrollado una rama muy prolifera, a saber, el aprendizaje automático o aprendizaje de máquina (machine learning o ML por sus siglas en inglés). Esta rama se enfoca en crear sistemas que adquieran conocimiento por sí mismos para ser capaces de detectar comportamientos o patrones, esto no necesariamente está ligado a tener la guía de un experto a diferencia de los sistemas expertos. Sin embargo, algunos sistemas han contribuido a crear modelos en ML, por ejemplo, las redes neuronales probabilísticas. Incluso algunos sistemas expertos son considerados como modelos en el campo de ML, tal es el caso de las redes bayesianas.

La forma en que los modelos en ML generan su propio conocimiento es a través de grandes cantidades de datos. Hoy en día el número de datos que genera un solo individuo es impresionante, es por eso que los modelos en ML han ganado tanta popularidad. Sin embargo, esto no significa que reemplacen a los sistemas expertos, puesto que la colaboración de un experto en ocasiones es indispensable y no puede ser substituida por algoritmos, por más sofisticados que sean. La elección depende en el contexto y costos que impliquen. Los modelos de ML pueden verse como sistemas expertos, en los cuales los datos pueden ser considerados como el rol del experto en un sistema experto. Estas dos ramas de la inteligencia artificial tienen múltiples similitudes que ayudan a progresar tanto a una como a la otra.

Los PGM's tienen diversas aplicaciones en diferentes industrias, ejemplos de estas aplicaciones son:

- Reconocimiento de patrones.

- Clasificación de documentos y páginas web.

- Diagnóstico médico.

- Genética.

- Visión de computadora.
- Reconocimiento de voz y gestos.
- Modelos físicos y químicos.
- Robótica.
- Diagnóstico de fallas.
- Clasificación de textos.

1.3. Objetivos generales de este trabajo.

El presente trabajo aborda de manera particular el PGM conocido como red bayesiana. Este modelo está caracterizado por ser un grafo acíclico dirigido (*directed acyclic graph* o DAG por sus siglas en inglés) asociado a una probabilidad conjunta sobre un conjunto de variables $\mathbf{X}$. Las independencias condicionales y marginales obtenidas a partir del grafo permiten disminuir el número de parámetros requeridos en la especificación de la función de probabilidad conjunta, lo cual se logra utilizando la regla de la cadena en grafos dirigídos y el teorema de Bayes. Gracias a este último se conocen como redes bayesianas (RB).

El principal uso de las redes bayesianas es el diagnóstico médico o de fallas, además provee una estructura en la cual se pueden realizar consultas sobre cualquier variable cuando no se posee información sobre un subconjunto de variables en $\mathbf{X}$.

La construcción de una RB se puede reducir a 3 etapas: representación, inferencia y aprendizaje. En este trabajo se presenta la teoría básica antes de iniciar el proceso de construcción de la RB, el capítulo 2 ofrece la teoría de grafos y de probabilidad necesaria para la comprensión de los temas descritos en capítulos posteriores. Si bien las RB son DAG, también se incluye teoría sobre grafos no dirigidos. El capítulo 3 representa la parte esencial de la construcción de un RB, aquí se describe como asociar un grafo a una función de probabilidad conjunta aprovechando las independencias de las variables y se presenta el criterio de separación dirigida. En la cuarta sección se exponen diversos algoritmos correspondientes al motor de inferencia de nuestro sistema, estos se dividen en aquellos que son exactos y los que son aproximados. El quinto apartado muestra los métodos de aprendizaje de una RB, segmentándolos en aprendizaje paramétrico, que consta de estimar las tablas de probabilidad asociadas a las variables, y el aprendizaje estructural, asociado a la independencia de variables y funciones de probabilidad. Por último, el capítulo 6 desarrolla un caso de estudio en el cual se aplica todo lo expuesto en los capítulos anteriores. El caso de estudio es referente a la detección de diabetes en mujeres y pretende evidenciar la necesidad de las redes bayesianas en problemas reales, para esto, se creará una RB sobre un conjunto de variables relacionadas al padecimiento para una población particular y se utilizará el modelo como clasificador para exhibir las interacciones que las variables tienen sobre la presencia o ausencia de la enfermedad.

El desarrollo de la teoría será acompañado de ejemplos con ayuda del software R. Aunque hoy se pueden encontrar diversos softwares especializados en redes, se eligió R por su practicidad y por ser un software libre en el cual podemos acceder a los códigos de funciones y modificarlos de acuerdo a nuestras necesidades. Los códigos utilizados serán incluidos en el apéndice de este trabajo.

Este trabajo abordará las redes bayesianas desde el punto de vista discreto, es decir, todas las variables en $\mathbf{X}$ solo pueden tomar valores en un conjunto finito, a este tipo de RB se les conoce como redes bayesianas multinomiales. A pesar de esto, parte de la teoría presentada aquí se puede aplicar a modelos con variables continuas o modelos con variables continuas y discretas. En la bibliografía se incluyen libros en los que se detallan las redes distintas a las multinomiales.

2. Fundamentos.

El desarrollo y comprensión de las RB depende de conceptos pertenecientes a la teoría de probabilidad y teoría de grafos, la primera para modelar la incertidumbre asociada a las variables involucradas y la segunda para representar de manera compacta a la función de probabilidad asignada. A continuación se presentan solo los conceptos básicos de dichas teorías.

2.1. Teoría de probabilidad.

La probabilidad (P) es una medida asociada a la ocurrencia de un evento de carácter aleatorio, este evento es un elemento de (B), el conjunto de todos los subconjuntos formados por uniones e intersecciones numerables de elementos de (Ω) que representa el espacio de todos los resultados posibles de un experimento aleatorio. La terna (Ω,B,P) se conoce como espacio de probabilidad.

Una variable a aleatoria (X) es una función que asocia un valor numérico o categórico a cualquier resultado en el espacio muestral. A partir de esta definición, surgen los vectores aleatorios, que son vectores cuyas entradas corresponden a variables aleatorias. Los vectores aleatorios se denotan como: $\mathbf{X} = (X_1, \ldots, X_n)$.

Existen dos tipos de variables aleatorias de acuerdo a los valores que pueden tomar. Se dice que una variable es *discreta* si el conjunto de valores posibles es un conjunto finito numerable y se dice que es una variable aleatoria *continua* si los valores posibles se encuentran en un conjunto no numerable. Por ejemplo, la variable aleatoria que determina el resultado de un lanzamiento de una moneda es una variable discreta, puesto que solo existen dos posibles valores, mientras que una variable aleatoria asociada al tiempo de recuperación de un paciente es continua. Al conjunto de valores que una variable puede tomar se le denomina *rango* de la variable ($Ran(X)$) y para el caso de una variable aleatoria discreta, la cardinalidad de este conjunto ($\#Ran(X)$) se conoce como *niveles* o *categorías*.

Las variables aleatorias representan los eventos de interés, de los cuales no se conoce su valor con certeza, es por eso que se les puede asociar una *función de densidad de probabilidad* ($P(X)$). La función de densidad asigna un número entre 0 y 1 a cualquier valor x que pueda tomar la variable aleatoria. Este número entre 0 y 1 representa la probabilidad de que la variable aleatoria tome dicho valor, lo cual se denota como $P(X = x) \in [0, 1]$. La notación se puede simplificar a $P(x)$. A las funciones de densidad de probabilidad sobre una sola variable aleatoria se les conoce como *función de distribución marginal* o simplemente *distribución marginal*. Este concepto se puede extender a vectores aleatorios, dando origen a las *distribuciones conjuntas* ($P(\mathbf{X})$), al igual que las distribuciones marginales asignan un valor entre 0 y 1 a la posible combinación de valores de las variables de un vector aleatorio. Continuando con la misma notación tenemos:

$$P(\mathbf{x}) = P(\mathbf{X} = \mathbf{x}) = P(X_1 = x_1, \ldots, X_n = x_n) \in [0, 1]$$

En el presente documento se utilizará el término función de distribución sin detallarse si

se trata de la distribución marginal o conjunta, a menos que lo amerite. Para distinguir a cual de las dos se refiere, basta con considerar si esta función se aplica sobre un conjunto de variables o a solo una variable.

Adicionalmente, todas las funciones de distribución sobre variables aleatorias discretas, ya sea marginales o conjuntas, satisfacen que la suma de las probabilidades de cada posible valor (de las variables o del vector) es igual a 1, es decir:

$$\sum_{x} P(x) = 1$$

Una vez que se conoce la distribución conjunta de una colección de variables (discretas), se puede obtener la distribución marginal de cada una de ellas, basta con determinar el valor que se desea en la función marginal, y se sumaran todas las probabilidades que contengan dicha asignación, a este proceso se le conoce como *marginalización*:

$$P(x_i) = \sum_{\mathbf{x} \setminus x_i} P(\mathbf{x}) = \sum_{\mathbf{x} \setminus x_i} P(x_1 \dots x_n) \qquad \forall i \in 1 : n$$

Como se mencionó, las funciones de distribución calculan la probabilidad de que una variable o vector aleatorio sean iguales a cierta asignación de valores que aun no se conocen y por lo tanto son inciertos. En el caso de un vector aleatorio, se podría conocer el valor de algunas de sus componentes, lo cual implicaría que solo las variables que no se han observado son inciertas y que la probabilidad de observar la asignación de las variables restantes se modifique. La probabilidad de un subconjunto de variables aleatorias, una vez que ya se conoce el valor de otro subconjunto disjunto, se conoce como *probabilidad condicional*. La probabilidad de $\mathbf{X}$ dado $\mathbf{Y}$ se denota como $P(\mathbf{X} \mid \mathbf{Y})$ y se define de la siguiente manera:

$$P(\mathbf{X} \mid \mathbf{Y}) = \frac{P(\mathbf{X} \cap \mathbf{Y})}{P(\mathbf{Y})} = \frac{P(\mathbf{X}, \mathbf{Y})}{P(\mathbf{Y})}, \tag{2.1}$$

lo cual está definido siempre que $P(\mathbf{Y}) > 0$. A partir de la ecuación anterior se puede obtener una expresión para la distribución conjunta de dos conjuntos de variables que dependa de una distribución conjunta y una distribución condicional.

$$P(\mathbf{X} \cap \mathbf{Y}) = P(\mathbf{X}, \mathbf{Y}) = P(\mathbf{X} \mid \mathbf{Y}) P(\mathbf{Y}) \tag{2.2}$$

A la ecuación (2.2) se le conoce como la *regla de la cadena* y se puede extender para la distribución conjunta de n subconjuntos de variables, esto se logra aplicando $n - 1$ veces la igualdad de la ecuación (2.2).

$$P(X_1, \dots X_n) = P(X_1) P(X_2 \mid X_1) \dots P(X_n \mid X_1 \dots X_{n-1}) \tag{2.3}$$

Se puede decir que los elementos de una distribución de probabilidad son conmutativos, es decir, $P(\mathbf{X}, \mathbf{Y}) = P(\mathbf{Y}, \mathbf{X})$, pues ambas representaciones indican la probabilidad asociada a una asignación de valores sobre una colección de variables. Aplicando esta idea a la ecuación (2.2) se obtiene:

$$P(\mathbf{X} \mid \mathbf{Y}) P(\mathbf{Y}) = P(\mathbf{X}, \mathbf{Y}) = P(\mathbf{Y}, \mathbf{X}) = P(\mathbf{Y} \mid \mathbf{X}) P(\mathbf{X}) \tag{2.4}$$

Sustituyendo la igualdad (2.4) en la ecuación (2.1) se obtiene un caso particular de lo que se conoce como el *teorema de Bayes*, logrando especificar la distribución condicional a través de otra distribución condicional y otras dos distribuciones de probabilidad.

$$P(\mathbf{X} \mid \mathbf{Y}) = \frac{P(\mathbf{X}, \mathbf{Y})}{P(\mathbf{Y})} = \frac{P(\mathbf{Y} \mid \mathbf{X})P(\mathbf{X})}{P(\mathbf{Y})} \tag{2.5}$$

Una propiedad importante es la independencia y la independencia condicional que existe entre variables aleatorias. La independencia entre dos variables indica que no existe relación de una variable sobre otra. Formalmente se dice que dos conjuntos de variables son independientes ($\mathbf{X} \perp \mathbf{Y}$) si:

$$P(\mathbf{X}, \mathbf{Y}) = P(\mathbf{X})P(\mathbf{Y}) \qquad \forall \mathbf{x} \in Ran(\mathbf{X}), \mathbf{y} \in Ran(\mathbf{Y}) \tag{2.6}$$

Intuitivamente, si no existe relación entre las variables, entonces el valor de una no puede influir en el valor de la otra, esto se demuestra aplicando la ecuación (2.6) en la ecuación (2.1). Si dos variables son independientes, entonces la probabilidad condicional es igual a la distribución marginal o conjunta que no considera a las variables sobre las cuales se está condicionando.

$$P(\mathbf{X} \mid \mathbf{Y}) = \frac{P(\mathbf{X}, \mathbf{Y})}{P(\mathbf{Y})} = \frac{P(\mathbf{X})P(\mathbf{Y})}{P(\mathbf{Y})} = P(\mathbf{X}) \tag{2.7}$$

El concepto de independencia condicional es similar, sucede si no existe dependencia entre dos conjuntos de variables siempre que se conozca el valor de un tercer conjunto de variables. Dos conjuntos de variables son independientes dado un tercer conjunto ($(\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z})$) si:

$$P(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z}) = P(\mathbf{X} \mid \mathbf{Z})P(\mathbf{Y} \mid \mathbf{Z}) \qquad \forall \mathbf{x} \in Ran(\mathbf{X}), \mathbf{y} \in Ran(\mathbf{Y}), \mathbf{z} \in Ran(\mathbf{Z}) \tag{2.8}$$

Si existe independencia condicional entre dos conjuntos de variables, entonces la ecuación (2.8) implica que:

$$P(\mathbf{X} \mid \mathbf{Y}, \mathbf{Z}) = P(\mathbf{X} \mid \mathbf{Z}) \tag{2.9}$$

Las ecuaciones (2.6) y (2.7) son equivalentes a las ecuaciones (2.8) y (2.9) respectivamente cuando $\mathbf{Z}$ es el conjunto vacío.

A continuación se enuncian las propiedades de la independencia condicional, las cuales jugaran un papel fundamental en temas subsecuentes:

1. **Simetría:** Sean **X,Y,Z** tres conjuntos de variables entonces:

$$(\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z}) \Leftrightarrow (\mathbf{Y} \perp \mathbf{X} \mid \mathbf{Z}) \tag{2.10}$$

2. **Descomposición:** Sean **X,Y,Z,W** cuatro conjuntos de variables entonces:

$$(\mathbf{X} \perp \mathbf{Y},\mathbf{W} \mid \mathbf{Z}) \Rightarrow (\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z}) \tag{2.11}$$

3. **Unión débil:** Sean **X,Y,Z,W** cuatro conjuntos de variables entonces:

$$(\mathbf{X} \perp \mathbf{Y},\mathbf{W} \mid \mathbf{Z}) \Rightarrow (\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z},\mathbf{W}) \tag{2.12}$$

4. **Contracción:** Sean **X,Y,Z,W** cuatro conjuntos de variables entonces:

$$(\mathbf{X} \perp \mathbf{W} \mid \mathbf{Z},\mathbf{Y}) \cap (\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z}) \Rightarrow (\mathbf{X} \perp \mathbf{Y},\mathbf{W} \mid \mathbf{Z}) \tag{2.13}$$

5. **Intersección:** Sean **X,Y,Z,W** cuatro conjuntos de variables entonces para una distribución positiva:

$$(\mathbf{X} \perp \mathbf{W} \mid \mathbf{Z},\mathbf{Y}) \cap (\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z},\mathbf{W}) \Rightarrow (\mathbf{X} \perp \mathbf{Y},\mathbf{W} \mid \mathbf{Z}) \tag{2.14}$$

2.2. Teoría de grafos.

Un grafo ($\mathcal{G}$) es una pareja de conjuntos, el primero de ellos denota un conjunto de vértices o nodos (V) y el segundo es el conjunto de aristas o arcos (A). Un grafo $\mathcal{G} = (V, A)$ representa las relaciones existentes entre dos objetos por medio de los nodos y aristas, los nodos están vinculados a los objetos y una arista entre dos nodos indica que ambos están relacionados. Gráficamente, los nodos son representados por círculos y las aristas por líneas en caso de ser no dirigidas o flechas en caso de ser dirigidas.

Los grafos se pueden clasificar de acuerdo al tipo de aristas que contenga. Los grafos más importantes para las redes bayesianas son aquellos que solo contienen aristas no dirigidas o solo aristas dirigidas, a estos grafos se les conoce como grafos no dirigidos y grafos dirigidos, respectivamente. La figura 1 ejemplifica los dos tipos de grafos mencionados, ambas representaciones están conformadas por el conjunto de nodos $V = \{a, b, c, d\}$.

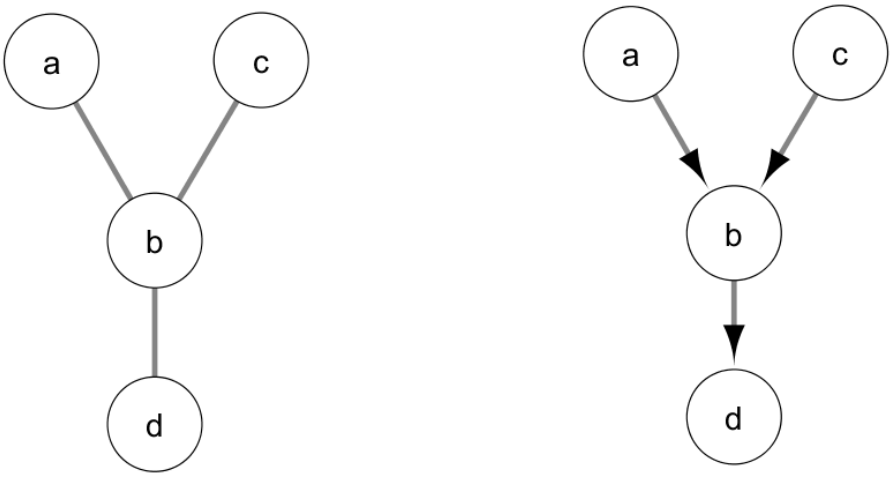


Figura 1: (izq.) Grafo no dirigido, (der.) Grafo dirigido

Sea $V = \{V_1, \cdots, V_n\}$ el conjunto de vértices, se dice que V_j es adyacente a V_i si existe un arco dirigido que comience en V_i y termine en V_j ($V_i \rightarrow V_j$) ó existe un arco no dirigido entre V_i y $V_j (V_i - V_j)$. En el caso de aristas dirigidas, si $V_i \rightarrow V_j$ entonces se dice que V_i es padre de V_j y V_j es hijo de V_i. Para el caso de aristas no dirigidas, si $V_i - V_j$ entonces V_i y V_j son vecinos.

El conjunto de todos los padres de un nodo V_i se denota como Pa_{V_i} o $Pa(V_i)$. El conjunto de los hijos del mismo nodo es Ch_{V_i} o $Ch_(V_i)$. Para denotar el conjunto de los vecinos de un nodo V_j se utiliza Nb_{V_j}. En un grafo dirigido la familia de un nodo V_i se define como $Fam_{V_i} = V_i \cup Pa_{V_i}$

Un camino entre los nodos V_i y $V_j \in V$ se define como la sucesión de nodos $(V_1, \cdots, V_n)$ en la que se satisface que V_{k+1} es adyacente a V_k $\forall k \in 1 : n-1$, además de que $V_i = V_1$ y $V_j = V_n$. Si se cumple que $V_i = V_j$ entonces se denomina como camino cerrado.

Si un camino contiene solo aristas dirigidas se denomina camino dirigido, si además es cerrado entonces se considera como un ciclo. Un camino cerrado en donde al menos uno de sus aristas no es dirigida se le conoce como bucle.

En general, un camino es no dirigido si alguna de las aristas es no dirigida. En futuras referencias, se entenderá como *camino no dirigido* a aquellos caminos en los cuales se ignora la dirección de las aristas de todo el camino, es por esto que se podrá hablar de caminos no dirigidos dentro de grafos dirigidos.

Uno de los conceptos más relevantes en redes bayesianas, son los grafos dirigidos acíclicos (DAG), este tipo de grafo corresponden a aquellos que son dirigidos y además no existen ciclos.

En un grafo dirigido se dice que V_j es ancestro de V_i si existe al menos un camino dirigido que comience en V_j y termine en V_i. El conjunto de todos los ancestros de un nodo se representa por $Ancestros_{V_i}$. De forma similar, si existe un camino dirigido que comience en V_i y termine en V_k, entonces V_k es un descendiente de V_i y el conjunto de todos los descendientes se denota por $Descendientes_{V_i}$.

Los conjuntos anteriores pueden definir otros conjuntos importantes para el desarrollo de las redes bayesianas, el primero de ellos es conocido como los no descendientes de V_i, el cual está definido como $NoDescendientes_{V_i} = V \setminus Descendientes_{V_i}$. El segundo se conoce como *conjunto ancestral*, se constituye a partir de un conjunto de nodos W y todos los ancestros de cada nodo en el conjunto, es decir, $Ancestral(W) = W \cup Ancestros(w_1) \cup \ldots \cup Ancestros(w_n)$ con $W = (w_1, \ldots w_n) \subseteq V$

A partir de un subconjunto de nodos $U \subset V$ en un grafo $\mathcal{G}$ se puede inducir otro grafo $\mathcal{G}'$ que corresponda al subconjunto de nodos U y al subconjunto de aristas A' que contiene solo las aristas que conectan en $\mathcal{G}$ a dos nodos pertenecientes al subconjunto U. A $\mathcal{G}'$ se le conoce como el subgrafo inducido por V'. En la figura 2 se muestra el grafo G con vértices $V = \{a, b, c, d\}$ y el subgrafo inducido por los nodos $\{a, c, d\}$.

Todo grafo dirigido $\mathcal{G}$ tiene asociado un grafo no dirigido $\mathcal{G}'$ que consiste en el mismo conjunto de nodos V y el conjunto de aristas A' está definido por aquellas aristas $V_i - V_j$ siempre y cuando $V_i \to V_j$ o $V_j \to V_i$ en $\mathcal{G}$. Es decir, $\mathcal{G}'$ corresponde al grafo que ignora la dirección de las aristas de $\mathcal{G}$. Por ejemplo, en la figura 1 se muestra un grafo dirigido y su grafo no dirigido asociado.

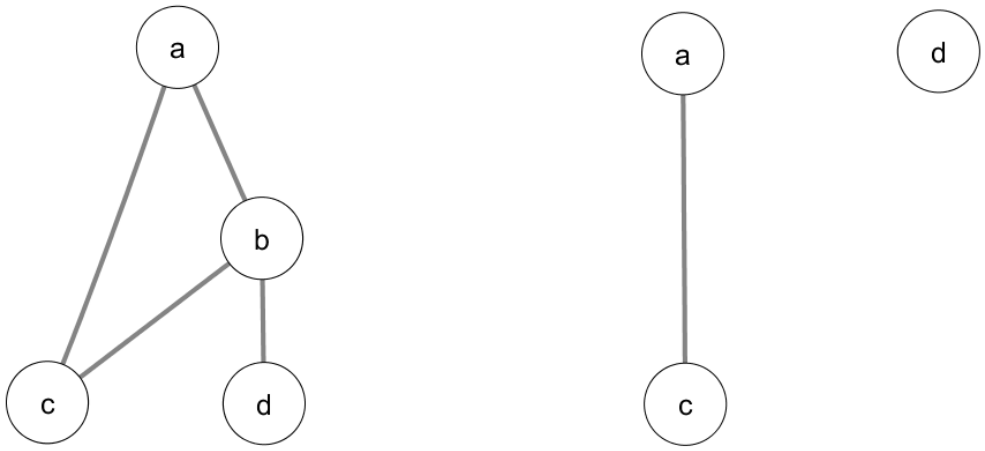


Figura 2: (izq.) Grafo no dirigido, (der.) Subgrafo inducido por $\{a, c, d\}$

Si en un grafo no dirigido $\mathcal{G}$ se tiene que existe al menos un camino entre V_i y $V_j \in V$ $\forall i \neq j$ entonces $\mathcal{G}$ es un grafo conexo. Para el caso de los grafos dirigidos, es conexo si el grafo no dirigido asociado es conexo. En la figura 2 el subgrafo inducido por $\{a, c, d\}$ es un grafo inconexo, pues no existe algún camino de d a c ó de d a a. Mientras que el grafo original sí es conexo.

Un grafo $\mathcal{G}$ es completo si existe una arista entre cualesquiera dos nodos. Formalmente, es completo si $V_i \to V_j$ ó $V_j \to V_i$ ó $V_i - V_j \ \forall \ V_i, V_j \in V$. En la figura 3 el primer grafo no es completo ya que no existe una arista que relacione a e con c ni d con c. El segundo grafo es completo por tener una arista para cualquier pareja de nodos.

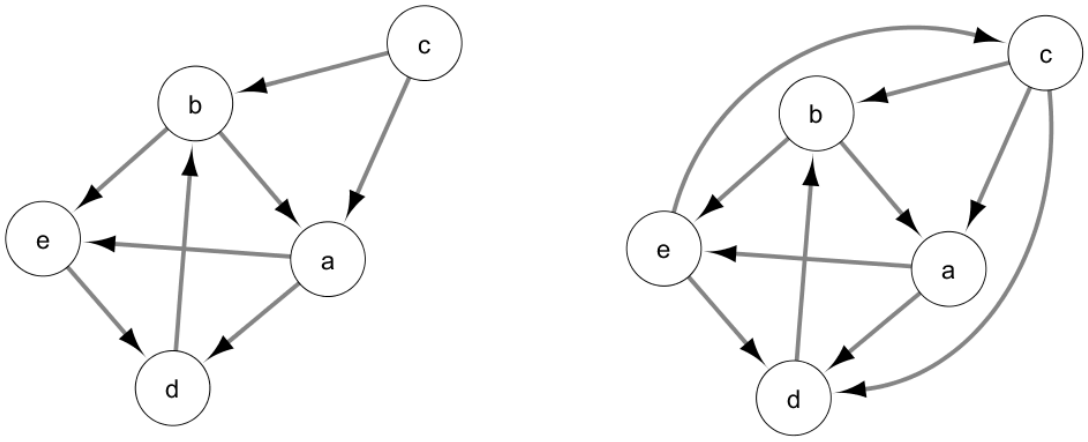


Figura 3: (izq.) Grafo no completo, (der.) Grafo completo

Se le conoce como conglomerado o clique al subconjunto de vértices $U \subset V$ que satisface que el subgrafo inducido por U es completo y es maximal, es decir, $\forall\ W \subset U$ el subgrafo inducido por W no es completo. Por ejemplo, el primer grafo de la figura 3 posee dos conglomerados, a saber, $\{a, b, d, e\}$ y $\{a, b, c\}$. Por su parte, el segundo grafo posee un único conglomerado, $\{a, b, c, d, e\}$.

En algunas ocasiones, la representación de un grafo puede revelar ciertas propiedades de forma más sencilla si la colocación de los nodos se realiza de alguna manera en específico. Usualmente, los nodos se ordenan de arriba hacia abajo, comenzando con los primeros nodos padres que aparezcan en la definición del grafo, a esta forma de colocar los nodos se le conoce como jerárquica. Otra representación común es la circular, la cual es de gran ayuda para comprobar si un nodo es completo. La figura 4 muestra los grafos de la figura 3 pero con la representación circular.

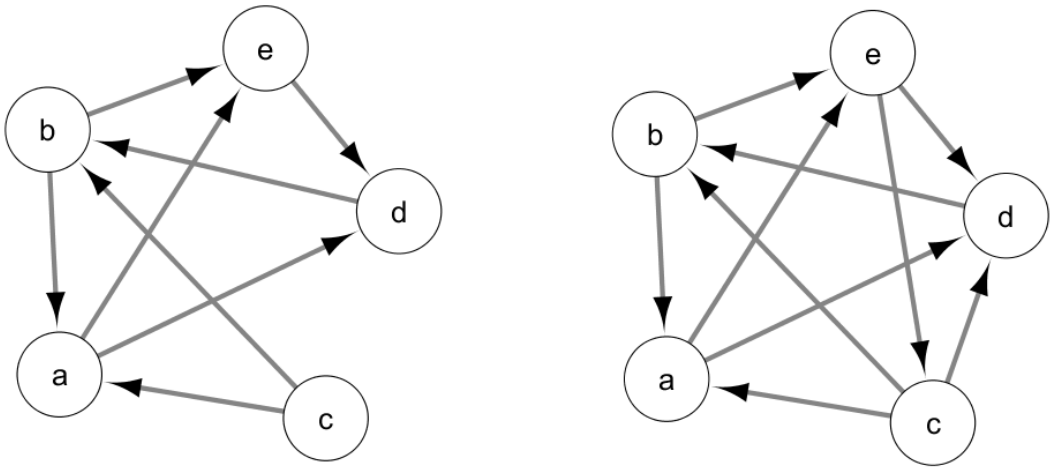


Figura 4: Representación circular de los grafos de la figura 3

Una de las estructuras más importantes dentro de la categoría de grafos dirigidos son las *estructuras-v*. Dichas estructuras están conformadas por una tercia de nodos, en la cual uno de ellos es hijo de los dos nodos restantes. En la figura 5 se muestran dos *estructuras-v*, en ambas, el nodo c es el hijo de a y b. La diferencia entre los dos grafos radica en la arista que une a los padres de c. Cuando no existe una arista entre los padres de una *estructura-v* se le conoce como *inmoralidad*. Recibe este nombre pues se considera como moral que dos padres de un hijo común estén unidos o tengan alguna forma de relacionarse, lo cual lleva a la siguiente definición.

El grafo moral de un grafo dirigido $\mathcal{G}$, es el grafo no dirigido asociado a $\mathcal{G}$ en el que se agrega una arista no dirigida entre los padres de cualquier inmoralidad contenida en $\mathcal{G}$. La figura 6 muestra un grafo dirigido y el grafo moral asociado.

Un grafo no dirigido conexo se denomina árbol si existe un único camino entre cualesquiera dos nodos. A partir de esta definición se puede deducir que un árbol es acíclico, ya que de existir un ciclo, podrían formarse dos caminos diferentes entre cualesquiera dos nodos

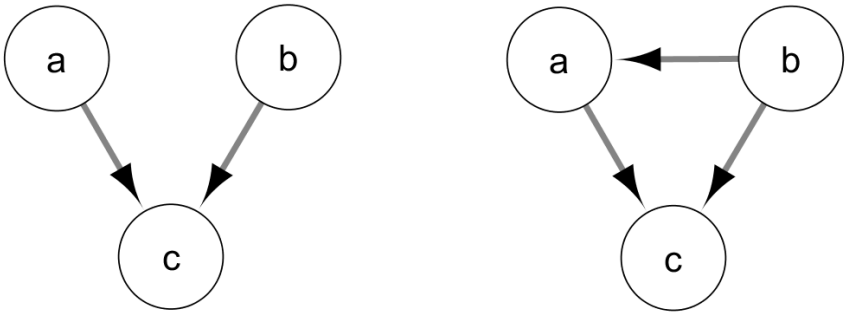


Figura 5: (izq.) Inmoralidad, (der.) *estructura-v* moral

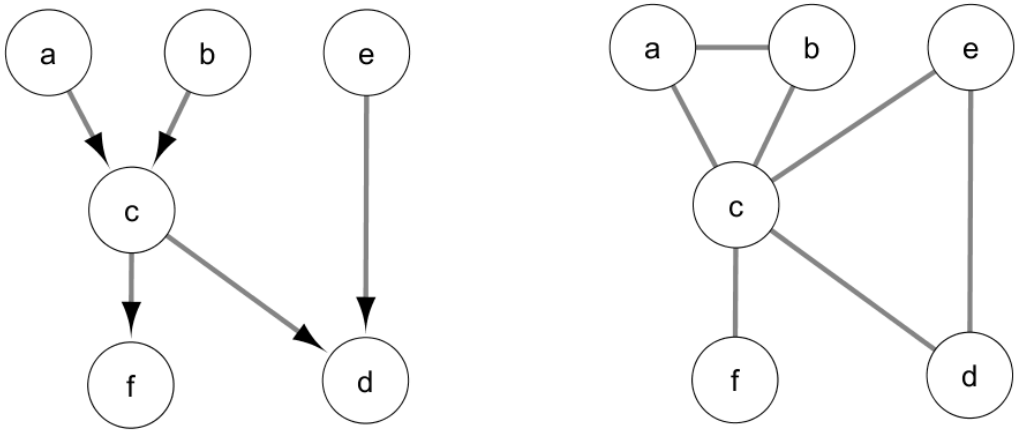


Figura 6: (izq.) Grafo dirigido y su (der.) grafo moral.

contenidos en el ciclo.

En un grafo dirigido $\mathcal{G}$, si el grafo no dirigido asociado es un árbol, entonces $\mathcal{G}$ es un árbol.

Si un grafo dirigido es un árbol, entonces será un árbol simple si cualquier nodo tiene a lo más un padre. Cuando alguno de los nodos tiene más de un padre se denomina poliárbol.

En un grafo no dirigido, en el cual existe un bucle de longitud mayor o igual a 4 ($V_1 - V_2 - \ldots - V_n - V_1$ $n \geq 4$) se dice que una arista es una cuerda si conecta a dos nodos no consecutivos en el bucle, es decir, $V_i - V_j$ es una cuerda si $\mid i - j \mid > 1$. Una cuerda en un bucle de longitud 4 forma dos bucles de longitud 3 en forma de triangulo, los cuales ya no pueden poseer cuerdas dado que todos sus nodos son adyacentes.

Un grafo no dirigido en el que cualquier bucle de longitud mayor o igual a 4 posee al menos una cuerda se denomina cordal o triangulado. A los grafos dirigidos se les considera triangulados si el grafo no dirigido asociado es cordal. Aunque un grafo no sea triangulado se

puede convertir en un grafo cordal si se agregan ciertas aristas para satisfacer la definición, este proceso se conoce como triangulación y existen diversos algoritmos para agregar aristas al grafo, aunque la triangulación no necesariamente es única. La figura 7 exhibe un grafo no cordal y su respectiva triangulación.

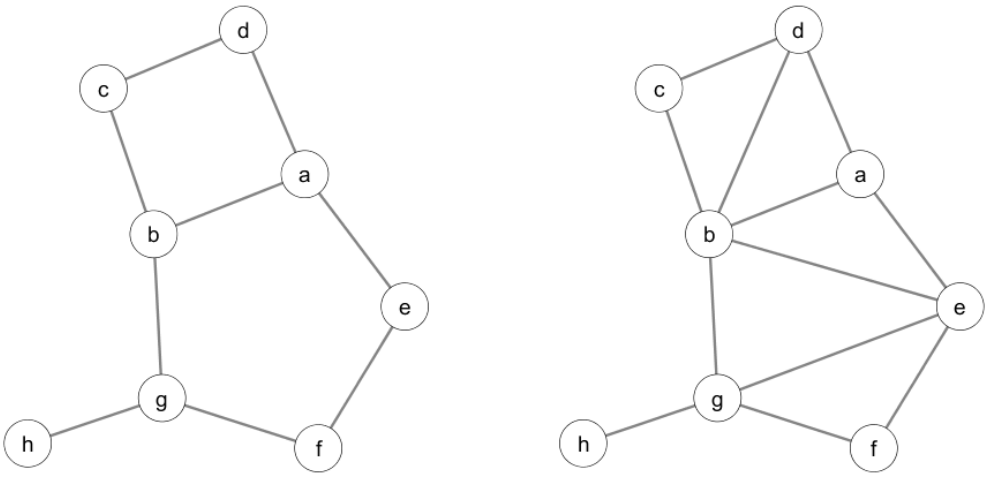


Figura 7: (izq.) Grafo no cordal y (der.) su triangulación.

Una ordenación (α) de un grafo $\mathcal{G} = (V, A)$ en el que $\mid V \mid = n$, es una función que asocia un número entero entre 1 y n a cada vértice. Por lo tanto, $\alpha = (\alpha_1, \ldots, \alpha_n)$ es una permutación de los nodos del grafo $\mathcal{G}$

En un grafo dirigido, una ordenación se denomina ancestral si siempre que $\alpha_i \rightarrow \alpha_j$ entonces $i < j$, es decir, que los ancestros de un nodo se encuentran antes que el nodo en la ordenación. Cabe resaltar que la ordenación ancestral de un grafo dirigido, no necesariamente es única. Por ejemplo, una posible ordenación ancestral para el grafo de la izquierda en la figura 6 es $\{e, a, b, c, d, f\}$, otra posible ordenación es $\{b, a, c, f, e, d\}$

Una ordenación de V en un grafo no dirigido se considera perfecta si para cualquier nodo α_i se tiene que el subgrafo asociado a todos sus nodos adyacentes ordenados anteriormente forman un grafo completo. Formalmente, se dice que una ordenación es perfecta si $\forall i > 1$ el subgrafo asociado a $Nb(\alpha_i) \cap (\alpha_1 \cup \ldots \cup \alpha_{i-1})$ es un grafo completo.

La definición de ordenación se puede extender para conjuntos de vértices de un grafo, tal es el caso de los cliques de un grafo. Considérese un grafo $\mathcal{G}$ que contiene los cliques $C_1, \ldots, C_n$, una ordenación de cliques (α) es una función que asocia un número entero entre 1 y n a cada clique.

Una ordenación α de cliques $C_1, \ldots, C_n$ satisface la *propiedad de intersección dinámica* si $\forall i \in (1, \ldots, n)$, $\alpha_i \cap (\alpha_1 \cup \ldots \cup \alpha_{i-1}) \subset \alpha_j$ para al menos un $j \in (1, \ldots, i-1)$. Cuando una ordenación satisface la *propiedad de intersección dinámica* implica que los nodos en

común entre el i-ésimo clique y los cliques anteriores, están contenidos en al menos un conglomerado anterior. A estos nodos en común se les denomina *separadores* y se denotan por $S_i = \alpha_i \cap (\alpha_1 \cup \ldots \cup \alpha_{i-1})$.

En un grafo no dirigido un conjunto separador, S de dos conjuntos de nodos A y B satisface que cualquier camino entre los nodos debe pasar por S. Los separadores S_i de una ordenación de cliques reciben este nombre por satisfacer la propiedad en estructuras particulares mostradas en los siguientes capítulos.

Una condición necesaria y suficiente para que exista la ordenación de cliques que satisfagan la propiedad de intersección dinámica es que el grafo sea triangulado. De la misma manera, un grafo admite una numeración perfecta si y sólo si es un grafo triangulado.

A un conjunto de conglomerados en $\mathcal{G}$ se le puede asociar un nuevo grafo $\mathcal{K} = (V', A')$ conocido como *grafo de conglomerados*. El conjunto V' consiste en cada uno de los conglomerados y existirá una arista entre dos nodos siempre que la intersección de dos conglomerados no sea vacía.

Un caso particular de un grafo de conglomerados es el *árbol unión* ó *árbol de cliques*. Se caracteriza por ser un árbol en el cual si existe un camino entre dos cliques, entonces los nodos en común de dichos cliques deben estar contenidos en todos los conglomerados presentes en el camino que los une.

En la figura 8 se observa un grafo no dirigido y su respectivo árbol unión. Cabe destacar que dicho árbol ya posee una de las posibles ordenaciones de cliques, además de satisfacer la propiedad de intersección dinámica. Los cliques son los siguientes:

- $C1 = \{a, b, c, d\}$
- $C2 = \{a, b, e, f, g\}$
- $C3 = \{a, i, c\}$
- $C4 = \{h, g\}$
- $C5 = \{j, d\}$

Para evidenciar la propiedad de intersección dinámica y la construcción del árbol unión, la figura 9 muestra cada uno de los cliques del grafo de la figura 8. El primer clique del árbol unión será aquel que tenga mas intersecciones con el resto de los cliques, esto para que sea el nodo raíz:

- $C1 = \{a, b, c, d\}$
- $C2 = \{a, b, f, g, e\}$
- $C3 = \{a, c, i\}$

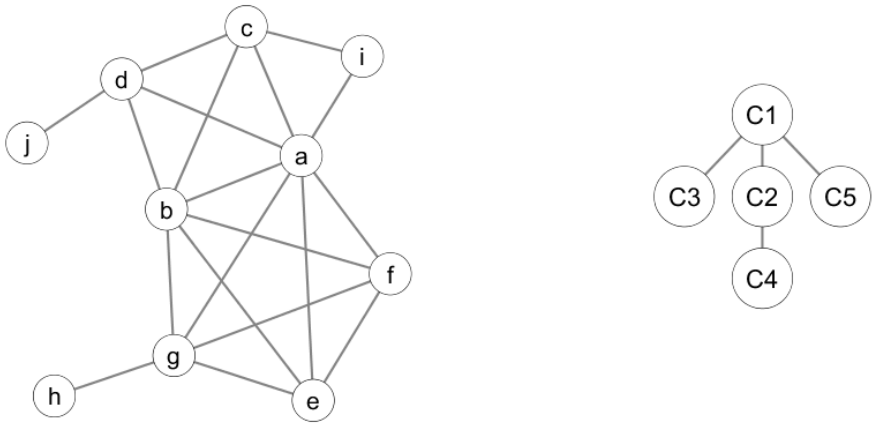


Figura 8: (izq.) Grafo no dirigido y (der.) su árbol de conglomerados asociado.

- $C2 = \{d, j\}$
- $C2 = \{h, g\}$

Se observa que $C1 \cap C2 = \{a, b\} \subset C1$, es por eso que existe una arista de $C1$ a $C2$. De manera similar, $C3 \cap (C1 \cup C2) = \{a\} \subset C1, C2$. Sin embargo, solo se agrega el arco de $C1$ a $C3$ para satisfacer la definición del árbol unión. Aplicando el razonamiento 2 veces más se obtiene que los siguientes conjuntos separadores son $\{g\} \subset C2$ y $\{d\} \subset C1$ que generan dos arcos más de $C2$ a $C4$ y de $C1$ a $C5$.

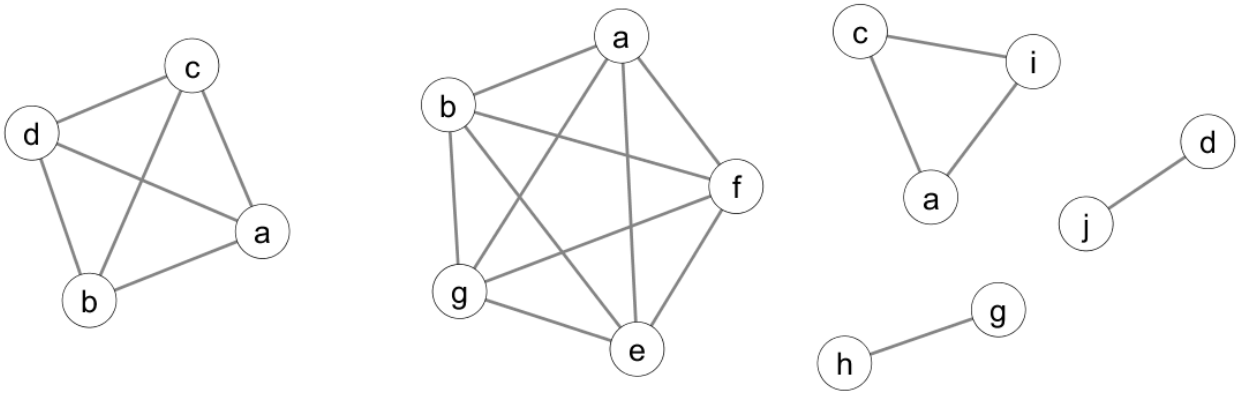


Figura 9: Cliques de la figura 8

Además de la representación gráfica, existe la representación numérica para un grafo $\mathcal{G}$. La representación numérica se realiza con base en matrices cuadradas, donde cada renglón, así

como cada columna, representa un nodo. Para los grafos dirigidos, la i, j-ésima entrada de la matriz representa la existencia de una arista que inicie i y termine en j si es igual a 1, en caso contrario será igual a 0. En el caso de grafos no dirigidos la i, j-ésima entrada es 1 si existe una arista entre el nodo i y el nodo j y 0 en cualquier otro caso. A esta matriz se le llama *matriz de adyacencia*. Por ejemplo, la matriz de adyacencia asociada al grafo dirigido de la figura 1 es:

$$\begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Una de las ventajas de este tipo de representación reside en la potenciación de la matriz de adyacencia, la cual nos indicará si entre dos nodos existe un camino que los conecte. Si existe un camino de longitud n entre i y j, entonces la i, j-ésima entrada de la matriz de adyacencia elevada a la n-ésima potencia sera distinta de 0.

3. Representación.

Una **red bayesiana** (**RB**) es un grafo acíclico dirigido $\mathcal{G} = (V, A)$ que representa de manera compacta a una probabilidad conjunta P sobre una colección de variables $\mathbf{X} = (X_1, \ldots, X_n)$ y el conjunto de independencias condicionales existentes entre las variables. Cada uno de los nodos en V está asociado a cada una de las variables aleatorias en $\mathbf{X}$ y las aristas entre dos nodos indican la dependencia directa entre dos variables. Si $X_i \to X_j$ se dice que X_i es una causa de X_j y X_j es un efecto de X_i.

La tarea de representación de una RB se basa en especificar la estructura y parámetros del modelo. Se entiende como estructura al DAG, ya sea en forma gráfica o numérica. A la estructura también se le conoce como la parte cualitativa del modelo, pues indica las relaciones que se satisfacen entre las variables. Mientras que los parámetros son las funciones de probabilidad condicional seleccionadas para cada nodo, conocidas como la parte cuantitativa del modelo porque determinan de manera numérica la intensidad de cada una de las relaciones descritas por la estructura.

3.1. Fundamentos de la representación mediante RB.

La especificación de una función probabilidad conjunta sobre un conjunto de variables aleatorias $\mathbf{X}$ se puede simplificar utilizando las independencias condicionales entre variables y la regla de la cadena (ecuación (2.3)). Por ejemplo, en una terna de variables aleatorias discretas X_1, X_2 y X_3, con n_1, n_2 y n_3 categorías, respectivamente, la función de probabilidad está dada por:

$$P(\mathbf{X}) = P(X_1, X_2, X_3) = P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_2, X_1)$$

Si además se conoce que las variables X_1 y X_2 son independientes ($X_1 \perp X_2$) y X_3 y X_2 son independientes dado X_1 ($X_3 \perp X_2 \mid X_1$) entonces, debido a las propiedades de independencia, la función de probabilidad puede expresarse como:

$$P(\mathbf{X}) = P(X_1)P(X_2)P(X_3 \mid X_1)$$

Antes de conocer las independencias entre las variables, la definición de la función de probabilidad conjunta requería especificar $n_1 * n_2 * n_3$ valores entre cero y uno, correspondientes a cada posible asignación de valores x_1, x_2 y x_3 de las variables aleatorias. Posterior a conocer las independencias y utilizar la regla de la cadena y propiedades de independencia, solo se necesitan especificar n_1 valores para la distribución marginal de X_1, n_2 valores para la distribución marginal de X_2 y $n_1 * n_3$ valores para la distribución condicional, dando un total de $n_1 + n_2 + (n_1 * n_3)$ valores a especificar. En caso de que el número de niveles de las 3 variables fuese el mismo, n, entonces en un inicio se deben especificar n^3 valores, mientras que aplicando las independencias solo se necesitan $2n + n^2$. La figura 10 muestra el comportamiento de la especificación de valores para el caso considerado. Sin embargo, el número de valores que tenemos que especificar para una función de distribución no solo depende de los niveles de las variables, sino que también del número de variables consideradas y las independencias que existan. Además, estos cálculos representan una cota superior, pues el número de parámetros a especificar es menor gracias a una dependencia con el número de categorías que más

adelante se explicará como parámetros no redundantes.

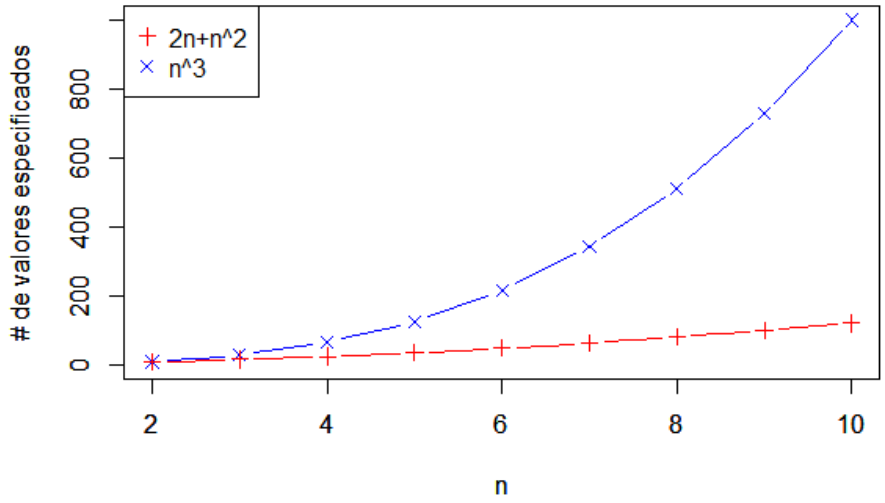


Figura 10: Especificación de la función de distribución considerando (azul) y sin considerar (rojo) relaciones de independencia entre tres variables

Otra de las ventajas de expresar la probabilidad conjunta como producto de funciones de probabilidad condicionada surge cuando se necesita incorporar nuevas variables al modelo. En el ejemplo mencionado anteriormente, si se desea agregar una cuarta variable X_4 al modelo, que posea n niveles se tendría que incorporar un cuarto factor, a saber, $P(X_4 \mid X_3, X_2, X_1)$. La incorporación de este factor implica la especificación de n^4 nuevos valores solo para esa función de probabilidad, mientras que el resto de los factores permanecen como antes al no involucrar a X_4. Por otro lado, si se opta por una probabilidad conjunta $P(X_1, X_2, X_3, X_4)$ se deben de especificar n^4 nuevos valores para cada asignación de $\mathbf{X}$. Hasta este punto pareciera que cualquiera de las dos opciones tienen el mismo costo, especificar n^4 valores, sin embargo, si la nueva variable resulta ser independiente a alguna de las tres variables consideradas anteriormente, la especificación se reduce a n^3 valores si se elige la especificación por producto de probabilidades condicionadas, mientras que en la probabilidad conjunta no se consigue ninguna reducción en el número de valores por especificar. Además, la probabilidad conjunta implica crear una función totalmente nueva, mientras que la opción de factores respeta las probabilidades anteriores y solo las multiplica por un nuevo factor.

Este tipo de razonamiento aplicado a un conjunto de variables constituye la base de la representación de una RB.

3.1.1. Independencias gráficas básicas.

Un grafo $\mathcal{G}$ refleja las independencias de las variables asociadas a los nodos del grafo, al conjunto de independencias existentes en el grafo se le denotará como $\mathcal{I}(\mathcal{G})$. Estas independencias se deducen a través de las aristas que conectan a los nodos, pues las aristas en una RB reflejan la influencia que tiene el nodo padre en el nodo hijo y viceversa. Por lo tanto, si no existe un camino de un nodo X_1 al nodo X_2, estos son independientes, pues no existe manera de como uno influya en el otro. Sin embargo, también existen independencias entre nodos conectados por algún camino. Si se considera el caso de un grafo con solo 2 nodos A y B, los únicos grafos posibles son donde $A \to B, B \to A$ y el grafo donde no existe ninguna arista. En los grafos que contienen una arista el conjunto $\mathcal{I}(\mathcal{G})$ es vacío, ya que el grafo indica que al conocer el valor de una variable, entonces la otra se ve influenciada y la probabilidad de tomar cierto valor puede verse modificada. En cambio, en el grafo sin aristas se tiene que $\mathcal{I}(\mathcal{G}) = \{(A \perp B)\}$, pues en el grafo al conocer el valor de A no hay manera de influir en B y viceversa.

Aumentando el número de nodos a 3 (A, B, C) y considerando solo grafos conexos, tenemos 4 tipos de grafos que contengan un solo camino no dirigido entre A y C:

- Camino causal: $A \to B \to C$
- Camino evidencial: $A \leftarrow B \leftarrow C$
- Causa común: $A \leftarrow B \to C$
- Efecto común: $A \to B \leftarrow C$

Cada grafo contiene un conjunto distinto de independencias, para evidenciarlas se consideran los siguientes casos.

Se conoce como disnea a la dificultad para respirar. Una de sus posibles causas es la bronquitis que a su vez es causada por fumar. Considerando disnea (D), bronquitis (B) y fumar (F) como una terna de variables binarias, las cuales representan la presencia o no de la enfermedad o hábito, entonces un camino causal sería el siguiente:

$$F \to B \to D$$

Intuitivamente, saber si un paciente fuma o no puede cambiar la probabilidad de presentar o no disnea, ya que al saber que una persona fuma, entonces la probabilidad de tener bronquitis aumenta y por lo tanto, la probabilidad de padecer disnea aumenta también. En caso de saber que una persona no fuma, entonces la probabilidad de bronquitis disminuye, al igual que la probabilidad de padecer disnea. Este razonamiento muestra que no existe una independencia entre D y F, pues fumar influye en la presencia de disnea. La independencia entre cualquier otra combinación de dos variables no existe, pues estarán unidas por una arista dirigida, lo cual representa una dependencia directa.

Sin embargo, si se conociera el estado de la variable B, entonces F dejaría de influir en D. Es decir, suponiendo que el paciente presenta bronquitis, entonces saber que el paciente

fuma o no ya no es relevante para determinar la probabilidad de padecer disnea, puesto que el saber que tiene bronquitis es suficiente para determinarlo. De la misma manera si no presenta bronquitis. Por lo tanto, $(D \perp F \mid B) \subseteq \mathcal{I}(\mathcal{G})$. En general, si existieran más causas para la variable B e incluso para la variable F, la influencia de estas se vería *bloqueada* cuando se conoce el valor de B.

El caso de un camino evidencial es similar, considerando las mismas variables tenemos el siguiente camino evidencial:

$$D \leftarrow B \leftarrow F$$

Este es el mismo grafo que el considerado para ejemplificar el camino causal. Visto desde el punto de vista de evidencias, conocer el estado de la variable D puede modificar la probabilidad de la variable B y en consecuencia F. Sin embargo, si conocemos el valor de B, entonces es suficiente para determinar la probabilidad de que el paciente fume. Lo cual induce que $(D \perp F \mid B) \subseteq \mathcal{I}(\mathcal{G})$. En general, cualquier otro efecto de la variable B no influye sobre F una vez que se conoce el valor de B.

Para el caso de una causa común, se considera que fumar es una causa para presentar tanto bronquitis como cáncer de pulmón (C). Dicho lo anterior, la estructura que modela a las variables F, B y C es:

$$C \leftarrow F \rightarrow B$$

En este caso, cada uno de los nodos hijo puede influir en el otro. Por ejemplo, el valor de la variable C representa evidencia para influir en el valor de F y al ser B causado por F, esta influencia se puede reflejar en B. En otras palabras, si conocemos que un paciente padece cáncer de pulmón, entonces la probabilidad de que padezca bronquitis se eleva, en caso de conocer que no padece este tipo de cáncer, entonces la probabilidad de que tenga bronquitis desciende.

Sin embargo, si conocemos el valor del nodo padre (la causa común), entonces uno de los nodos hijo ya no influye en el otro. Tratando de aplicar el mismo razonamiento que cuando no se conoce el valor de F, C ya no influye en el valor de F. Esto ya que cualquier valor que tome C no cambiará el valor que ya tomó F. Por lo tanto, la influencia de C en B se bloquea al conocer el valor de F. De manera similar, B no puede influir en C una vez que se conoce el valor de F. Se concluye que $(C \perp B \mid F) \subseteq \mathcal{I}(\mathcal{G})$

Una causa común es la base del modelo conocido como clasificador Naive Bayes, el cual consta de una variable C de clases, que es el nodo padre de las características $C_1, \ldots C_n$ y asume que estas características son independientes entre ellas dado la variable C, $(C_i \perp C_j \mid C)$.

Por último, se considera que la tuberculosis (T) y el cáncer de pulmón (C) son causas de disnea (D). La estructura asociada a estas relaciones se conoce como efecto común.

$$T \rightarrow D \leftarrow C$$

Un efecto común posee las características de una *estructura-v*, dos nodos padre de un mismo nodo hijo.

En este caso al conocer el estado de la variable intermedia D, los nodos padre son dependientes al ser causas directas que ocasionan el efecto compartido. Dependiendo del valor observado del nodo hijo, los nodos padres tenderán a tomar los valores que causan el estado de D, pero el valor de un nodo padre influye en el otro, por ejemplo, si uno de los padres toma un valor que no es tan probable que cause el estado de D, entonces el otro padre tiene que tomar los valores que hagan más probable la asignación de D, es decir, si sabemos que el paciente no presenta tuberculosis y sí disnea, entonces la probabilidad de que padezca cáncer de pulmón aumenta. A diferencia del resto de las estructuras de 3 nodos mencionadas anteriormente, no existe la independencia entre dos variables una vez que se conoce la variable intermedia, $(C \perp T \mid D) \notin \mathcal{I}(\mathcal{G})$.

Incluso si se observara un hijo o descendiente de D influiría en el valor que D pudiese tomar y se presentaría el mismo comportamiento de dependencia entre T y C.

Por otro lado, cuando la variable intermedia no es observada se obtiene la independencia entre las dos variables padre, $(T \perp C) \subseteq \mathcal{I}(\mathcal{G})$. Aplicando un razonamiento similar a los casos anteriores se muestra que la influencia de un nodo padre al otro esta *bloqueada* por el nodo hijo. Supongamos que conocemos el valor de T, entonces este valor influye en D, pero a diferencia del resto de las estructuras, D tiene más de un padre y para poder inferir sobre su posible valor es necesario conocer todas las causas que lo provocan. Por lo tanto, al no existir influencia de T en C se tiene que son independientes.

En las distintas estructuras presentadas, se observa como el conocer el valor de los padres de un nodo X es suficiente para determinar la probabilidad del nodo X e impide que nodos que no sean descendientes de X influyan en el valor de X. De manera similar, si conocemos el valor de los hijos de X, es suficiente para no permitir que nodos que no sean ascendientes de X influyan en el valor de X.

Incluso, para un nodo X, si se conociera el valor de los padres, (Pa_X) y los hijos (Ch_X), además de los nodos con los que comparte hijos (con los que forma *estructuras-v*), esto sería suficiente para que ningún otro nodo pudiera influir sobre X. Al conjunto $Pa_X \cup Ch_X \cup \{Y \in V | X \rightarrow C \leftarrow Y, C \in Ch_x\}$ se le conoce como *Markov blanket* de X (Mb_X).

En general, lo anterior se satisface para conjuntos de más de un nodo y caminos más largos, lo cual permite la caracterización de una RB por medio de los padres e hijos de cada uno de los nodos contenidos en V.

3.1.2. Factorización.

El principal supuesto que se realiza sobre las redes bayesianas se conoce como *el supuesto de Markov*, el cual establece que sobre un grafo $\mathcal{G}$, cualquier nodo $X \in V$ es independiente

de cualquier nodo que no sea descendiente de X, (conjunto $NoDescendientes_X$) una vez que se conoce el valor de los nodos que son padres de X, es decir:

$$(X \perp NoDescendientes_X \mid Pa_X) \qquad \forall X \in V \tag{3.1}$$

El conjunto de todas las independencias derivadas del supuesto de Markov se conoce como independencias locales del grafo $\mathcal{G}$ ($\mathcal{I}_l(\mathcal{G})$) y representa un subconjunto de las independencias derivadas por el grafo ($\mathcal{I}_l(\mathcal{G}) \subseteq \mathcal{I}(\mathcal{G})$).

El supuesto de Markov permite que la probabilidad conjunta a la cual está representando el grafo se simplifique de manera considerable, reexpresandola como producto de funciones de probabilidad condicional.

Más aún, si se considera a $\mathbf{Y} = (Y_1, \ldots, Y_n)$ como un ordenación ancestral de $\mathbf{X}$ y al grafo $\mathcal{G} = (\mathbf{X}, A)$, entonces la probabilidad conjunta se reduce a:

$$P(\mathbf{X}) = P(\mathbf{Y}) = \prod_{i=1}^{n} P(Y_i \mid Y_1, \ldots, Y_{i-1}) = \prod_{i=1}^{n} P(Y_i \mid Pa_{Y_i}) \tag{3.2}$$

A la ecuación (3.2) se le conoce como *regla de la cadena para redes bayesianas* y se satisface debido a que ninguno de los descendientes de Y_i puede estar en el conjunto de los $i-1$ nodos anteriores al ser una ordenación ancestral. Por lo tanto, el conjunto sobre el cual se condiciona cada uno de los factores se reduce al conjunto de los padres del respectivo nodo.

Si una función de probabilidad conjunta P sobre un espacio $\mathbf{X}$ se puede expresar como la ecuación (3.2) para un grafo $\mathcal{G} = (\mathbf{X}, A)$, entonces se dice que la función P factoriza de acuerdo al grafo $\mathcal{G}$.

Para ejemplificar las ventajas del supuesto de Markov, se considera la base de datos ASIA que contiene 8 variables binarias ($\mathbf{X}$=(D,T,C,B,A,F,X,P)) correspondientes al siguiente problema planteado en Lauritzen y Spiegelhalter (1998):

"La falta de aliento, disnea (D) puede deberse a la presencia tuberculosis (T), cáncer de pulmón (C) o bronquitis (B), o ninguno de ellos, o más de uno de ellos. Una visita reciente a Asia (A) aumenta las posibilidades de padecer tuberculosis, mientras que se sabe que fumar (F) es un factor de riesgo tanto para el cáncer de pulmón como para la bronquitis. Los resultados de una radiografía de tórax (X) no distinguen entre el cáncer de pulmón y la tuberculosis (P), como tampoco la presencia o ausencia de disnea."

El problema planteado se ha modelado con la red bayesiana presentada en la figura 11, cabe resaltar que el nodo P (presencia de alguna de las dos enfermedades; tuberculosis o cáncer de pulmón) no es una variable aleatoria, ya que su valor se define de manera deterministica a través de las variables T y C. Es decir, si alguna de las dos variables T o C muestra un resultado positivo, entonces P estará definida como $'si'$ y $'no'$ en cualquier otro caso.

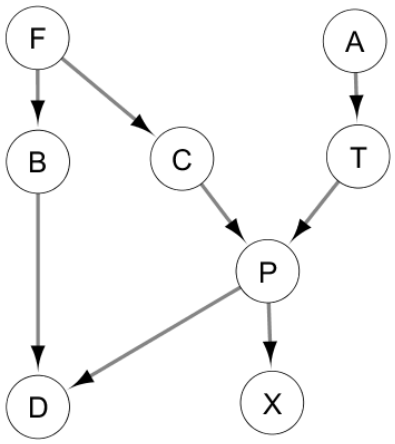


Figura 11: Grafo asociado al problema de disnea (grafo $\mathcal{A}$).

La estructura del grafo $\mathcal{A}$ junto con el supuesto de Markov establecen las siguientes independencias locales:

$$
\begin{aligned}
(F \perp T, A \mid \emptyset) &\in \mathcal{I}_l(\mathcal{A}) \\
(A \perp C, B, F \mid \emptyset) &\in \mathcal{I}_l(\mathcal{A}) \\
(C \perp A, B, T \mid F) &\in \mathcal{I}_l(\mathcal{A}) \\
(B \perp T, C, A, X, P \mid F) &\in \mathcal{I}_l(\mathcal{A}) \\
(T \perp B, F, C \mid A) &\in \mathcal{I}_l(\mathcal{A}) \\
(P \perp B, F, A \mid T, C) &\in \mathcal{I}_l(\mathcal{A}) \\
(X \perp D, T, C, B, A, F \mid P) &\in \mathcal{I}_l(\mathcal{A}) \\
(D \perp T, C, A, X, F \mid P, B) &\in \mathcal{I}_l(\mathcal{A})
\end{aligned}
$$

Todas las independencias obtenidas surgen de considerar la ecuación (3.1)

La primera independencia condicional de la lista anterior se deduce que a partir de que el conjunto Pa_F es un conjunto vacío, es por eso, que el conjunto $NoDescendientes_F$ se conforma únicamente por nodos en los que cualquier camino no dirigido que conecte con F está presente una v-estructura del grafo $\mathcal{A}$. En este caso son todos los nodos que no sean T ó A.

Mientras que la última independencia condicional de la lista, $Descendientes_D$ es el conjunto vacío, lo cual implica que $NoDescendientes_D$ es el conjunto de nodos en los que cualquier camino no dirigido hacia D contiene a algún padre de D como penúltimo nodo del camino. Al conocer el valor de los elementos en Pa_D la influencia de cualquier otro nodo se ve bloqueada.

El resto de las independencias locales del grafo $\mathcal{A}$ se deduce a través de encontrar los padres y no descendientes de cada nodo.

Considerando $\mathbf{Y}$=(F,A,B,C,T,P,D,X) como una ordenación ancestral de $\mathbf{X}$, la regla de la cadena para redes bayesianas establece la siguiente factorización:

$$P(\mathbf{X}) = P(F)P(A)P(B \mid F)P(C \mid F)P(T \mid A)P(P \mid C,T)P(D \mid B,P)P(X \mid P) \qquad (3.3)$$

Recordando que la suma sobre todos los posibles valores del rango de un vector aleatorio es igual a 1 y considerando que para una distribución conjunta la probabilidad de cada posible asignación es un parámetro, entonces, se define como parámetro redundante al último que se especifica, esto se debe a que este parámetro está condicionado al valor que tomaron el resto de parámetros, en otras palabras, suponiendo n posibles asignaciones, el parámetro redundante debe ser igual $1 - \Sigma_{i=1}^{n-1} parametro_i$. Los parámetros que no son el parámetro redundante se conocen como parámetros libres o parámetros no redundantes, pues no existe alguna restricción sobre ellos que dependa de algún otro parámetro.

Una vez que se conoce está definición y considerando que en $\mathbf{X}$ todas las variables son binarias, la especificación de la probabilidad conjunta $P(\mathbf{X})$ requiere la especificación de $2^8 - 1$ (255) parámetros no redundantes. En cambio, aplicando el mismo razonamiento, la ecuación (3.3) requiere 1 parámetro no redundante para el primer y segundo factor respectivamente, 2 parámetros no redundantes para cada uno de los factores tres, cuatro, cinco y ocho y 4 parámetros para los factores sexto y séptimo, obteniendo así un total de 18 parámetros.

Por ejemplo, considerando $P(B|F)$, se tienen que especificar dos funciones de probabilidad correspondientes a $P(B|F = si)$ y $P(B|F = no)$, como B es binaria, entonces, los parámetros libres podrían ser $P(B = si|F = si)$ y $P(B = si|F = no)$. Con esto, $P(B = no|F = si) = 1 - P(B = si|F = si)$ y $P(B = no|F = no) = 1 - P(B = si|F = no)$. Por lo que se especificó toda la distribución $P(B|F)$ con solo 2 parámetros.

La reducción de parámetros obtenida a partir de las independencias locales del grafo $\mathcal{A}$ es considerable, además de que facilita la tarea de la adquisición de conocimiento, ya que si se consulta a un experto para determinar la probabilidad de los eventos, será más sencillo, tanto para el experto como para el diseñador de la RB, definir 18 parámetros asociados a asignaciones especificas que encontrar los 255 valores para cada posible asignación de las 8 variables de interés.

La correcta definición de la parte cualitativa (estructura del grafo) será fundamental para evidenciar el valor agregado que tiene el optar por un modelo gráfico probabilístico, pues de ella emanan las independencias que se consideraran en el modelo. La parte cuantitativa es la definición de cada una de las funciones de probabilidad condicionada que constituyen los factores de la regla de la cadena para redes bayesianas. Conforme su definición sea lo más apegada a la realidad del problema, los resultados de las tareas de inferencia podran ser más precisos.

3.2. Estructura, parte cualitativa.

Las independencias generadas a partir del supuesto de Markov y un grafo $\mathcal{G}$ solo son un subconjunto de todas las independencias que pueden ser deducidas de la estructura de $\mathcal{G}$. Por

lo tanto, existen más independencias que pueden encontrarse en el grafo $\mathcal{G}$. Afortunadamente, existe un criterio para deducir la mayoría de las relaciones de dependencia e independencia contenidas en la estructura de un grafo.

3.2.1. D-separación

El criterio de *separación dirigida* o *d-separación* sirve para deducir las independencias contenidas en la estructura de un grafo $\mathcal{G}$. De manera general, este criterio se basa en analizar todos los posibles caminos entre dos conjuntos de nodos y se puede definir con base en el estudio realizado anteriormente con los grafos conexos de 3 nodos.

Antes de definir formalmente el criterio de *d-separación*, es importante definir la separación en grafos no dirigidos, o simplemente, criterio de *separación*.

En un grafo no dirigido $\mathcal{H} = (V, A)$, se dice que dos conjuntos disjuntos $X, Y \subset V$ son separados por un tercer conjunto $Z \subset V$ si y solo si todo camino de un elemento de X a un elemento de Y contiene al menos un nodo de Z. Es decir, para pasar de un nodo de X a uno de Y, necesariamente se tiene que pasar por un nodo en Z.

Si X y Y están separados por Z en el grafo $\mathcal{H}$, implica que $(X \perp Y \mid Z) \subset \mathcal{I}(\mathcal{H})$, ya que la influencia de un nodo de X a uno de Y se ve bloqueada al conocer el valor de los nodos en el conjunto Z.

Por ejemplo, la figura 12 muestra como los conjuntos $\{C, D\}$ y $\{A, F\}$ están separados por el conjunto $\{B, E\}$.

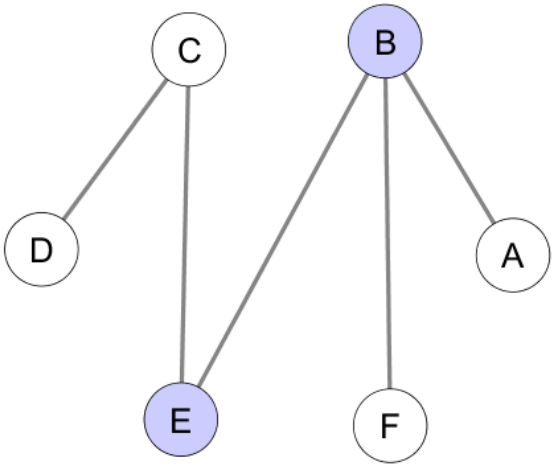


Figura 12: Ejemplo de separación.

El entender el criterio de separación permite visualizar que se espera del criterio de *d-separación*, básicamente, es definir una regla mediante la cual la influencia de un conjunto de variables no pueda llegar a otro conjunto disjunto. Sin embargo, como se analizó en la sección 3.1.1, en los grafos dirigidos, conocer el valor de una variable en algunos casos puede implicar una relación de independencia y en otros tantos una relación de dependencia. De cualquier forma, conocer en que estructuras implica dependencias y en cuales independencia será la clave para definir el criterio de *d-separación*. De acuerdo al análisis, las v-estructuras eran los caminos que presentaban dependencia al conocer el valor de una variable intermedia, mientras que en el resto de estructuras implicaba una independencia.

En un grafo dirigido $\mathcal{G} = (V, A)$, se dice que dos conjuntos disjuntos $X, Y \subset V$ están *separados dirigidamente* o *d-separados* por un conjunto disjunto Z si y solo si en todo camino no dirigido entre un nodo de X a un nodo de Y existe al menos un nodo B tal que:

1. Si B es el hijo en una v-estructura en el grafo $\mathcal{G}$ ($A \rightarrow B \leftarrow C$), entonces B ni ninguno de sus descendientes ($Descendientes_B$) existe en el conjunto separador Z.

2. Si B no es el hijo en una v-estructura en el grafo $\mathcal{G}$, entonces B existe en el conjunto separador Z.

La *d-separación* de dos conjuntos X y Y por medio del conjunto Z implica la independencia $(X \perp Y \mid Z)$. Por ejemplo, considerando de nuevo el grafo $\mathcal{A}$ de la figura 11, se puede observar en la figura 13 como el conjunto $\{P, B\}$ separa a los conjuntos $\{F, C\}$ y $\{D\}$. Por lo tanto, $(F, C \perp D \mid P, B) \subset \mathcal{I}(\mathcal{G})$

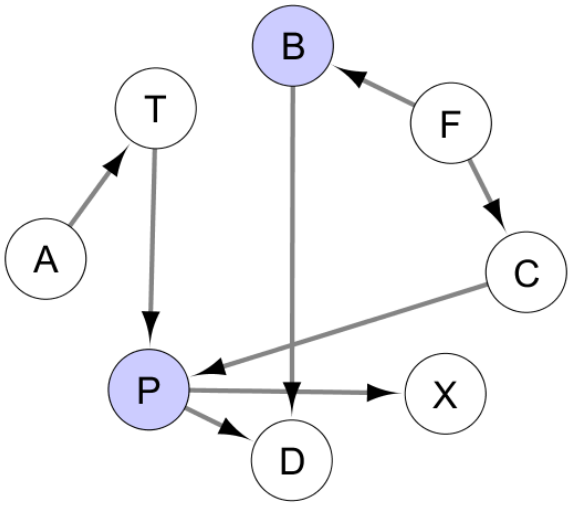


Figura 13: Ejemplo de *d-separación* en el grafo $\mathcal{A}$.

A los conjuntos que no satisfacen el criterio de separación o *d-separación* se le asigna una relación de dependencia, ya que la existe al menos un camino por el cual un conjunto puede influir en los nodos del otro conjunto. Es por esto que las relaciones de independencias $(X \perp Y \mid Z)$ para 3 conjuntos $X, Y, Z \in \mathcal{G}$ que no satisfacen los criterios de separación o *d-separación*, no existen en el conjunto de independencias que pueden ser deducidas a

través del grafo $\mathcal{G}$. Por ejemplo, en la figura 12 los nodos A y E no están separados por F, por lo tanto, $(A \perp E \mid F) \notin \mathcal{I}(\mathcal{G})$. De manera similar, en la figura 13, $(A \perp X \mid C, D) \notin \mathcal{I}(\mathcal{A})$

Se puede decir que el criterio de separación es más sencillo que el criterio de *d-separación*, puesto que la separación no considera la estructura de los caminos que conectan a dos conjuntos disjuntos X y Y, mientras que la *d-separación* si. Afortunadamente, el criterio de *d-separación* se puede replantear utilizando el criterio de separación:

En un grafo dirigido $\mathcal{G} = (V, A)$, se dice que dos conjuntos disjuntos $X, Y \subset V$ están d-separados por un conjunto disjunto Z si y solo si Z separa a X y Y en el grafo moral del grafo inducido por el conjunto $Ancestral(\{X, Y, Z\})$.

Analizando la misma relación de independencia, $(F, C \perp D \mid P, B)$, la figura 14 muestra tanto el grafo inducido por el conjunto $Ancestral(F, C, D, P, B)$, como el grafo moralizado. En este último se observa que $\{P, B\}$ efectivamente separa al conjunto $\{F, C\}$ del conjunto $\{D\}$.

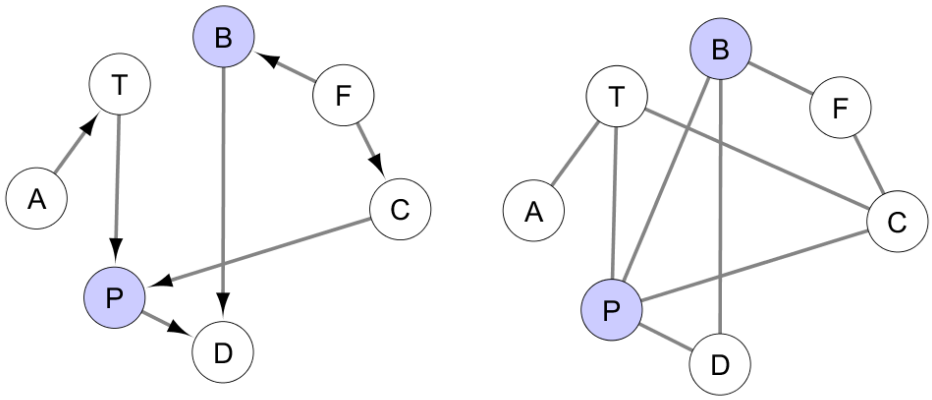


Figura 14: (izq) Grafo inducido por $Ancestral(F, C, D, P, B)$, (der) grafo inducido moralizado.

3.2.2. Más independencias.

Aplicando el criterio de *d-separación* a todos los posibles conjuntos disjuntos de un grafo $\mathcal{G}$ se pueden obtener un gran número de independencias asociadas a la estructura del grafo. Sin embargo, en casos específicos, el criterio de *d-separación* no es suficiente para detectar ciertas independencias.

Un caso particular de independencias que no son deducidas a partir del criterio de *d-separación*, surge cuando un nodo es una función determinística de sus padres. Por ejemplo, el nodo P (presencia de cáncer pulmonar o tuberculosis) en el grafo $\mathcal{A}$.

El criterio de *d-separación* indica que $(D \perp X \mid C, T) \notin \mathcal{I}(\mathcal{A})$, es decir, $\{C, T\}$ no separa a D de X. Pero analizándolo a detalle, si se conoce el valor de C y T, entonces el valor de P también se conoce, pues es una función determinística de sus padres. Se concluye que $(D \perp X \mid C, T) \subset \mathcal{I}(\mathcal{A})$, ya que P bloquea la influencia de D en X al ser su único padre. Este argumento se visualiza en la figura 15

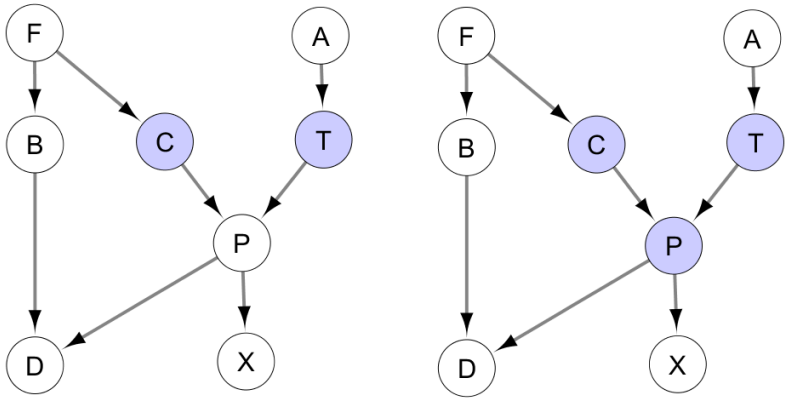


Figura 15: Incorporación de una variable determinística al conjunto de variables observadas al conocer sus padres.

Incluso, al conocer ciertos valores puntuales de una sola variable, el valor de P podría quedar determinado. Por ejemplo, si sabemos que $T = si$, entonces $P = si$ sin importar el valor que pueda tomar su nodo padre C. Esto deriva en independencias como $(F \perp X \mid T = si)$, pues P se incluye como un valor conocido o evidencia cuando $T = si$ y de manera análoga cuando $C = si$.

A este tipo de independencias, que surgen al conocer el valor particular de alguna variable, se conocen como *independencias de contexto específico*. Sin embargo, con el criterio de *d-separación* no se pueden deducir estas independencias.

Así como el conocer el valor de los padres o valores específicos de algunos de ellos determina el valor de un nodo que es una función determinística, también el conocer el valor de ese nodo puede revelar el valor de sus padres. En el grafo $\mathcal{A}$, si se conoce que $P = no$, entonces $C = no$ y $T = no$, lo cual puede generar más independencias de contexto específico. Por ejemplo, $(D \perp C \mid P = no)$, ya que al conocer que $P = no$, $C = no$ es un hecho y conocer si existe la presencia o no de disnea no lo cambiará.

3.2.3. I-mapas y P-mapas

El objetivo de la estructura es plasmar las relaciones de independencia que existen entre las variables en $\mathbf{X}$ y que estas sean satisfechas en la función de probabilidad que se desea representar. Es por eso que se esperaría que todas las relaciones deducidas por el grafo se cumplan en la función de probabilidad y viceversa. Denotando al conjunto de independencias de la función de probabilidad P como $\mathcal{I}(P)$, entonces se espera que $\mathcal{I}(\mathcal{G}) = \mathcal{I}(P)$. Cuando

esto sucede, se dice que $\mathcal{G}$ es un *mapa perfecto* o *P-mapa* para P.

Una de las desventajas de los modelos gráficos probabilísticos es que no todas las funciones de distribución tienen asociado un P-mapa y tampoco es posible caracterizar a aquellas que sí lo poseen.

Cuando no existe un P-mapa se opta por un grafo que reporte tantas relaciones de independencia en P como sea posible y no contenga relaciones que no se cumplan en P. En otras palabras, que el conjunto de independencias del grafo sea un subconjunto de las independencias de la función de probabilidad, $\mathcal{I}(\mathcal{G}) \subset \mathcal{I}(P)$. En estos casos se dice que $\mathcal{G}$ es un *mapa de independencias* o *I-mapa* para P.

Usualmente, para construir una RB no se conoce la función de probabilidad P, pues puede ser muy complicada su definición y es por eso que se opta por las RB. Si se conociera esta función también se conocerían las relaciones de independencia que se esperan del modelo gráfico probabilístico. En cambio, cuando no se conoce la función de probabilidad, se comienza proponiendo una lista inicial de independencias que corresponden al negocio y solo un experto puede establecerlas. Posteriormente, a la lista inicial se le aplican las propiedades de simetría, descomposición, unión débil y contracción, las relaciones resultantes son validadas por el experto y si el conjunto de relaciones de independencia resultante es cerrado bajo las 4 propiedades mencionadas anteriormente en este párrafo, entonces existe un I-mapa de P que satisface las independencias.

De acuerdo a lo anterior, se puede argumentar que si P factoriza de acuerdo a $\mathcal{G}$ es equivalente a señalar que $\mathcal{G}$ es un I-mapa de P.

3.3. Parámetros, parte cuantitativa.

Como se ha visto anteriormente, una RB simplifica la especificación de una probabilidad conjunta. En el caso de una RB multinomial, esto se ve reflejado en la especificación de un parámetro de una distribución (bernoulli) en caso de que el factor pertenezca a una variable binaria o múltiples parámetros si la variable tiene más de dos categorías.

La representación más común de los parámetros es una tabla de probabilidad, en la cual se encuentren los valores de la probabilidad de una variable dado el valor de sus padres. Se asigna una tabla de probabilidad a cada nodo en el grafo y cada una de estas tablas representa un factor en la regla de la cadena para redes bayesianas. Por ejemplo, en el grafo $\mathcal{A}$, una posible selección de parámetro para el nodo D se muestra en la tabla 1.

La tabla 1 ejemplifica la reducción de especificación de parámetros mencionada con anterioridad para la ecuación (3.3). Se estableció que para el factor $P(D \mid P, B)$ se necesitaba especificar solo 4 parámetros no redundantes. Estos 4 parámetros corresponden a los valores numéricos de la columna $D = si$ o la columna $D = no$. Una vez que se eligen los valores de una columna, el resto de los valores se determinan considerando que por renglones, los valores numéricos deben sumar 1.

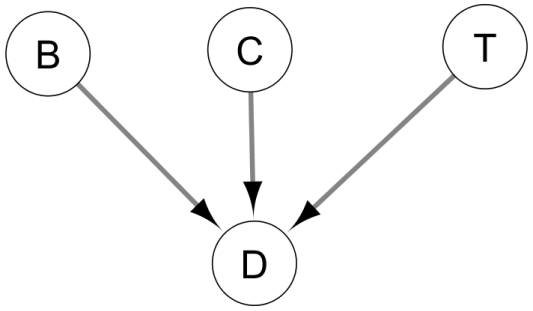
	D	si	no
P	**B**		
si	si	.90	.10
	no	.70	.30
no	si	.80	.20
	no	.10	.90

Tabla 1: $P(D \mid P, B)$

Aunque las tablas simplifican la especificación de valores, puede persistir el mismo problema que la función de distribución original a medida que el número de padres y categorías crece. Conforme existan más padres el número de valores requeridos crece de manera exponencial. Un par de posibles soluciones son la especificación de parámetros a través del modelo noisy-or en caso de que cada uno de los padres tenga un efecto independientemente de los otros, generalmente se utiliza en los problemas enfermedades-síntomas. La segunda solución son los árboles de decisiones, aplicables cuando la tabla contienen una gran cantidad de valores repetidos, esto regularmente sucede cuando se presentan independencias derivadas de relaciones determinísticas o secuenciales. Ambas soluciones se presentan a continuación.

Para ejemplificar el caso de los arboles de decisión, supongamos que la disnea es el nodo hijo de 3 enfermedades, tuberculosis, cáncer de pulmón y bronquitis. Además, suponga que la selección numérica de parámetros está dada por la figura 16.

Cabe resaltar que la selección de valores numéricos en la figura 16 implica independencias que no se detectan a partir del criterio de *d-separación*. Por ejemplo, $(D \perp C, B \mid T = si)$.



			D	si	no
T	**C**	**B**			
si	si	si		.90	.10
		no		.90	.10
	no	si		.90	.10
		no		.90	.10
no	si	si		.80	.20
		no		.80	.20
	no	si		.70	.30
		no		.10	.90

Figura 16: Disnea como efecto de 3 enfermedades y su tabla de probabilidad.

Esto se debe a que la selección hace que el nodo D pueda verse como una función de

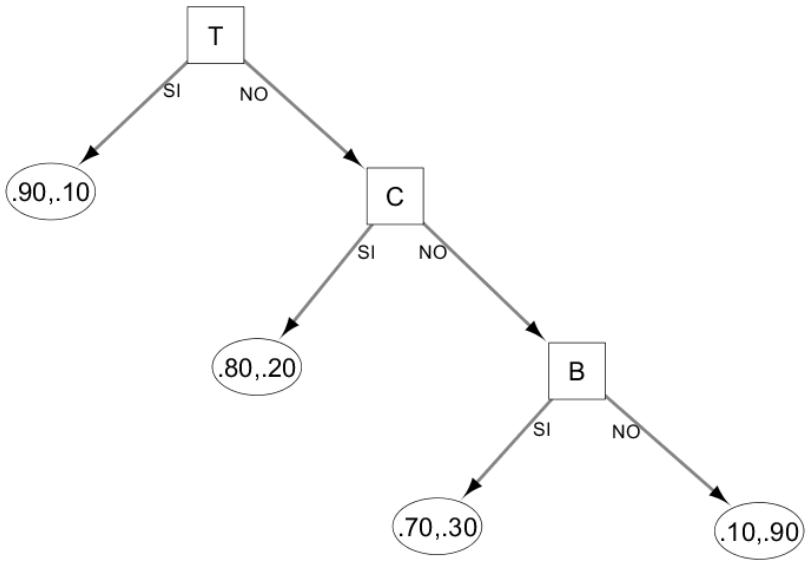


Figura 17: Árbol de decisión de $P(D \mid Pa_D)$.

determinística de sus padres. Esta función puede resumirse a:

$$P(D = si \mid T, C, B) = \begin{cases} .90 & \text{si } T = si \\ .80 & \text{si } C = si, T = no \\ .70 & \text{si } B = si, C = no, T = no \\ .10 & \text{en otro caso} \end{cases}$$

La repetición de valores en la tabla se omite al expresarlo como una función, además de que reduce los 8 parámetros no redundantes a solo 4 parámetros.

La función determinística y la selección de sus parámetros se puede expresar de manera gráfica mediante el árbol de decisión presentado en la figura 17. Cada camino en el árbol que comience en el nodo raíz (el nodo sin padres) representa una asignación de los nodos padres de D y el nodo hoja (nodo sin hijos) representa el valor asociado a la dupla $(P(D = si \mid Pa_D), P(D = no \mid Pa_D))$.

Se opta por el modelo noisy-or para representar un factor de la expresión dada por la regla de la cadena para redes bayesianas para una variable binaria, cuando se conoce la influencia por separado de cada uno de los padres sobre la variable en cuestión. Además la influencia de cada uno de ellos es independiente y los padres también son variables binarias. Bajo estos supuestos, la función de densidad queda determinada por el producto de la influencia de solo las variables que están presentes en la evidencia. Es decir, solo se considerarán las variables que tengan una categoría acordada desde un inicio.

Considerando el grafo de la figura 16, a continuación, se presenta el ejemplo de la selección

de valores para la influencia de cada una de los padres sobre la variable D:

$$P(D = si \mid T = si, B = no, C = no) = .70 = \lambda_T$$
$$P(D = si \mid T = no, B = si, C = no) = .85 = \lambda_B$$
$$P(D = si \mid T = no, B = no, C = si) = .90 = \lambda_C$$

Utilizando funciones indicadoras ($\mathbb{1}$) con la categoría de interés como sí, el modelo nosiy or establece que la función de densidad para D es:

$$P(D = no \mid T, B, C) = \Pi_{x \in Pa_D}(1 - \lambda_x)^{\mathbb{1}(x)} = (1 - \lambda_T)^{\mathbb{1}(T)} * (1 - \lambda_B)^{\mathbb{1}(B)} * (1 - \lambda_C)^{\mathbb{1}(C)} \quad (3.4)$$

De la ecuación (3.4) se deduce que:

$$P(D = si \mid T, B, C) = 1 - (1 - \lambda_T)^{\mathbb{1}(T)} * (1 - \lambda_B)^{\mathbb{1}(B)} * (1 - \lambda_C)^{\mathbb{1}(C)} \quad (3.5)$$

Al modelo noisy-or se le puede agregar un parametro λ_0, el cual indica la influencia de variables no consideradas en el modelo. Usualmente esta influencia no debe ser tan significativa como las variables que se encuentran en el modelo. Lo anterior agrega el factor $(1 - \lambda_0)$ al producto en la ecuación (3.4), considerado en la ausencia de la carecterística de interés en todas las variables. Por ejemplo, para la variable D se puede agregar $\lambda_0 = .10$ y solo se considera en el producto cuando todas las variables son *no*.

La especificación de los valores de λ_x, $x \in Pa_D$ simplifica la selección de valores en una tabla de probabilidad, pues como se ve en la tabla 2, se requieren 8 valores no redundantes, mientras que el modelo noisy-or solamente 4, 3 valores para cada padre y uno para la influencia no explicada por dichas variables.

		D	si	no
T	**C**	**B**		
si	si	si	.9955	.0045
		no	.97	.03
	no	si	.955	.045
		no	.70	.30
no	si	si	.985	.015
		no	.90	.10
	no	si	.10	.15
		no	.10	.90

Tabla 2: Tabla de probabilidad para D inducida por la selección de λ_T , λ_B, λ_C , λ_0

3.4. RB equivalentes.

A pesar de que dos RB tengan diferentes grafos asociados, estas pueden poseer el mismo conjunto de independencias deducido por sus respectivos grafos. Es decir, para dos grafos $\mathcal{G}, \mathcal{G}'$ se tiene que $\mathcal{I}(\mathcal{G}) = \mathcal{I}(\mathcal{G}')$. Por ejemplo, consideremos 4 redes de tres variables A, B y C, correspondientes a un camino causal, camino evidencial, causa común y efecto común. Las

primeras 3 redes comparten el mismo conjunto de independencias, $(A \perp C \mid B)$, mientras que el grafo de la red de efecto común contiene una independencia que el resto no posee, $(A \perp C)$.

En general, se dice que dos grafos son equivalentes si:

1. los grafos no dirigidos asociados son iguales y
2. poseen las mismas v-estructuras.

Claramente, por la definición, dos grafos equivalentes deben poseer el mismo número de nodos. Además de que esta equivalencia genera una partición de los grafos de n nodos y sus respectivas clases de equivalencia.

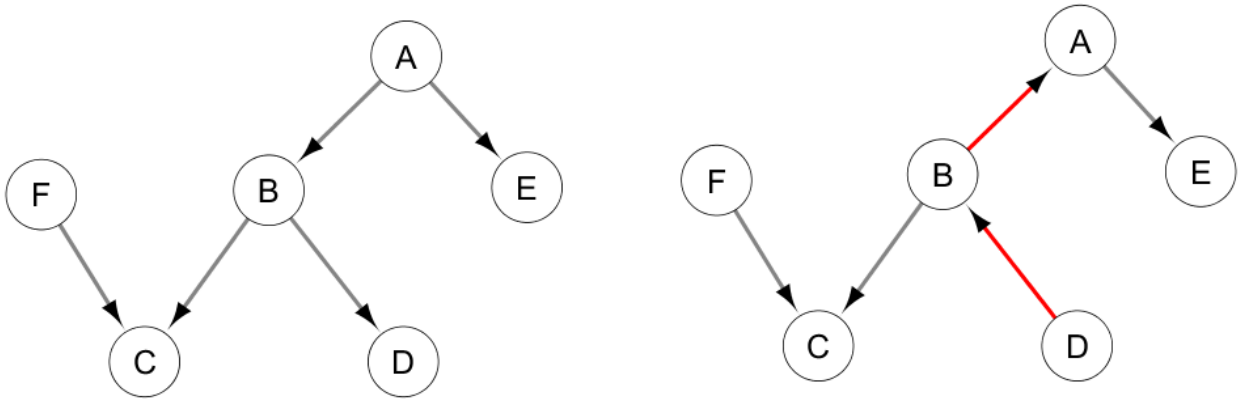


Figura 18: RB equivalentes

Se dice que una arista es irreversible si en cada uno de los grafos de la clase de equivalencia tiene la misma dirección. En otro caso, se denomina arista reversible.

Un ejemplo de grafos equivalentes, se muestra en la figura 18, dicha figura muestra dos grafos, los cuales difieren por dos aristas que tienen diferente dirección. Dichas aristas están marcadas de color rojo y como puede apreciarse, no implican, para el grafo izquierdo, ninguna v-estructura más o ninguna estructura menos que el grafo de la sección izquierda. Se concluye que los grafos son equivalentes, ya que, también tienen el mismo grafo no dirigido asociado.

Finalmente, si dos RB tienen grafos en la misma clase de equivalencia, se denominan redes bayesianas equivalentes. Si además, están definidas sobre la misma función de probabilidad P, entonces se denominan redes bayesianas probabilísticamente equivalentes.

4. Inferencia.

Una vez definida la RB sobre un espacio $\mathbf{X}$, se espera poder utilizar el conocimiento incluido en la red para obtener la probabilidad de una asignación $\mathbf{x}$. Sin embargo, en muchas ocasiones, no se cuenta con información de todas las variables y es justo para lo que los modelos están diseñados. Con base en un conjunto de datos, el modelo es capaz de inferir el comportamiento de otro conjunto de variables no observadas hasta ese momento.

Existen diversas técnicas para realizar inferencia, también conocida como propagación de evidencia (el conjunto de variables observadas), sobre una red bayesiana y un conjunto de datos. La idea intuitiva es conocer como cambian las probabilidades de cierto evento una vez que se conocen los estados de algunas variables. Cabe resaltar, que la calidad de los resultados de la inferencia dependerá en gran manera en la representación de la RB, ya que todos los algoritmos de propagación utilizan la estructura del grafo y las independencias emanadas de ella.

Los algoritmos de propagación de evidencia, pueden dividirse en dos grandes grupos. El primero de ellos abarca a los algoritmos exactos, que son aquellos en los cuales se realiza una serie de operaciones matemáticas para poder encontrar la probabilidad de un grupo de variables dado que se conoce otro grupo de variables. Mientras que el segundo grupo considera a los algoritmos aproximados, estos surgen a partir de simulaciones y se opta por ellos cuando la complejidad de un algoritmo exacto es elevada.

Todos los algoritmos buscan encontrar $P(\mathbf{X}|\mathbf{E})$, donde $\mathbf{X}$ y $\mathbf{E}$ son conjuntos disjuntos. $\mathbf{E}$ representará el conjunto de variables observadas (evidencia) y $\mathbf{X}$ las variables de interés.

En la práctica, el conjunto de variables observadas puede ser vacío, así que la expresión $P(\mathbf{X}|\mathbf{E})$ se reduce a marginalizar la función de distribución conjunta a la función de distribución de las variables de interés. Este último conjunto puede poseer una o más variables. Si su cardinalidad es 1, entonces hablaremos de la función de densidad marginalizada sobre una sola variable.

4.1. Algoritmos exactos.

Los algoritmos exactos se caracterizan por tener una serie de fórmulas que involucran operaciones algebraicas para el cálculo de las probabilidades dado un conjunto de evidencias.

Existen múltiples algoritmos exactos, los cuales fueron creados para diferentes tipos de RB. Muchos de ellos nacen a partir de los grafos que son árboles, ya que, este tipo de estructuras permiten una mejor implementación de los algoritmos, al menos computacionalmente.

Antes de analizar cada uno de los algoritmos, se puede analizar la tarea de encontrar la probabilidad $P(\mathbf{X}|\mathbf{E})$ a partir de la ecuación (2.1)

$$P(\mathbf{X}|\mathbf{E}) = \frac{P(\mathbf{X}, \mathbf{E})}{P(\mathbf{E})}$$

Dado que el denominador, $P(\mathbf{E})$ es una constante respecto a las variables de interés, el problema se puede reducir a encontrar el numerador y posteriormente una constante de normalización. En ese caso, si $\mathbf{Y}$ es el conjunto de variables aleatorias asociadas a los nodos de la RB, entonces:

$$P(\mathbf{X}|\mathbf{E}) = k * \sum_{y \setminus \{X,E\}} P(Y)$$

En el caso de que el conjunto de evidencias sea vacío, entonces solo se procede por marginalizar:

$$P(\mathbf{X}|\mathbf{E}) = \sum_{y \setminus X} P(Y)$$

Sin embargo, optar por esta metodología puede ser sumamente laborioso, ya que, dependiendo del número de variables en $\mathbf{Y}$ y las categorías de cada una de ellas, encontrar el valor de la suma especificada implicaría realizar bastantes operaciones. En caso de que la evidencia fuera nula y teniendo n variables con c categorías, entonces, la suma se reduciría a realizar $e^{n-|\mathbf{X}|} - 1$ sumas con cada una de las combinaciones de valores para la asignación de la función de distribución conjunta.

Además de que este tipo de razonamiento se pudo haber realizado desde un inicio sin la necesidad de la estructura de una RB, la cual posee información adicional sobre las relaciones que guardan cada una de las variables.

Para algunos algoritmos, será necesario definir el concepto de *factores* o *funciones potenciales*. Un factor es una función definida sobre el conjunto de posibles valores que pueden tomar ciertas variables aleatorias mapeado a un número real. $\phi(\mathbf{X}) : \mathbf{X} \to \mathbb{R}^+$. Al conjunto de variables sobre el cual está definido un factor se le conoce como *alcance del factor* y se denota por $alc[\phi]$.

La definición de marginalización en probabilidades, se puede extender de manera similar a factores y puede decirse que esta operación es conmutativa, es decir, si se marginaliza sobre una variable X y posteriormente se marginaliza sobre una variable Y, es lo mismo si primero se hubiese marginalizado sobre Y y después sobre X. Una vez que se marginaliza una variable, esta queda fuera del alcance del factor.

Mientras que la multiplicación de factores se realiza mediante la unión de sus alcances y devuelve un factor con el alcance especificado.

$$\phi_1(x)\phi_2(y) = \phi_3(z), \quad Z = X \cup Y$$

Dicha operación es conmutativa y asociativa.

4.1.1. Suma-producto

El primer algoritmo a analizar y tal vez el más conocido es llamado *suma-producto*. Este algoritmo busca realizar el proceso de marginalización tomando en cuenta las independencias asociadas al grafo y considerando cada uno de los términos en la regla de la cadena para redes bayesianas como factores, es decir, $P\left(X_i|Pa_{X_i}\right) = \phi_{X_i|_{Pa_{X_i}}} \forall X_i \in \mathbf{X}$. y se tiene que $alc\left[\phi_{X_i}\right] = X_i \cup Pa_{X_i}$

Para ejemplificar las bases de este algoritmo, suponga que se tiene el grafo $A \to B \to C$ y queremos calcular la función de distribución marginal de C:

$$P(C) = \sum_A \sum_B P(A, B, C) = \sum_A \sum_B P(A)P(B|A)P(C|A, B)$$

La estructura del grafo nos indica que $(C \perp A|B)$, por lo tanto:

$$P(C) = \sum_A \sum_B P(A)P(B|A)P(C|B)$$

Hasta aquí, se ha utilizado la estructura del grafo y sería suficiente para proceder a calcular la probabilidad solicitada mediante la fórmula mostrada, sin embargo, aún se puede reducir el número de operaciones a realizarse si se ocupan algunas propiedades algebraicas:

$$P(C) = \sum_A P(A) \sum_B P(B|A)P(C|B) = \sum_A \phi_A \sum_B \phi_{B|A}\phi_{C|B}$$

La última suma es el proceso de marginalización del producto de dos factores, por lo tanto, el factor resultante, $\psi_1(A, C)$, tendrá un alcance que solo involucre a las variables A, C.

$$\therefore P(C) = \sum_A \phi_A \psi_1(A, C)$$

$alc\left[\phi_A\psi_1(A, C)\right] = \{A, C\}$, sin embargo, al final se realiza el proceso de marginalización sl sumar sobre A y se obtiene un nuevo factor $\psi_2(C)$ cuyo alcance es solo la variable C.

El proceso de este algoritmo toma una serie de factores y un orden sobre el cual ir *eliminando* variables hasta llegar a marginalizar a las variables de interés. Para llegar a este objetivo, en cada etapa se calculan nuevos factores, correspondiente a multiplicar los factores originales que involucren cierta variable indicada en el ordenamiento y sumar sobre todos los valores de esa variable en el nuevo factor.

El ordenamiento de las variables para construir los nuevos factores no es único y así como en el ejemplo, primero se marginalizó sobre B, también se pudo comenzar por marginalizar por la variable A gracias a la conmutatividad y asociatividad de los factores.

Los ordenamientos que optimizan este algoritmo, son aquellos que implican factores con la menor cardinalidad en los alcances.

Hasta ahora, el proceso analizó el caso en el que la evidencia es nula. Este tipo de consulta, se considera como a *priori* pues no hay información acerca de ninguna variable. Este tipo de razonamiento es extendible de manera sencilla al caso en el que se conoce el valor que toman ciertas variables en la RB, este tipo de consultas se conocen como *posteriori*, pues se calcula la probabilidad una vez que ya se conoce evidencia.

El cálculo de las probabilidades posterioris se realiza aplicando la ecuación (2.1) y el proceso presentado para calcular las probabilidades *a priori*.

Sean $\mathbf{Y}$ las variables de una RB, $\mathbf{X}$ las variables de interés y $\mathbf{e}$ los valores de la evidencia, entonces, la probabilidad posteriori $P(\mathbf{X}|\mathbf{e})$ se calcula como:

$$P(\mathbf{X}|\mathbf{e}) = \frac{P(\mathbf{X}, \mathbf{e})}{P(\mathbf{e})}$$

El numerador de la última expresión equivale a marginalizar la probabilidad conjunta sobre todas las variables que no están en $\mathbf{X}$ ni en $\mathbf{e}$. Una vez obtenido esta cantidad, se procede a eliminar las variables $\mathbf{X}$ con el mismo proceso, obteniendo así el denominador.

Tanto en el numerador como en el denominador, los factores que contengan variables de evidencia en su alcance, solo se consideran si poseen una asignación en la que las variables de evidencia tomen el valor correspondiente del conjunto $\mathbf{e}$.

Este método se ejemplifica considerando el grafo $\mathcal{A}$. Para calcular la probabilidad de que un paciente fume o no dado que ha sido detectado con bronquitis, $P(F|B = si)$, se sigue el método descrito, además de considerar la ecuación (3.3).

$$P(F|B = si) = \frac{P(F, B = si)}{P(B = si)}$$

Donde el numerador se obtiene mediante $\sum_{\mathbf{X}\setminus(F,B)} \phi_F \phi_A \phi_{B|F} \phi_{C|F} \phi_{T|A} \phi_{P|C,T} \phi_{D|B,P} \phi_{X|P}$ y los factores que involucran a la variable observada, B, solo serán considerados si tienen la misma asignación que el valor observado en la evidencia, $B = si$. Mientras que el denominador se resume a $\sum_F P(F, B = si)$.

En general, este método de inferencia es aplicable a cualquier estructura. Sin embargo, este algoritmo, suele ser más eficiente en RB cuyos grafos tengan estructura de árbol o poliárbol.

Cuando no se presentan dichas estructuras, existen métodos denominados como *podar* nodos o aristas que reducen la estructura a un árbol o poliárbol, lo cual implica un menor número de operaciones que se deben realizar. Estos dos métodos solo dependerán de las variables de interés y de las variables ya observadas.

Para calcular la probabilidad $P(\mathbf{X}|\mathbf{e})$, se dice que un nodo hoja se puede *podar* si ese nodo no aparece en las variables de interés ($\mathbf{X}$) ni en la evidencia ($\mathbf{e}$) y simplemente no se considera ese nodo en el cálculo de la probabilidad.

La ventaja de este método es que es iterativo e implica la reducción de cálculos, por ejemplo, en la consulta mencionada anteriormente, $P(F|B = si)$, se puede podar todos los nodos hasta llegar a la estructura $F \rightarrow B$.

El método de *podar* aristas, elimina las dependencias de un árbol. Esto se hace cuando un nodo aparece en la evidencia, entonces, se eliminan todas las aristas que lleven a uno de sus hijos en el grafo original y el parámetro o factor asociado al nodo hijo se sustituye por aquel en el que la evidencia es verdadera sobre ese nodo. Esta operación implica considerar un menor número valores que pueda tomar cada factor. Dicho de otra manera, se sustituyen los nodos hijos de la evidencia por nodos no conectados a la evidencia en los cuales su parámetro considere el valor de las variables ya observadas.

Una vez aplicados estos dos métodos, se puede garantizar que el cálculo de la probabilidad $P(\mathbf{X}|\mathbf{e})$ sobre el nuevo grafo es igual a la probabilidad obtenida con el grafo original.

4.1.2. Propagación de probabilidad.

El algoritmo de Pearl, o de propagación de probabilidad solo es aplicable a las RB cuya estructura corresponda a un árbol o un poliárbol. A pesar de que se enfoca en solo cierto tipo de redes, este algoritmo es la base para el desarrollo de algunos más que son aplicables a cualquier tipo de red.

Recordando que en las estructuras de árbol y poliárbol solo existe un camino entre cualesquiera dos nodos, entonces, se deduce que al retirar cualquier nodo X y sus aristas de la estructura, quedan dos arboles o poliarboles no conexos. El primero corresponde a los padres de X y cualquier nodo que tenga un camino con X y pase por alguno de los padres de X. El segundo contiene a los hijos de X y cualquier nodo que en el grafo original tuviera un camino con X en el cual apareciera alguno de sus hijos. Por ejemplo, la figura 19 muestra las estructuras inducidas que se mencionaron si se considera el nodo B.

Así, cuando se quiere conocer la probabilidad de cierta variable X dada cierta evidencia $\mathbf{E}$ (variables ya observadas), la estructura permite separar el conjunto $\mathbf{E}$ en dos subconjuntos, a saber, $\mathbf{E}^+$ y $\mathbf{E}^-$, los cuales satisfacen:

- $\mathbf{E}^+$ es el subconjunto de la evidencia que se encuentra en el subgrafo inducido por los padres de X
- $\mathbf{E}^-$ es el subconjunto de la evidencia que se encuentra en el subgrafo inducido por los hijos de X
- $\mathbf{E} = \mathbf{E}^+ \cup \mathbf{E}^-$

Cabe resaltar que $\mathbf{E}^+, \mathbf{E}^-$ son subconjuntos de $\mathbf{E}$, por lo que sus valores también son conocidos.

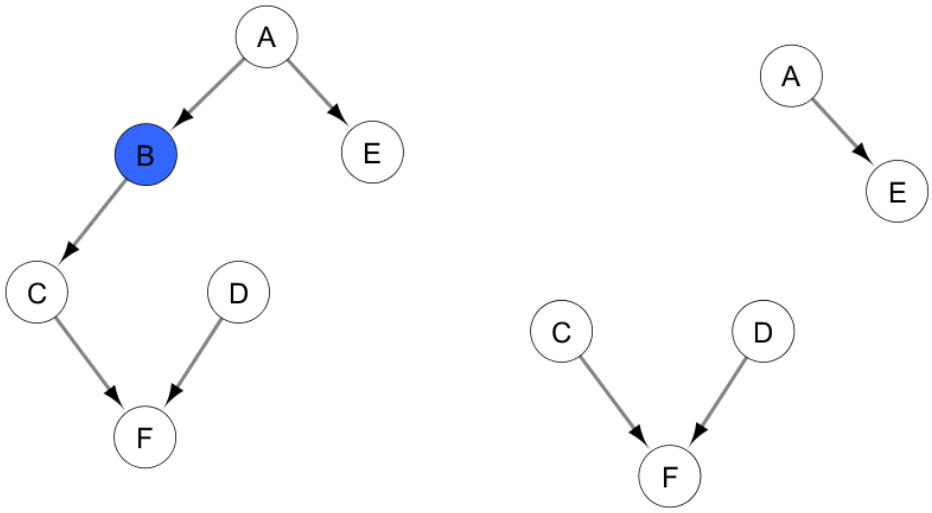


Figura 19: Árboles generados al retirar un nodo de un grafo con estructura de poliárbol.

A partir de esta división del conjunto evidencia, se puede observar que el conjunto $\mathbf{E}^+$ está separado de $\mathbf{E}^-$ cuando se conoce el valor de X.

La separación de estos dos conjuntos permite replantear la probabilidad de X dada la evidencia $\mathbf{E}$

$$\begin{aligned} P(X \mid \mathbf{E}) &= \frac{P(X,\mathbf{E})}{P(\mathbf{E})} = k * P(X,\mathbf{E}) \\ &= k * P(X)P(\mathbf{E} \mid X) \\ &= k * P(X)P(\mathbf{E}^+, \mathbf{E}^- \mid X) \\ &= k * P(X)P(\mathbf{E}^+|X)P(\mathbf{E}^- \mid X) \\ &= k * P(X,\mathbf{E}^+)P(\mathbf{E}^- \mid X) \\ &= k * \pi(X)\lambda(X) \end{aligned} \tag{4.1}$$

La ecuación (4.1), indica como la probabilidad posterior (no normalizada) de una variable X al conocer un conjunto de variables $\mathbf{E}$ se puede descomponer en el producto de un factor, $(\pi(X))$, que solo depende de la información que solo se puede añadir a través de los padres y otro factor, $(\lambda(X))$, que depende solo de la información que se puede recolectar a través de los hijos.

Para el cálculo de $\pi(X)$, consideremos que $Y = Pa_X = \{Y_1, \ldots, Y_n\}$, entonces, la evidencia la podemos dividir a lo más en n subconjuntos no vacíos, cada uno de ellos contiene a los nodos observados que son accesibles desde el $i-$ésimo padre. $\mathbf{E}^+ = E^+_{Y_1} \cup \ldots \cup E^+_{Y_n}$

Considerados estos conjuntos y que $y = y_1, \ldots, y_n$ es cada posible asignación de los padres de X, se tiene:

$$\begin{aligned}
\pi(X) &= P(X, \mathbf{E}^+) \\
&= \sum_y P(X, \mathbf{E}^+, y) \\
&= \sum_y P(X \mid \mathbf{E}^+, y) P(\mathbf{E}^+, y) \\
&= \sum_y P(X \mid \mathbf{E}^+, y) P(E^+_{Y_1}, \ldots, E^+_{Y_n}, y_1, \ldots y_n) \\
&= \sum_y P(X \mid \mathbf{E}^+, y) P(E^+_{Y_1}, y_1, \ldots, E^+_{Y_n}, y_n)
\end{aligned}$$

Debido al tipo de estructura que estamos considerando, los conjuntos $E^+_{Y_i}, y_i$ y $E^+_{Y_j}, y_j$, $i \neq j$, son independientes, ya que cualquier camino contiene al nodo X como v-estructura. Por lo tanto:

$$\begin{aligned}
\pi(X) &= \sum_y P(X \mid \mathbf{E}^+, y) P(E^+_{Y_1}, y_1) * \ldots * P(E^+_{Y_n}, y_n) \\
&= \sum_y P(X \mid \mathbf{E}^+, y) \prod_{i=1}^{n} P(E^+_{Y_i}, y_i)
\end{aligned}$$

Cada uno de los factores dentro del producto de la última igualdad se puede ver como $\pi_{Y_i X}(y_i)$, que mide la influencia de cada uno de los padres de X. En otras palabras, la probabilidad proveniente de los padres se puede descomponer en la información que aporta cada uno de los padres.

Mientras que para $\lambda(X)$ consideramos $Z = Ch_X = Z_1, \ldots, Z_m$ y de manera análoga $\mathbf{E}^- = E^-_{Z_1} \cup \ldots \cup E^-_{Z_m}$. Por lo tanto:

$$\begin{aligned}
\lambda(X) &= P(\mathbf{E}^- \mid X) \\
&= P(E^-_{Z_1}, \ldots, E^-_{Z_m} \mid X)
\end{aligned}$$

Una vez más, gracias a la estructura, $E^-_{Z_i}$ es independiente de $E^-_{Z_j}$ dado X , $i \neq j$.

$$\begin{aligned}
\lambda(X) &= P(E^-_{Z_1} \mid X) * \ldots * P(E^-_{Z_m} \mid X) \\
&= \prod_{i=1}^{m} P(E^-_{Z_i} \mid X)
\end{aligned}$$

Cada uno de los factores se considera como $\lambda_{Z_i X}(X)$, correspondiendo a la influencia que tiene cada uno de los hijos de la variable de interés sobre la misma.

De manera análoga, la información proveniente de los hijos se puede descomponer como la información proveniente de cada uno de ellos.

Por lo tanto, $P(X \mid \mathbf{E}) \propto (\sum_y P(X \mid \mathbf{E}^+, y) \prod_{i=1}^n \pi_{Y_i X}(y_i)) * (\prod_{i=1}^m \lambda_{Z_i X}(X))$.

La influencia que se mencionó en los factores $\pi_{YX}(y)$ y $\lambda_{ZX}(x)$ se refieren a los mensajes que pueden llegar de los nodos adyacentes a un nodo de interés, en este caso de sus padres e hijos, respectivamente. En general, para calcular la probabilidad posterior necesitamos todos los mensajes que llegan de los padres e hijos. Para esto se desarrolla un algoritmo que calcule todos los mensajes que se puedan mandar entre cualquier par de nodos adyacentes, para así calcular cualquier probabilidad $P(X \mid \mathbf{E})$.

Dos nodos adyacentes $A \to B$ comparten dos mensajes entre ellos, a saber, $\pi_{AB}(a)$ y $\lambda_{BA}(a)$, en donde a es la asignación de la variable A

Si se considera $C = \{C_1, \ldots, C_n\} = Ch_A \setminus B$ y $D = \{D_1, \ldots, D_m\} = Pa_B \setminus A$, los mensajes se pueden calcular aprovechando la descomposición de los conjuntos de evidencia y las independencias:

$$
\begin{aligned}
\pi_{AB}(a) &= P(A \cup E_A^+) \\
&= P(A \cup E_A^+ \cup_i E_{C_i}^+) \\
&= P(E_A^+ | A \cup_i E_{C_i}^+) P(A \cup_i E_{C_i}^+) \\
&= P(E_A^+ | A \cup_i E_{C_i}^+) P(\cup_i E_{C_i}^+ | A) P(A) \\
&= P(E_A^+ | A \cup_i E_{C_i}^+) P(A) P(\cup_i E_{C_i}^+ | A) \\
&= P(E_A^+ | A) P(A) P(\cup_i E_{C_i}^+ | A) \\
&\propto P(A | E_A^+) P(\cup_i E_{C_i}^+ | A) \\
&= P(A | E_A^+) \prod_i P(E_{C_i}^+ | A) \\
&= \pi(A) \prod_i \lambda_{C_i, A}(A)
\end{aligned}
$$

$$
\begin{aligned}
\lambda_{BA}(a) &= P(E_B^-|A) \\
&= \sum_{B,D} P(E_B^-, D, B|A) \\
&= \sum_{B,D} P(E_B^-, E_D^+, D, B|A) \\
&= \sum_{B,D} P(E_B^-|B, E_D^+, D, A) P(B|E_D^+, D, A) P(D, E_D^+|A) \\
&= \sum_{B,D} P(E_B^-|B) P(B|E_D^+, D, A) P(D, E_D^+|A) \\
&= \sum_{B,D} P(E_B^-|B) P(B|D, A) P(D, E_D^+) \\
&= \sum_{B} P(E_B^-|B) \sum_{D} P(B|Pa_B) P(D, E_D^+) \\
&= \sum_{B} \lambda(B) \sum_{D} p(B \mid Pa_B) \prod_{k=1}^{m} \pi_{D_k B}(d_k)
\end{aligned}
$$

Así, el proceso para calcular los mensajes que se envían en una RB, puede verse como un algoritmo iterativo, en el cual se necesita iniciar los cálculos con los nodos que no tienen hijos o no tienen padres. Para esto, cualquier nodo X que no tenga padres, tendrá $\pi(X) = P(X)$ y cualquier nodo Y que no tenga hijos tendrá $\lambda(Y) = 1$.

Posteriormente, se deberán calcular cada uno de los mensajes que se vayan requiriendo conforme se explore el árbol. Al calcular todos los mensajes dado un conjunto de evidencias, se garantiza que se calcularan todos las funciones $\pi(X)$ y $\lambda(X)$ de cualquier nodo en la red. Por lo tanto, se podrán calcular las probabilidades posteriores $P(X \mid \mathbf{E})$ para cualquier nodo.

Para un nodo arbitrario X, como el mostrado en la figura 20, los distintos mensajes que existen entre el nodo y sus nodos adyacentes son:

1. Los enviados de X a sus padres:
   - $\lambda_{XY_1}(y_1)$
   - $\lambda_{XY_2}(y_2)$
   - $\lambda_{XY_3}(y_3)$
   - $\lambda_{XY_4}(y_4)$
2. Los enviados de X a sus hijos:
   - $\pi_{XZ_1}(x)$
   - $\pi_{XZ_2}(x)$
   - $\pi_{XZ_3}(x)$

3. Los enviados de los padres de X a X:

   - $\pi_{Y_1X}(y_1)$
   - $\pi_{Y_2X}(y_2)$
   - $\pi_{Y_3X}(y_3)$
   - $\pi_{Y_4X}(y_4)$

4. Los enviados de los hijos de X a X:

   - $\lambda_{Z_1X}(x)$
   - $\lambda_{Z_2X}(x)$
   - $\lambda_{Z_3X}(x)$

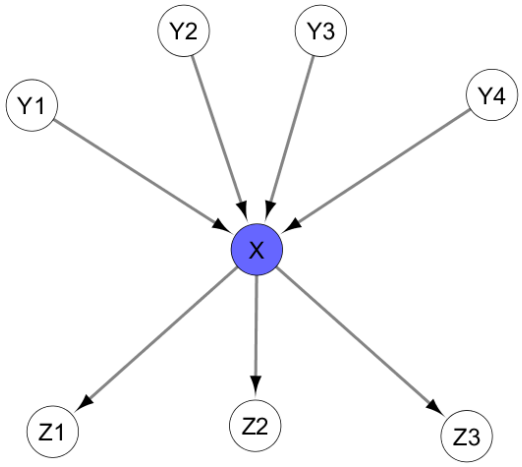


Figura 20: Mensajes propagados en el árbol.

4.1.3. Condicionamiento

El tercer algoritmo se utiliza en redes bayesianas cuya estructura no permita aplicar alguno de los dos algoritmos anteriores o cuando estos algoritmos no sean eficientes en las redes.

El caso más común, es cuando la estructura de la RB no corresponde a la de un árbol o poliárbol, entonces, el algoritmo anterior no es aplicable debido a que en algunos casos los nodos evidencia no separan la estructura en 2 árboles. Por lo tanto, lo que busca el algoritmos de condicionamiento es generar esta separación asignando valores a un conjunto de nodos con el cual pueda asegurarse que se divide la estructura en 2 no conectadas.

Al conjunto de nodos a los cuales se les asignará valores se le conoce como *conjunto de corte*. De aquí el algoritmo recibe su nombre, pues se condicionará sobre las variables del conjunto de corte, considerando todas las posibles asignaciones de las variables de corte.

Al asignar valores al conjunto de corte, las variables en ese conjunto podrían considerarse como evidencia, por lo tanto, se pueden podar algunas aristas que llevan a alguno de sus hijos, buscando así llegar a la estructura de un árbol y aplicar el algoritmo de propagación de probabilidad.

Lo anterior se ilustra en la figura 21, la cual muestra un grafo que no tiene estructura de árbol y como el considerar el nodo C permite que se transforme en una estructura de árbol.

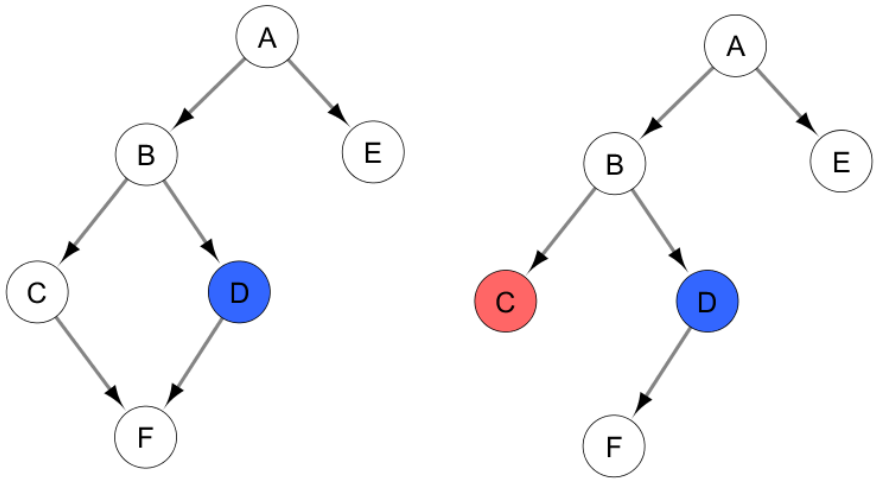


Figura 21: Grafo transformado a árbol a través de la asignación $C = c$ al conjunto de corte.

Una vez obtenida la estructura de poliárbol, se puede calcular la función de probabilidad condicionada a la evidencia E y al conjunto de corte C como:

$$\begin{aligned} P(X|E) &= \sum_c P(X|E,C)P(C|E) \\ &= \sum_c P(X|E,C)\frac{P(E|C)P(C)}{P(E)} \\ &\propto \sum_c P(X|E,C)P(E|C)P(C) \end{aligned}$$

Por lo tanto, el algoritmo se resume en calcular 3 probabilidades, las cuales pueden ser calculadas directamente de la estructura o utilizando el algoritmo de propagación de probabilidad.

La desventaja de este algoritmo es que no es tan eficiente cuando la cardinalidad del conjunto de corte comienza a aumentar y conforme el número de asignaciones c crezca.

4.1.4. Árbol unión.

El último algoritmo exacto que se presenta, parte de la idea de agrupar grupos de nodos para conseguir una nueva representación de la RB e inferir directamente de esta nueva representación.

La idea intuitiva de este algoritmo, es llevar la RB a una red de Markov (red no dirigida) cordal, en la cual se preserven las independencias de la red original. Para esto, es necesario moralizar y triangular el grafo, en ese orden.

Posterior a esto, se identifican los m conglomerados $\{C_1, \ldots, C_m\}$ de la nueva estructura que satisfagan la propiedad de intersección dinámica, con esto se busca expresar la función de probabilidad como el producto de m factores $\{\phi_1, \ldots, \phi_m\}$, donde el alcance del $i-ésimo$ factor es el $i-ésimo$ clique.

A cada uno de los nodos $X_1, \ldots, X_n$ se le asocia a un único clique en el cual esté contenido el conjunto $Fam_{X_i} = X_i \cup Pa(X_i)$. Además, se define la variable auxiliar A_i, la cual contiene a los nodos asociados al conglomerado i

La manera de definir estos factores (por lo tanto, la reexpresión de la función de probabilidad conjunta) no necesariamente es única, pues cada uno de las funciones potenciales dependen de los nodos asociados a cada conglomerado y puede darse el caso en el que en más de un conglomerado se encuentre la familia de cierto nodo.

Los factores se definen como:

$$\phi(c_i) = \begin{cases} \prod_{X_i \in A_i} P(X_i | Pa_{X_i}) & si \qquad A_i \neq \emptyset \\ 1 & e.o.c. \end{cases}$$

Bajo esta definición se satisface que $P(\mathbf{X}) = \prod_{i=1}^{m} \phi_i(c_i)$

Con los elementos que se poseen hasta este punto, se puede construir un grafo correspondiente a un árbol unión con el cual se pueda propagar la evidencia y realizar inferencia de los nodos de la red original con la información ya observada. Para esto, primero se debe construir el árbol unión.

Bajo la definición de árbol, solo debe existir un camino entre cualesquiera dos nodos y al ser un árbol unión, si dos vertices del árbol poseen nodos en común, entonces están contenidos en los vertices del camino que los une.

La idea intuitiva es calcular para cada conglomerado su función de probabilidad conjunta de los nodos de cada clique y a partir de esas funciones calcular las marginales de cada nodo o conjunto de nodos solicitado bajo el proceso de marginalización. Para esto, la misma idea del algoritmo de propagación de probabilidad funciona.

En el caso del árbol unión, el mensaje que se pasa de un nodo C_i a otro C_j se considera como una función del conjunto separador de esos nodos $S_{i,j}$, en otras palabras, es función de

los nodos X_k que tengan en común C_i y C_j.

Sea $\mathcal{B}$ el árbol unión asociado a la RB, consideremos el subgrafo que se genera al eliminar la arista que conecta a C_i y C_j y que además contiene al nodo C_i. Este subgrafo $\mathcal{B}_{ij}$ contiene toda la información que se propagara en el mensaje del nodo i al nodo j, dicho mensaje se puede denotar como $M_{ij}(s_{ij})$ y se calcula como:

$$M_{ij}(s_{ij}) = \sum_{(x \in \mathcal{B}_{ij}) \setminus S_{ij}} \prod_{c_k \in \mathcal{B}_{ij}} \phi_k(c_k)$$

Una vez calculados los mensajes, es posible calcular la función de probabilidad de cada uno de los nodos C_i. Esta se calcula como:

$$P(c_i) = \phi_i(c_i) \prod_{j \in Nb_{C_i}} M_{ji}(s_{ij})$$

Se observa que la probabilidad de un conglomerado solo puede calcular una vez que todos los mensajes de sus vecinos han sido calculados.

El desarrollo anterior puede considerarse que es el caso cuando la evidencia es nula. Sin embargo, el proceso considerando evidencia es exactamente el mismo salvo que se considera que alguna modificación en los factores para que estos *absorban* la evidencia.

Para un subconjunto de variables Z del grafo, decimos que una asignación $Z = z$ es consistente con la evidencia E si las variables de Z que esten contenidos en E son iguales al valor ya observado en la evidencia. Por ejemplo, si se tiene el conjunto de variables A, B de algún grafo y se tiene la evidencia $B = 1$, entonces, $(A, B) = (0, 1)$ es consistente con la evidencia, mientras que $(A, B) = (0, 0)$ no es consistente con la evidencia.

Cuando el conjunto de evidencia es distinto del vacío, es decir, $\mathbf{E} = \{E_1, \ldots, E_n\}$, entonces, para cualquier conglomerado C_i que contenga al menos un nodo que tambien se encuentre en la evidencia, su factor se modificará de acuerdo a la siguiente regla:

$$\phi'(c_i) = \begin{cases} \phi(c_i) & si \quad \text{la asignación } c_i \text{ es consistente con } \mathbf{E} \\ 0 & e.o.c. \end{cases}$$

Una vez que se redefinen los factores considerando la evidencia, el mismo proceso nos arrojará la probabilidad la asignación de cada uno de los conglomerados. Si se desea la función de distribución marginal posterior de un nodo X_i en particular, solo se debe marginalizar la función de probabilidad de algún conglomerado que contenga a dicha variable, optando por el de menor cardinalidad en caso de que la variable se encuentre en distintos cliques.

Vale la pena destacar que este algoritmo, a diferencia de los anteriores, nos puede dar la probabilidad conjunta de varios nodos una vez observada la evidencia, siempre y cuando los nodos se encuentren en un mismo conglomerado. Esta ventaja permite al usuario final realizar consultas más complejas que se adecuan a la necesidades reales.

4.2. Algoritmos aproximados.

Los algoritmos exactos tienen dos deficiencias notables. En primer lugar, algunos algoritmos no son aplicables a cualquier RB, sino que se dirigen exclusivamente a cierto tipo de estructuras. Además, los algoritmos más generales tienden a ser bastante complejos cuando el número de nodos se incrementa.

Los algoritmos aproximados se presentan como una posible solución a estos problemas. Estos, se basan, principalmente, en simular una muestra de la función de probabilidad conjunta y con base en esta muestra estimar los valores de las consultas cuando se posee cierta evidencia, a través de frecuencias entre las simulaciones que son consistentes con la asignación y el total de simulaciones.

La muestra debe ser suficientemente grande para que la aproximación sea precisa, es decir, la diferencia entre la aproximación y el valor real no difiera significativamente.

En este capítulo se presentan diversos algoritmos, los cuales se distinguen en cómo obtener la muestra con la que se realizará el proceso de inferencia.

La idea general es simular las observaciones de la distribución $P(\mathbf{X})$ (distribución real) a partir de una distribución que sea más sencilla de simular $H(\mathbf{X})$ (distribución simulada). De manera general, la distribución real puede verse como:

$$P(\mathbf{X}) = \frac{P(\mathbf{X})}{H(\mathbf{X})} H(\mathbf{X}),$$

donde $L(\mathbf{X}) = \dfrac{P(\mathbf{X})}{H(\mathbf{X})}$ se puede interpretar como el peso que se le asigna a $H(\mathbf{X})$ respecto a $P(\mathbf{X})$

Bajo esta consideración, una buena aproximación de $P(X|E), X \subseteq \mathbf{X}$ a partir de una muestra de n observaciones $\mathbf{X}_1, \ldots, \mathbf{X}_n$ de $H(X)$, esta dada por:

$$P(X|E) \simeq \frac{\sum_{i=1}^{n} L'(\mathbf{X}_i)}{\sum_{i=1}^{n} L(\mathbf{X}_i)}, \tag{4.2}$$

donde $L'(\mathbf{X}_i)$ es solo una función indicadora, es decir, es igual al peso $L(\mathbf{X}_i)$ asociado a la observación $i-$é$sima$, en caso de que la observación $\mathbf{X}_i$ sea consistente con X y E y es igual a 0 en caso contrario.

En el caso particular de las redes bayesianas, se puede aprovechar la estructura del grafo y si $P(\mathbf{X})$ factoriza de acuerdo a un grafo $\mathcal{G}$, entonces $P(\mathbf{X}) = \prod_{x_i \in \mathbf{X}} P(X_i|Pa_{X_i})$ y se puede encontrar una función $H(\mathbf{X})$ tal que $H(\mathbf{X}) = \prod_{x_i \in \mathbf{X}} H(X_i|Pa_{X_i})$

Lo anterior implica que la función de pesos se puede calcular como $L(\mathbf{X}) = \prod_{x_i \in \mathbf{X}} L(X_i|Pa_{X_i})$.

Con estas consideraciones, la simulación se puede realizar bajo un orden, proponiendo así, distribuciones simuladas más sencillas, que solo dependen de una variable y sus padres.

Cada método puede proponer una distribución $H(\mathbf{X})$ y/o forma de muestrear de esta distrbución. En las siguientes secciones se introducen posibles elecciones de la distribución en cuestión.

Con este esquema, una de las ventajas más notables de los algoritmos aproximados reside en que las consultas se pueden realizar sobre cualquier subconjunto de nodos, a diferencia de algunos algoritmos exactos.

4.2.1. Aceptación-rechazo

El algoritmo de aceptación-rechazo aprovecha la estructura de una RB proponiendo un orden **ancestral** (no necesariamente único) y con base en este se generan los valores de cada una de las variables para así obtener n observaciones.

Para una RB con j nodos $X_1, \ldots, X_j$, para obtener n observaciones $\mathbf{X}_1, \ldots, \mathbf{X}_n$, primero se asigna un orden ancestral. Para cada observación se simulan de acuerdo a este orden las variables mediante $P(X|Pa_X)$. Gracias al orden ancestral, cada variable solo será simulada una vez que se hayan simulado los valores de sus padres.

En cada una de las simulaciones de una variable X_i se rectificara la consistencia con la evidencia proporcionada, es decir, en caso de que $X_i \in E$, si el valor simulado no es igual al valor señalado por la evidencia ($x_i \neq e_i$), entonces se rechazan los valores hasta ese punto simulados y se vuelve a generar otra observación. En caso de que todos los valores simulados de los nodos evidenciales sean consistentes con la evidencia, entonces, se acepta esa observación y se procede a generar una nueva, hasta alcanzar las n observaciones deseadas.

Al haber elegido $H(\mathbf{X}) = P(\mathbf{X}) = \prod_{x_i \in \mathbf{X}} P(X_i|Pa_{X_i})$, los pesos $L(\mathbf{X})$ de cada observación serán iguales a 1, por lo que la estimación de la ecuación (4.2) pasa a ser el cociente de las observaciones que son consistentes con X entre el número de observaciones simuladas.

La principal desventaja de este tipo de simulación se encuentra cuando la categoría observada de algún nodo evidencial posee una probabilidad pequeña a comparación de las otras categorías, pues, al simular los valores se obtendrá con poca frecuencia el valor observado, lo que llevará a rechazar la muestra.

4.2.2. Muestreo Uniforme

El siguiente algoritmo a analizar presenta una variación significativa al método de aceptación-rechazo, la cual radica en la forma de elegir la función $H(\mathbf{X})$ y que no es necesario tener un orden ancestral para implementar el algorimo.

Para cada nodo se define la función:

$$H(X_i) = \begin{cases} \frac{1}{\#Ran(X_i)} & si \quad X_i \notin E \\ 1 & si \quad X_i \in E \text{ y } X_i = e_i \\ 0 & e.o.c. \end{cases}$$

Lo que hace esta distribución es aceptar todas las muestras pues al simular un nodo evidencial siempre se obtendrá el valor observado. Y aún más importante, esta distribución muestrea de manera uniforme a los nodos no evidenciales, otorgándoles la misma probabilidad a cada posible categoría.

Definiendo $H(\mathbf{X}) = \prod H(X_i)$ se tiene que el peso de cada observación es:

$$L(\mathbf{X}_j) = \frac{P(\mathbf{X}_j)}{\prod_{X_i \notin E} H(X_i)}$$

Dado que $\prod_{X_i \in E} H(X_i)$ es una constante para todas las observaciones, se puede considerar que $L(\mathbf{X}_j) \propto P(\mathbf{X}_j)$, por lo tanto se puede calcular de esta manera los pesos, realizando una normalización al final.

Posterior a esto el cálculo de la probabilidad deseada se obtiene mediante la ecuación (4.2).

El comportamiento de este algoritmo será bueno cuando no existan nodos con probabilidad marginal condicionadas a sus padres que asignen demasiada o poca probabilidad a alguna o algunas categorías. En caso contrario, los valores simulados no se asemejarán tanto a las observaciones reales esperadas por el conjunto de variables.

4.2.3. Función de verosimilitud pesante

El siguiente algoritmo surge tomando las ventajas de los dos algoritmos anteriores, para esto se define una función similar a la del método de muestreo uniforme.

$$H(X_i) = \begin{cases} P(X_i|Pa(X_i)) & si \quad X_i \notin E \\ 1 & si \quad X_i \in E \text{ y } X_i = e_i \\ 0 & e.o.c. \end{cases}$$

La función permite que todas las muestras simuladas sean aceptadas, pues los nodos evidenciales son los valores observados con probabilidad 1. Mientras que los nodos no evidenciales se simulan a partir de la distribución marginal condicionada a sus padres.

Debido al condicionamiento sobre los padres de un nodo, debe existir un orden ancestral para poder simular una observación, de manera similar al primer algoritmo presentado.

Definiendo $H(\mathbf{X}) = \prod H(X_i)$, los pesos de cada una de las observaciones son:

$$L(\mathbf{X}_j) = \prod_{X_i \in E} P(X_i|Pa(X_i))$$

Una vez obtenidos los pesos, la probabilidad deseada se calcula utilizando la ecuación (4.2).

4.2.4. MCMC

Otra forma de generar una muestra de la distribución $P(\mathbf{X})$ es mediante métodos de simulación estocástica conocidos como cadenas de Markov Monte Carlo, MCMC por sus siglas en inglés.

De manera muy general, estos métodos buscan simular una cadena de Markov, en la que cada uno de los estados es una observación de $\mathbf{X}$ y su distribución estacionaria es justo la distribución objetivo $P(\mathbf{X})$

En particular el método de Gibbs sampling o Gibbs sampler se adapta a las características de una RB.

El muestreo de Gibbs busca simular una observación de una distribución de la cual no es tan sencillo simular. Esto lo logra simulando a partir de distribuciones más sencillas. Para esto, en un vector aleatorio $\mathbf{X} = (X_1, \ldots, X_n)$ se utilizan las distribuciones condicionales $P(X_i|X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n) = P(X_i|\mathbf{X} \setminus X_i)$

A continuación se presentan los pasos que sigue Gibbs sampling.

1. Se inicializa el vector aleatorio con valores fijos arbitrarios. $\mathbf{X}_0 = (x_1, \ldots, x_n)$.
2. Para cada componente X_i de $\mathbf{X}$, se simula un nuevo valor a partir de la función $P(X_i|\mathbf{X}\setminus X_i)$. Al completar la simulación de las n componentes se guardara el valor del vector aleatorio como una observación.
3. Se realiza el paso 2 considerando el valor de la última observación hasta obtener el número de observaciones deseadas.

Cabe resaltar que en el paso 2, para una similar una componente con $P(X_i|\mathbf{X} \setminus X_i)$, el conjunto sobre el cual está condicionado, $\mathbf{X} \setminus X_i$, utiliza los valores de las variables hasta ese momento de la iteración, es decir, si se está simulando la observación $\mathbf{X}_j$ entonces para $X_1, \ldots, X_{i-1}$ se utilizan los valores de la observación $\mathbf{X}_j$ y para las variables $X_{i+1}, \ldots, X_n$ se utilizan los valores de la observación $\mathbf{X}_{j-1}$, pues aun no se han simulado en la actual iteración.

Este método depende, en gran medida, en que se pueda simular de manera más sencilla de las distribuciones $P(X_i|\mathbf{X} \setminus X_i) \forall i = 1, \ldots, n$. Para el caso de las RB, esta distribución se simplifica significativamente debido a las propiedades de independencia inmersas en la estructura.

Para cualquier RB con nodos $X_1, \ldots, X_n$ se satisface:

$$P(X_i|\mathbf{X} \setminus X_i) = \frac{P(\mathbf{X})}{P(\mathbf{X} \setminus X_i)} \tag{4.3}$$

$$\propto \prod_j P(X_j|Pa(X_i))$$
$$\propto P(X_i|Pa(X_i)) \prod_{Y \in Ch(X_i)} P(Y|Pa(Y))$$

La penúltima equivalencia es debido a que el denominador es una constante respecto a X_i y a la regla de la cadena para redes bayesianas. Mientras que la última es debido a que el resto de factores es constante respecto a X_i.

Con esta propiedad de la ecuación (4.3), la distribución de la que se simula cada uno de los nodos solo depende del nodo mismo, sus padres, sus hijos y los padres de sus hijos. Además, todas las distribuciones se especifican como parámetros de la RB.

Por lo tanto para emplear un algoritmo *Gibbs sampler* sobre una RB con nodos $\mathbf{X} = X_1, \ldots, X_n$ e inferir la probabilidad $P(Z|E),\ \ Z, E \in \mathbf{X},\ \ Z \cap E = \emptyset$ se procede con los siguientes pasos:

1. Se inicializa el vector aleatorio $\mathbf{X}_0$ con valores arbitrarios para las variables no evidenciales y que sea consistente con la evidencia E.

2. Para cada componente $X_i \in \mathbf{X} \setminus E$ se simula el valor de dicha variable a partir de la distribución normalizada de la ecuación (4.3). Al completar la simulación de las componentes no evidenciales se guardara el valor del vector aleatorio como una observación.

3. Se realiza el paso 2 considerando el valor de la última observación hasta obtener el número de observaciones deseadas.

Por último, para calcular la probabilidad del evento de interés, $P(Z|E)$, basta con calcular la proporción de observaciones que son consistentes con Z respecto al total de observaciones simuladas.

5. Aprendizaje.

La representación de una RB se especificó como una tarea usualmente ejecutada por un experto en la que el conocimiento del tema en cuestión es plasmado en la estructura (grafo) y parámetros (funciones de probabilidad condicionada) de la RB.

Sin embargo, en ocasiones la recopilación del conocimiento de un experto puede ser una tarea imposible debido a los recursos con los que se cuenta, además de que puede ser una tarea subjetiva en la cual no es posible medir la calidad del conocimiento transferido. En el peor de los casos ni siquiera exista un experto al cual se pueda recurrir.

Cuando se presenta esta situación se puede recurrir a estimar o *aprender* la representación de una RB a través de datos ya observados de las variables de interés. Es decir, las mismas observaciones muestran cual es el grafo y funciones de distribución condicionales adecuadas para las variables $X_1, \ldots, X_n$ consideradas en el problema.

Las diversas técnicas de aprendizaje a través de los datos se dividen de acuerdo al objetivo que persiguen, logrando así una segmentación entre las técnicas para aprender la estructura y aquellas para aprender los parámetros de la RB.

Estas técnicas dependen totalmente de los datos observados, por lo que se pueden emplear dos diferentes enfoques para realizar el proceso de aprendizaje. El primero de ellos es el de máxima verosimilitud, el cual busca otorgar como resultado la red de la cual sea más probable que hayan sido obtenidos los datos. Por otro lado, el enfoque bayesiano utiliza los datos como evidencia y a través de una distribución realiza inferencia bayesiana.

5.1. Paramétrico.

Los métodos de aprendizaje paramétrico asumen una estructura ya proporcionada y con base en los datos y el grafo se infieren las tablas de probabilidad.

Para los dos enfoques es importante resaltar que existen diferencias en las técnicas dependiendo si los datos proporcionados no contienen datos faltantes o no observados. Cuando los datos poseen registros donde todos sus valores son conocidos se conoce como datos completos, en otro caso son datos incompletos.

Para los datos incompletos se diferenciará dos casos. Cuando en algunos registros no se conoce el valor de alguna variable, en ese caso se conocen como valores no observados. Mientras que si para todos los registros no se conoce el valor de alguna variable, entonces, dicha variable es una variable latente u oculta.

5.1.1. Máxima verosimilitud datos completos.

Cuando se poseen datos completos la forma de estimar los parámetros es bastante sencilla. Dado un conjunto de datos D sobre un espacio $\mathbf{X}$ y un grafo $\mathcal{G}$ sobre el mismo espacio. Sean $d_1, \ldots, d_N$ cada una de las observaciones en el conjunto de datos, entonces, la probabilidad empírica de un evento $Y = y, Y \subseteq \mathbf{X}$ de acuerdo a los datos D es igual a:

$$P_D(Y) = \frac{\#\{d_i | X_j \in Y = y\}}{N}$$

Es decir, la probabilidad de un evento es igual a la proporción de observaciones consistentes con el evento $Y = y$, respecto al total de datos proporcionados.

Una vez definida la estimación de un evento, nos interesa calcular los parámetros de la RB, dados por $P(X_i|Pa(X_i))$. La forma de obtener estos valores se basará en la ecuación (2.1). Así que la estimación de la función de distribución condicionada empírica de un nodo está dada por:

$$P_D(X_i|Pa(X_i)) = \frac{P_D(X_i \cap Pa(X_i))}{P_D(Pa(X_i))}$$

De manera equivalente, se puede calcular como el número de observaciones que satisface la asignación del nodo y sus padres, entre el número de observaciones que satisfacen la asignación de los padres.

Esta estimación tiene sentido cuando las probabilidades de los padres de un nodo son mayores que cero. Sin embargo, como está definida la estimación, en algunos casos puede ser cero, por ejemplo, cuando no se poseen demasiadas observaciones o cuando el evento asignado a los padres es muy poco probable.

Este inconveniente se resuelve mediante el enfoque bayesiano. Pero cuando es posible realizar esta estimación, los valores obtenidos serán los que estimen la parte paramétrica de una RB, $\hat{P}(\mathbf{X}) = (P_D(X_1|Pa(X_1), \ldots, P_D(X_n|Pa(X_n)))$ y además son los que maximizan las verosimilitud $\prod_i P(d_i)$.

5.1.2. Algoritmo E-M

Cuando el conjunto de datos D es incompleto se puede emplear el algoritmo *expectation-maximization*, EM, el cual emplea la idea utilizada para los datos completos.

Para este algoritmo, se iniciara con una valor inicial $\theta^0 = (\theta^0(X_1, Pa(X_1)), \ldots, \theta^0(X_n, Pa(X_n)))$ de las distribuciones de los cada nodo condicionadas a sus respectivos padres.

La idea general de este algoritmo se resume en realizar iterativamente 2 pasos, el paso *expectation* y el paso *maximization*.

El paso *expectation*, se encarga de calcular un valor esperado de los parámetros, θ^k, con base al parámetro inicial o anterior y el conjunto de datos proporcionado.

El paso *maximization*, evalúa la calidad de la nueva asignación de parámetros. La calidad se basará en cual asignación es la que maximiza la verosimilitud, $V(\theta^k|D) = \prod_i P_{\theta^k}(d_i)$, donde $P_{\theta^k}(d_i)$ es la probabilidad de la $i-$ésima observación considerando tanto a la iteración θ^k como los parámetros de la RB.

Por la manera en que se define este algoritmo, se garantiza que la verosimilitud de la siguiente iteración no puede ser menor a la anterior, así que el algoritmo itera hasta que dos estimaciones consecutivas sean iguales o su verosimilitud no difiera tanto.

Para una RB con una estructura y una distribución inicial θ_0 el algoritmo EM para RB se muestra a continuación:

- **Paso E:** En el paso *expectation* se calculan los nuevos parámetros a través del valor de la iteración anterior (k) y los datos. Para esto primero se define la distribución empírica esperada de un evento Y como:

  $$P_{D,\theta_k}(Y) = \frac{1}{N} \sum_i P_{\theta_k}(Y|d_i)$$

  Donde N es el número de observaciones en el conjunto de datos.

  Esta función de distribución empírica depende de que tan probable es una asignación en cada observación de acuerdo al valor de θ_k. Y para el cálculo de cada sumando se requiere de algún método de inferencia presentado en el capítulo 4 o simplemente se asigna a 0 en caso de que no sea consistente el evento y la observación.

  Por lo tanto, para calcular el parámetro de un nodo en particular se tiene que:

  $$\theta^{k+1}(X_i, Pa(X_i)) = \frac{P_{D,\theta_k}(X_i \cap Pa(X_i))}{P_{D,\theta_k}(Pa(X_i))} = \frac{\sum_i P_{\theta_k}(X_i \cap Pa(X_i)|d_i)}{\sum_i P_{\theta_k}(Pa(X_i)|d_i)}$$

  Es conveniente tomar un algoritmo que calcule de manera sencilla las probabilidades $P_{\theta_k}(X_i \cap Pa(X_i)|d_i)$.

  La fórmula anterior se calcula para los n nodos y se obtiene la asignación para θ^{k+1}.

- **Paso M:** Se calcula la verosimilitud de θ^{k+1} y se compara con la verosimilitud de θ^k, si su diferencia es menor a ϵ, fijado con anterioridad, entonces se concluye el algoritmo. En otro caso, se vuelve al paso E.

La convergencia del algoritmo EM depende del valor inicial otorgado, por lo que es sugerido ejecutarlo con diferentes valores de inicio, los cuales seguramente arrojaran diferentes resultados y tomar aquél que su verosimilitud sea mayor.

5.1.3. Enfoque Bayesiano

Bajo la perspectiva bayesiana, se busca llevar el problema de aprendizaje paramétrico a un problema de inferencia. De manera similar al enfoque de máxima verosimilitud, se parte de una estructura $\mathcal{G}$ dada y un conjunto de datos $D = (d_1, \ldots, d_N)$, pero, en este enfoque se construye una nueva red y con base en esta se estiman los valores de los parámetros.

Para cada nodo X_i y una asignación de sus padres, $Pa_{X_i} = y$, se define un conjunto parametral como $\theta_{X_i,y} = (\theta_{X_{i,1},y}, \ldots, \theta_{X_{i,k},y})$. Donde cada $\theta_{X_{i,j},y}$ es igual a $P(X_i = j|Pa_{X_i} = y)$.

El número de elementos de un conjunto parametral depende del número de categorías de X_i y todos los elementos de un conjunto parametral deben sumar 1.

Por otra parte, para una variable X_i existirán tantos conjuntos parametrales como asignaciones posibles para el conjunto de sus padres, todos los conjuntos parametrales para una variable se denotaran por θ_{X_i}

Si se consideran todos los conjuntos parametrales para una variable X_i es posible dar la función de distribución condicionada a sus padres, $P(X_i|Pa(X_i))$, lo que es el objetivo del proceso de aprendizaje.

Se define Θ como todas aquellos conjuntos parametrales para las variables de $\mathbf{X}$. Se busca estimar Θ para tener todos los parámetros de una red bayesiana.

$$\Theta = (\theta_{X_1}, \ldots \theta_{X_n})$$

El enfoque bayesiano esta caracterizado por partir de un a distribución *a priori*, la cual refleja el conocimiento previo que se tiene y al observar los datos se actualizan las probabilidades, dando paso a la distribución posterior. El proceso de aprendizaje no es la excepción a este procedimiento, por lo que se requiere una distribución *a priori* de los conjuntos parametrales.

Por ejemplo, supongamos que se tienen en la red $A \to B$ y ambas son variables dicotómicas. Por lo tanto, para la variable B se requiere especificar 2 conjuntos parametrales, a saber, $\theta_{B,A=0}$ y $\theta_{B,A=1}$

Una posible selección de la distribución *a priori* para este par de conjuntos parametrales podría ser:

$\theta_{B,0}$	$P(\theta_{B,0})$
(0.1,0.9)	0.1
(.25,.75)	0.5
(0.5,0.5)	0.4

$\theta_{B,1}$	$P(\theta_{B,1})$
(0.75,0.25)	0.5
(0.5,0.5)	0.5

Figura 22: distribuciones *a priori*.

En la distribución para $\theta_{B,0}$ se ve como puede tomar 3 posibles valores con distintas probabilidades, mientras que $\theta_{B,1}$ asume 2 posibles valores de manera uniforme.

Con la distribución *a priori*, es posible calcular ahora la distribución posterior, es decir, la distribución del conjunto parametral una vez observados los datos, $P(\theta_{X_i,Pa(X_i)}|D)$. De acuerdo al enfoque bayesiano, esta se calcula como:

$$P(\theta_{X_i,Pa(X_i)=y}|D) \propto P(\theta_{X_i,y}) \prod_k \theta_{X_{i,k},y}^{\mathbb{I}(D,k,y)} \tag{5.1}$$

Para esta ecuación $\mathbb{I}(D,k,y)$ es el número de observaciones en D que son consistentes con la asignación y de los padres del nodo y la $k-ésima$ categoría de X_i.

La ecuación (5.1) considera dos componentes. La primera es la distribución *a priori* del conjunto parametral y la segunda es la verosimilitud de la asignación $\theta_{X_i,Pa(X_i)=y}$ respecto a los datos D.

Considerando el siguiente conjunto de datos D y las distribuciones *a priori* en la figura 22 se puede ejemplificar el cálculo de $P(\theta_{B,1}|D)$

A	B	...
1	0	...
1	1	...
1	0	...
0	0	...
1	1	...
0	0	...
1	1	...
1	0	...
1	0	...
0	1	...

$$\begin{aligned} P(\theta_{B,1} = (0.75, 0.25)|D) &\propto P(\theta_{B,1} = (0.75, 0.25)) * (.75)^{\mathbb{I}(D,1,1)} * (.25)^{\mathbb{I}(D,2,1)} \\ &= .5 * (.75)^4 * (.25)^3 \\ &= 0.002471924 \end{aligned}$$

$$\begin{aligned} P(\theta_{B,1} = (0.5, 0.5)|D) &\propto P(\theta_{B,1} = (0.5, 0.5)) * (.5)^{\mathbb{I}(D,1,1)} * (.5)^{\mathbb{I}(D,2,1)} \\ &= .5 * (.5)^4 * (.5)^3 \\ &= 0.00390625 \end{aligned}$$

Al normalizar estas probabilidades, se obtiene el valor exacto, arrojando:

$$\begin{aligned} P(\theta_{B,1} = (0.75, 0.25)|D) &= \frac{0.002471924}{0.002471924 + 0.00390625} = 0.3875598 \\ P(\theta_{B,1} = (0.5, 0.5)|D) &= \frac{0.00390625}{0.002471924 + 0.00390625} = 0.6124402 \end{aligned}$$

Lo cual puede interpretarse como, una vez observados los valores de los datos D la probabilidad del conjunto parametral se actualiza y ahora la asignación $(0.5, 0.5)$ es más probable.

Por útlimo, el valor que se asignará como parámetro para la RB, será el valor esperado de la distribución posterior, $P(\theta_{X_i,Pa(X_i)=y}|D)$:

$$\hat{\theta}_{X_i=x_i,Pa(X_i)=y} = \sum_{\theta_{x_i,Pa(X_i)=y}} \theta_{x_i,Pa(X_i)=y} P(\theta_{X_i,Pa(X_i)=y}|D)$$

Esto se realiza para todo los nodos y en conjunto se obtiene una estimación para Θ, teniendo así los parámetros de la RB.

Siguiendo el ejemplo mostrado en esta sección, el conjunto parametral estimado es:

$$\begin{aligned} \theta_{B=0,1} &= (0.75 * 0.3875598) + (0.5 * 0.6124402) \\ &= 0.59689 \end{aligned}$$

$$\begin{aligned} \theta_{B=1,1} &= (0.25 * 0.3875598) + (0.5 * 0.6124402) \\ &= 0.40311 \end{aligned}$$

Por lo que $\theta_{B,1} = (0.6, 0.4)$

La forma de calcular los parámetros funciona para datos completos. En caso de trabajar con datos incompletos basta con modificar $\mathbb{I}(D, k, y)$ con el valor esperado de observaciones que serán consistentes con esa asignación.

5.2. Estructural.

El aprendizaje de la estructura de una RB solo depende de un conjunto de datos. Similar al proceso de aprendizaje paramétrico se buscará una estructura que satisfaga algún criterio con el cual determinar cuál es la mejor selección de la estructura de acuerdo a las observaciones otorgadas.

5.2.1. Máxima verosimilitud

El enfoque de máxima verosimilitud busca una estructura la cual maximice la verosimilitud dados los datos, $V(\mathcal{G}|D)$. En general, se buscará la estructura cuyos parámetros estimados por máxima verosimilitud, maximicen una medida de calidad del grafo.

Existen diversos algoritmos con distintas variantes, pues algunos de ellos dependen de una medida con la cual evaluar la selección de la estructura.

El algoritmo aquí presentado ocupa una medida de la información conocida como *Información mutua*, $MI(X, Y|D, \theta)$, la cual calcula el nivel de dependencia entre las dos variables X y Y dado un conjunto de datos. La información mutua se define como:

$$MI(X, Y) = \sum_{x,y} P_D(X, Y) * log\left(\frac{P_D(X, Y)}{P_D(X)P_D(Y)}\right)$$

Donde $P_D()$ es la distribución empírica de acuerdo a los datos, es decir, la frecuencia de observaciones consistentes con la asignación respecto al total de datos.

Además, para esta medida se asume que $0log(0) = 0$.

Definida la información mutua, se puede comenzar a evaluar las estructuras correspondientes a un árbol. Definiendo el *score* $tscore(\mathcal{G}|D) = \sum_{X_i} MI(X_i, Pa(X_i))$ se tiene que la estructura $\mathcal{G}$ que posea el mayor $tscore(\mathcal{G}|D)$ es la misma estructura que maximiza la verosimilitud sobre todos los posibles árboles que se puedan generar sobre $\mathbf{X}$

Para generar el árbol $\mathcal{G}$ con el mayor *tscore* se procede a crear el grafo no dirigido $\mathcal{H}$ que se caracteriza por ser el grafo completo en el que cada arista posee un peso correspondiente a la información mutua de los dos nodos que una. Posterior a esto, se encuentra el árbol de peso máximo el cual será la versión no dirigida del árbol cuya verosimilitud es la máxima.

Por lo tanto, se debe calcular la información mutua de todos los nodos para encontrar una estructura de árbol con la mayor verosimilitud. Por otro lado, el algoritmo para encontrar un árbol de peso máximo es relativamente sencillo. Se comienza con todos los nodos sin aristas y se van agregando aquellas que tengan mayor peso en cada paso, verificando antes de agregar una arista candidata que la estructura siga siendo un árbol después de agregar a dicha arista. Esta es la idea intuitiva del algoritmo Kruskal para encontrar el árbol de peso máximo

Una primera aproximación a una estructura valida, es elegir algún nodo del árbol de peso mínimo como nodo raíz y las direcciones de las aristas corresponden a aquellas que se alejan del nodo raíz.

Hasta ahora, solo se ha construido un árbol, pero puede que la *mejor* elección de estructura no corresponda a un árbol. En este punto es importante recalcar que la calidad de una red solo se basa en la verosimilitud del grafo. Sin embargo, se puede demostrar que si se tiene

un grafo $\mathcal{G}$, entonces el grafo $\mathcal{G}^*$ correspondiente a agregar al menos una arista tiene una verosimilitud mayor que el grafo $\mathcal{G}$, por lo que se deduce que el grafo de mayor verosimilitud es un grafo completo, el cual ni siquiera podría ser un DAG y, aún más importante, no persigue el objetivo principal de una RB, el cual es representar de manera *sencilla* la distribución de $\mathbf{X}$ aprovechando las independencias.

Por esto último, se presenta una nueva medida para evaluar la calidad del grafo asociado a una RB. Primero definimos la dimensión de un grafo $\mathcal{G}$, $Dim(\mathcal{G})$ como el número de parámetros libres necesarios para definir la distribución de $\mathbf{X}$, formalmente:

$$Dim(\mathcal{G}) = \sum_i (\#Ran(X_i) - 1) * \#Ran(Pa_{X_i})$$

Para un nodo sin padres, asumimos que el número de categorías de sus padres, $\#Ran(Pa_{X_i})$, es 1.

La nueva medida para evaluar la estructura sigue considerando la verosimilitud, tomando su logaritmo, pues alcanza el máximo en el mismo punto que lo hace la verosimilitud. Además, incorpora un término de penalización para grafos complejos, es decir, da preferencia a grafos en los que sea más sencillo definir los parámetros de la RB.

La medida se define como:

$$Score(\mathcal{G}|D) = log(V(\mathcal{G}|D)) - (\psi(N) * Dim(\mathcal{G})) \tag{5.2}$$

El término $\psi(N) * Dim(\mathcal{G})$ es la penalización, la cual da un peso de $\psi(N)$ a la dimensión del grafo. $\psi(N)$ es una función del número de observaciones dadas en los datos, N.

Cuando el peso $\psi(N)$ es una constante respecto a N, la medida se le conoce como criterio de información de Akaike, AIC por sus siglas en inglés.

Si el peso $\psi(N)$ es igual a $\frac{log_2 N}{2}$, entonces la medida adopta el nombre de criterio de información bayesiana, BIC por sus siglas en inglés.

La idea general de los algoritmos de busqueda de una estructura con la mayor calidad es ir modificando una estructura inicial mediante 3 operaciones sobre aristas e ir evaluando el nuevo grafo, prefiriendo aquellos que maximicen la medida seleccionada.

Las 3 operaciones son agregar una arista entre dos nodos, eliminar una arista y revertir la dirección de una arista. Estas 3 siendo válidas cuando la estructura siga siendo un DAG posterior a la operación.

Ahora, la medida *Score* se puede descomponer, de manera que pueda ser expresada con sumandos, cada uno en función de una arista, logrando así calcular el *Score* de dos grafos que difieran por una operación de manera sencilla.

La log-verosimilitud se puede reexpresar en términos de la entropía, la cual mide la incertidumbre de una variable:

$$log(V(\mathcal{G}|D)) = -N \sum_{X_i} ENT(X_i|Pa(X_i))$$

Donde la entropía se calcula como $ENT(X|Y) = -\sum_x P_D(X,Y) log_2(P_D(X|Y))$

Por lo tanto el *Score* se redefine como:

$$Score(\mathcal{G}|D) = \sum_{X_i}(-N * ENT(X_i|Pa(X_i))) - (\psi(N) * (\#Ran(X_i) - 1) * \#Ran(Pa_{X_i})$$

Cada operación cambia la entropía y el número de parámetros libres de al menos un sumando, pues algún conjunto de padres se ve alterado. Pero con esta reexpresión del *Score* basta con recalcular los términos donde el conjunto de padres se vio afectado, pues el resto de sumandos permanecen iguales.

5.2.2. Enfoque Bayesiano

El enfoque bayesiano utiliza ideas similares al enfoque de máxima verosimilitud. Variando, principalmente, en como se define el *score*, pues esté será la probabilidad posterior de un grafo dado un conjunto de datos, $P(\mathcal{G}|D)$.

Se busca el grafo $\mathcal{G}$ con el mayor *score* bayesiano, $BS()$, el cual, aprovechando la constante de proporcionalidad, se define como:

$$BScore(\mathcal{G}|D) = P(\mathcal{G})P(D|\mathcal{G})$$

Las dos componentes del *score* bayesiano se pueden expresar de la siguiente manera:

$$P(D|G) = \prod_{k=1}^{N} P_\Theta(d_k) = \prod_{X_i} \prod_{k=1}^{N} P_\Theta(X_i|Pa(X_i), d_k)$$

$$P(G) = \prod_{X_i} \prod_{X_i \in Pa(X_i)} p_{j,i} \prod_{X_i \notin Pa(X_i)} (1 - p_{j,i})$$

En donde $p_{j,i}$ es la probabilidad de que el nodo X_j sea padre del nodo X_i. Esta es una probabilidad *a priori* que tiene que ser especificada para el proceso de aprendizaje. Usualmente, si se carece de la experiencia necesaria para definir las probabilidades o no existe alguna relación de dependencia clara entre las variables, entonces, la distribución *a priori* es uniforme sobre todas los nodos.

Similar al enfoque de máxima verosimilitud se busca que la medida sea fácil de recalcular tras una de las operaciones de agregar, remover y revertir la dirección. Así que se toma el logaritmo:

$$
\begin{aligned}
log(BScore(\mathcal{G}|D) &= log(\prod_{X_i} \prod_{X_i \in Pa(X_i)} p_{j,i} \prod_{X_i \notin Pa(X_i)} (1 - p_{j,i})) + log(\prod_{X_i} \prod_{k=1}^{N} P_\Theta(X_i|Pa(X_i), d_k)) \\
&= \sum_{X_i} log(\prod_{X_i \in Pa(X_i)} p_{j,i} \prod_{X_i \notin Pa(X_i)} (1 - p_{j,i})) + \sum_{X_i} log(\prod_{k=1}^{N} P_\Theta(X_i|Pa(X_i), d_k)) \\
&= \sum_{X_i} log(\prod_{X_i \in Pa(X_i)} p_{j,i} \prod_{X_i \notin Pa(X_i)} (1 - p_{j,i})) + log(\prod_{k=1}^{N} P_\Theta(X_i|Pa(X_i), d_k))
\end{aligned}
$$

Al modificar un grafo con una sola operación, basta con calcular los sumandos en los que está involucrado el cambio en la arista. El resto de los componentes permanecen iguales, por lo que es sencillo recalcular el *score* bayesiano.

Esta definición permite que la búsqueda del grafo se pueda realizar de manera similar al enfoque de máxima verosimilitud.

5.2.3. Hill-climbing

Hasta ahora se han definido algunos *scores* y de manera general se explicó como se realiza la búsqueda de la mejor estructura a través de las operaciones sobre aristas.

En este apartado se detalla uno de los múltiples algoritmos de búsqueda que existen, especificando cada uno de sus pasos y modificaciones para su mejor desempeño.

El algoritmo se conoce como hill-climbing o de escalada, es un algoritmo de búsqueda local, el cuál considera solo a los vecinos de un nodo, las operaciones sobre aristas y un *score* para determinar la mejor estructura para un conjunto de observaciones D

El algoritmo requiere un grafo inicial $\mathcal{G}^0$ y a partir de este considera la mejor operación para llegar a la siguiente iteración. En ocasiones se opta por iniciar con el grafo vacío, aquél que no contiene aristas entre los nodos.

A continuación se detallan los pasos a seguir en este algoritmo:

1. **Inicialización:** Se determina el *score* S con el cual se evaluará, el conjunto de datos D y el grafo inicial $\mathcal{G} = \mathcal{G}^0$.

2. **Agregar:** Para cada nodo X_i y para cada nodo X_j que no sea su padre, si agregar la arista $X_j \to X_i$ no genera un ciclo dirigido, entonces se determina cuál de estas adiciones genera el mayor *score* y se guardan los siguientes valores $C^a_{j,i} = (\mathcal{G}^a_{j,i}, S(\mathcal{G}^a_{j,i}))$, que puede ser interpretado como el cambio con la operación agregar con los nodos i, j

3. **Eliminar:** Para cada nodo X_i y para cada nodo X_j que sea su padre, se determina cuál grafo tiene mayor *score* si se elimina la arista $X_j \to X_i$ y se guardan los siguientes

valores $C^e_{j,i} = (\mathcal{G}^e_{j,i}, S(\mathcal{G}^e_{j,i}))$, que puede ser interpretado como el cambio con la operación eliminar con los nodos i, j

4. **Revertir:** Para cada nodo X_i y para cada nodo X_j que sea su padre, si revertir la arista $X_j \rightarrow X_i$ no genera un ciclo dirigido, entonces se determina cual grafo tiene el mayor *score* tras revertir una arista y se guardan los siguientes valores $C^r_{j,i} = (\mathcal{G}^r_{j,i}, S(\mathcal{G}^r_{j,i}))$, que puede ser interpretado como el cambio con la operación revertir con los nodos i, j

5. **Cambio máximo:** Se determina cuál de las 3 operaciones tiene el mejor *score*, esto es, $sm = max(S(\mathcal{G}^a_{j,i}), S(\mathcal{G}^e_{j,i}), S(\mathcal{G}^r_{j,i}))$ y se guarda el grafo asociado a sm en la variable $\mathcal{G}$

6. **Comparación:** Si $S(\mathcal{G}) - sm < 0$, quiere decir que el grafo al iniciar la iteración era mejor que cualquier otro bajo una operación, por lo que se termina el algoritmo y se devuelve el grafo $\mathcal{G}$ y su *score*. En otro caso, quiere decir que se encontró un mejor grafo, por lo que se asigna $\mathcal{G} = \mathcal{G}*$ y se vuelve a iterar, comenzando en el paso 2.

Es importante resaltar que el algoritmo hill-climbing no garantiza obtener un máximo global. Solo se puede asegurar que el resultado es un máximo local.

Se puede mejorar el algoritmo ejecutándolo en varias ocasiones pero con diferente punto de inicio, es decir, variando $\mathcal{G}^0$ en cada ejecución. Esta técnica se conoce como reinicios aleatorios y el resultado será la mejor ejecución realizada.

Otro algoritmo, conocido como tabú, realiza algo similar, pero cada que se vuelve a ejecutar el algoritmo, se agrega la restricción de no considerar a los grafos que sean resultado de ejecuciones anteriores, lo que garantiza que cada ejecución llegue a un grafo diferente, a diferencia de los reinicios aleatorios.

Además, este algoritmo puede agregar restricciones sobre el grafo, que repercutan en que operaciones son posibles. Hasta el momento, el algoritmo solo realiza las operaciones entre 2 nodos si no generan un ciclo, pero se puede agregar restricciones sobre que aristas son obligatorias, que aristas son prohibidas, que direcciones son o no posibles, el número de padres máximo para un nodo, entre otros.

Así que, a la hora de considerar una operación, también se debe hacer la validación que el nuevo grafo satisfaga las restricciones.

Las restricciones, se emplean cuando hay una relación que sea muy marcada y se desee en el resultado final y ayudan a obtener grafos que modelen el problema de acuerdo a la realidad.

Además del *score*, que evalúa la red de manera general, existen medidas para determinar lo significativas que pueden ser algunas características de la red, por ejemplo, la presencia de un arco e incluso su dirección.

Para esto, se utilizan métodos de remuestreo, particularmente *Bootstrap*, el cual genera una nueva muestra a partir de los datos de entrada y emplea un método de aprendizaje con

esta nueva muestra. Al final, la *fuerza del arco* o la *fuerza de la dirección* será igual a la frecuencia con la que aparece dicha característica después de generar m nuevas muestras y aplicar el método de aprendizaje.

Esta nueva medida indica si los datos realmente sugieren la dirección o el arco, o si solo apareció en la red para satisfacer las restricciones del algoritmo de aprendizaje o al azar. Al ser una frecuencia, valores cercanos a 1 significan que la dirección o arco deben ser incluidos.

Para las direcciones, valores cercanos a 0.5 muestran que la dirección podría ser revertida, verificando primero que no se genere un ciclo al aplicar esta operación. También al revertir puede o no permanecer el mismo conjunto de independencias inducido por el grafo, lo cual dependerá de si los dos grafos son equivalentes, en este caso, vale la pena considerar si la o las independencias que se puedan generar o desaparecer al revertir la dirección tienen sentido. Por lo que en la práctica se opta por considerar direcciones y arcos cuya fuerza sea mayor a 0.5

Para finalizar este capítulo, cabe resaltar que, aunque se optó por el aprendizaje cuando conseguir el conocimiento de un experto no era una tarea sencilla, se recomienda recurrir a un experto para validar la coherencia de la red. Principalmente deben aprobarse las dependencias entre las variables del modelo e incluso las variables mismas.

Esta última tarea debe distinguirse de la transferencia de conocimiento, la cual es más costosa. Aprobar la red, sin duda, es una tarea que resulta más sencilla, pero se llega a ella a través del proceso de aprendizaje.

6. Caso de estudio.

Con el objetivo de ejemplificar más a detalle las técnicas hasta ahora presentadas, se hará uso de un conjunto de datos con el cual se modelará una RB enfocada al problema de clasificación binaria.

La base de datos seleccionada, proviene del Instituto Nacional de Diabetes y Enfermedades Digestivas y Renales, en Estados Unidos. Cada registro contiene las características relevantes de un paciente para poder determinar la presencia o ausencia de diabetes tipo 2.

Los datos fueron recolectados de la población femenina mayor de 21 años de un grupo tribu de nativos norteamericanos conocidos como Pima, con el objetivo de determinar las causas del alto índice de prevalencia de este padecimiento en este grupo.

La diabetes mellitus tipo 2 es un padecimiento muy relevante para la salud pública, ya que, en los últimos años representa una de las principales enfermedades, en cuanto a tasas de morbilidad y mortalidad se refiere.

La patología se ocasiona cuando el cuerpo humano no procesa o produce la insulina de manera correcta, provocando altos niveles de glucosa en la sangre.

La diabetes es una enfermedad considerada crónico-degenerativa, pues desemboca en problemas renales, cardíacos e incluso provoca el fallecimiento del paciente.

En México, actualmente, representa la primera causa de muerte, tanto en hombres como en mujeres y su prevalencia se relaciona principalmente a la alimentación, sedentarismo y antecedentes familiares.

Una detección temprana de este padecimiento, permitiría implementar tratamientos adecuados y oportunos que mejoren la calidad de vida del paciente y eviten las complicaciones secundarias que emanan del padecimiento. Además, esta detección temprana podría mitigar costos monetarios, materiales, humanos y de infraestructura asociados a dicha patología.

Se seleccionó una base de datos con esta temática debido a que es de interés social y su relevancia radica en poder llegar a un modelo con el cual se pueda mejorar la detección o al menos obtener resultados similares a los que tienen modelos en la actualidad.

6.1. Datos.

La base de datos contiene 768 registros en los que se detallan 9 variables. La variable Diabetes se considera como la variable objetivo y las 8 restantes son las características particulares de cada paciente.

A continuación, se presenta la descripción de cada una de estas variables:

- **Embarazo:** Variable numérica que representa el número de ocasiones en la que un paciente ha estado embarazada.
- **Glucosa:** Variable numérica que mide la concentración de la glucosa en plasma a partir de una prueba oral de tolerancia a la glucosa. La unidad de medición en miligramos por decilitro, mg/dL
- **Presión:** Variable numérica que mide la presión arterial diastolica. La unidad de medición es milímetros de mercurio, (mmHg)
- **Tricep:** Variable numérica que mide el espesor del pliegue cutáneo ubicado sobre el músculo trícep medido en milimetros (mm).
- **Insulina:** Variable numérica que mide el nivel de insulina sérica del paciente, medido en microunidades por mililitro, muU/mL.
- **IMC:** Variable numérica que mide el índice de masa corporal.
- **Pedigree:** Variable numérica que mide la función de pedigree de diabetes, la cual indica la tendencia a padecer diabetes con base a la información familiar. Puede ser considerada como una función de carga genética
- **Edad:** Variable numérica que mide la edad de un paciente en años.
- **Diabetes:** Variable categórica que indica la presencia (1) o ausencia (0) de diabetes mellitus tipo 2.

Se procede a realizar un breve análisis exploratorio de los datos.

Primero, se puede notar que no existen $NA's$. Sin embargo, existen valores 0 en las variables glucosa, presión, tricep, insulina e IMC. Estos valores no son consistentes con alguna posible medición realizada, por lo que se opta por considerarlos como valores faltantes.

Para implementar la RB multinominal se categorizarán las 8 variables numéricas.

La variable embarazo tendrá 3 categorías De cero a un embarazos (0), 2 a 4 embarazos (1) y más de 4 embarazos (2).

La variable glucosa tendrá 2 niveles. Menor a 140 se considera normal (0) y más de 140 es alto (1).

Para la presión se consideran 3 niveles. Menor a 60 es baja (0), de 60 a 90 normal (1) y más de 90 es alta (2).

La variable Tricep se dividirá en 2 categorías. Menor a 25 se considera normal (0) y en otro caso es alto (1.)

Para la Insulina, se consideran 2 niveles, normal en caso de ser menor o igual a 120 (0) y alto en otro caso (1).

El IMC se divide en 3 clases, se considera normal si es menor o igual a 25 (0), sobrepeso de 25 a 30 (1) y es obesidad si es mayor a 30 (2).

La función de pedigree se segmentará en tres clases, menor o igual .24 (0), entre .24 y .37 (1) y mayor a .37 (2).

Los pacientes serán clasificados en 2 grupos de edad. De 21 a 41 años (0) y de 42 años en adelante (1).

Una vez asignados los nuevos valores, se puede analizar la correlación de las 9 variables por pares. La figura 23 muestra un gráfico, del tipo mapa de calor, en el que se resume la correlación de las variables. En este, la intensidad del color y el tamaño del cuadro refleja que tan correlacionadas están dos variables, siendo las variables Tricep e IMC las que muestran mayor correlación.

Considerando la variable objetivo, glucosa e insulina son las variables con mayor correlación.

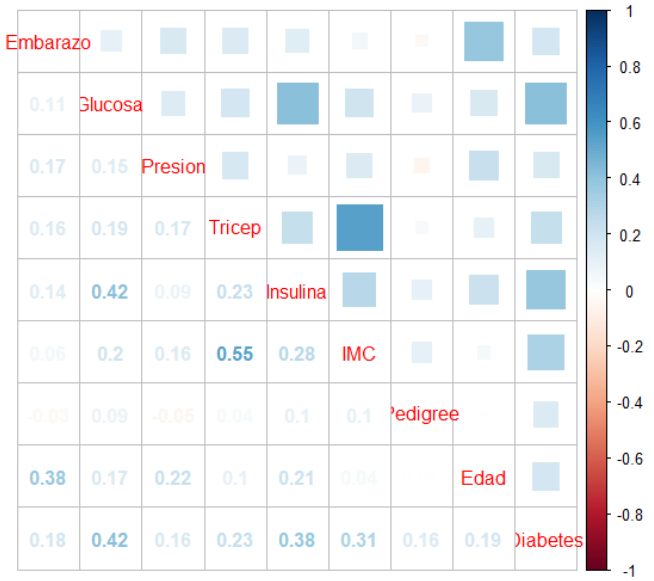


Figura 23: Correlación entre las variables.

6.2. Representación y aprendizaje.

La representación será modelada con técnicas de aprendizaje, es decir, los datos mostrarán la estructura y parámetros con los cuales se pueda modelar la red bayesiana. Comenzando

con un modelo sin ninguna restricción y a partir de este se agregan las opiniones de expertos acerca de las relaciones que guardan las variables.

En el software estadístico *R* existe una paquetería llamada *bnlearn*, en la cual se encuentran implementados los algoritmos de aprendizaje e inferencia mostrados en este texto.

Para iniciar el proceso de aprendizaje, seleccionaremos un subconjunto de observaciones (conjunto de entrenamiento) para implementar el modelo. Posteriormente, las observaciones no seleccionadas (conjunto de prueba) se utilizan para medir la calidad del modelo obtenido mediante el conjunto de entrenamiento.

En este caso la proporción de los conjuntos de entrenamiento y prueba son 80 % (614) y 20 % (154). La selección del conjunto de entrenamiento se realiza mediante muestreo aleatorio simple con 27 como la semilla para el muestreo.

Al contar con datos faltantes en el conjunto de entrenamiento, se opta por utilizar el algoritmo E-M para el aprendizaje estructural con 100 inicios aleatorios y el algoritmo Hill-Climbing con el enfoque de verosimilitud y el BIC como *score*.

Un primer acercamiento se realiza sin ninguna restricción a la red, obteniendo como resultado el grafo mostrado en la figura 24

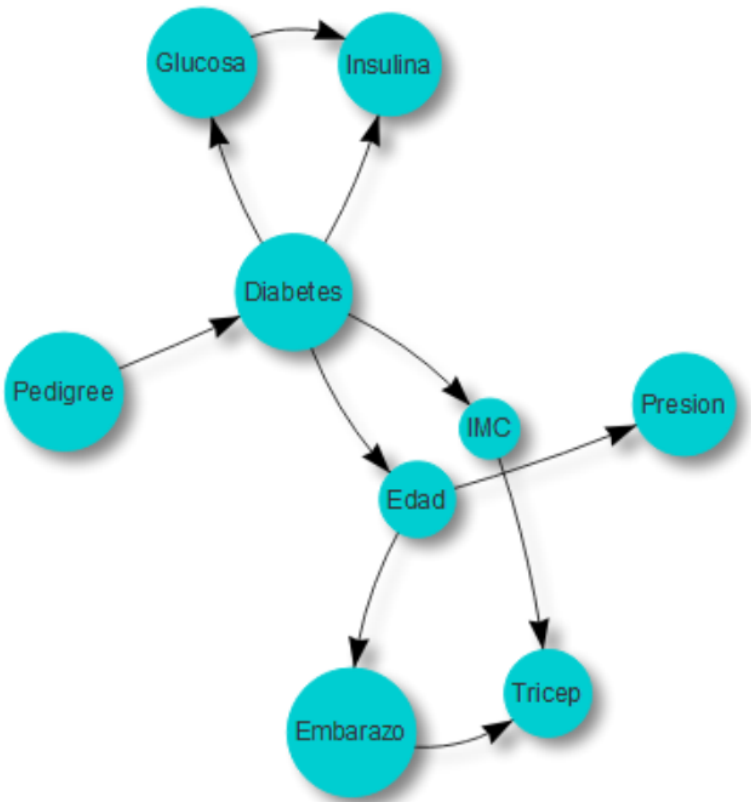


Figura 24: Grafo con algoritmo E-M sin restricciones.

En la figura 24 se aprecia que del nodo Diabetes salen 4 arcos a distintas variables, cuando se esperaría que la dirección fuera la contraria, pues las variables a las que dirige arcos deberían ser causas de diabetes y no en el sentido contrario.

Otra observación importante, es que la variable Diabetes es padre de Edad y ninguna variable debería entrar al nodo edad, pues ninguna la determina, pero la edad si podría determinar a otras variables.

Realizando un proceso de remuestreo tipo bootstrap con 200 muestras, se puede observar los siguientes resultados para la fuerza del arco y la fuerza de la dirección

Desde	**Hacía**	**arco**	**dirección**
Diabetes	Embarazo	0.275	0.5727273
Diabetes	Glucosa	**0.965**	0.4818653
Diabetes	Presión	0.220	0.6590909
Diabetes	Tricep	0.235	0.4893617
Diabetes	Insulina	0.485	0.5792079
Diabetes	IMC	**0.900**	0.4361111
Diabetes	Pedigree	**0.715**	0.6290323
Diabetes	Edad	0.265	0.5094340

Tabla 3: Fuerza del arco y dirección, m=200 sin restricciones

El último renglón sugiere que no debería considerarse el vertice $Diabetes - Edad$, pues la fuerza del arco es relativamente pequeña. Se puede interpretar que en las 200 muestras nuevas, solo en el 26 % de las redes aprendidas estuvo presente el arco de Diabetes a Edad o de Edad a Diabetes.

Se considerarán sólo los arcos cuya fuerza de arco sea mayor a 0.5. Así solo se considerará las aristas que unen Diabetes con Glucosa, Insulina, IMC y Pedigree.

De las observaciones anteriores se puede resumir que esperamos que ningún arco entre al nodo Edad y que las variables Glucosa, Insulina, IMC y Pedigree entren a Diabetes. Para esto se crea una lista negra (relaciones prohibidas) para las restricciones sobre edad y una lista blanca (relaciones obligadas) para asegurar los arcos que deben entrar al nodo Diabetes.

El primer modelo también permitió el acercamiento a expertos del Instituto Nacional de Geriatría los cuales sugirieron incluir los siguientes arcos en las modificaciones al modelo:

- IMC $\rightarrow$ Tricep
- Embarazo $\rightarrow$ Tricep

Estas dos últimas relaciones también se verifican en la simulación bootstrap, encontrando una fuerza del arco igual a 1. Es decir, en las 200 muestras aparecían las relaciones señaladas

y su fuerza de dirección era mayor de 0.8 para ambas. Por lo tanto, los datos confirman las sugerencias de los expertos y se agregan a la lista blanca, la cual posee los arcos que son obligatorios en la red.

Antes de crear otro modelo con las listas de relaciones prohibidas y obligadas es importante notar que las listas contienen relaciones que son congruentes con la figura 23, pues los arcos obligados coinciden con las variables que están altamente correlacionadas, aunque no hay una relación uno a uno entre las correlaciones y los modelos pues estos no se definen a partir de ellas.

Ahora, el grafo generado con el algoritmo E-M con 100 inicios aleatorios y las restricciones sobre los arcos se muestra en la figura 25.

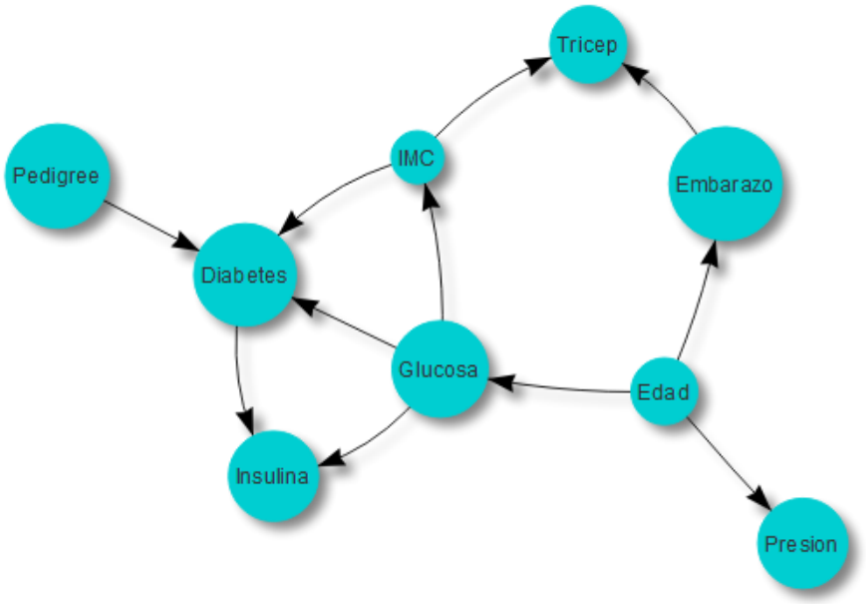


Figura 25: Grafo con algoritmo E-M y restricciones.

En este grafo, a diferencia del primer acercamiento, se incluye el conocimiento del experto sobre el tema y las relaciones causales esperadas en la realidad, gracias a las restricciones planteadas.

Similar al primer caso, se analizan las fuerzas de los arcos y direcciones. Se muestran en la tabla 4 y se puede apreciar que todas los arcos y direcciones aparecen en la mayoría de las 200 nuevas muestras en el algoritmo bootsrap.

Desde	**Hacía**	**arco**	**dirección**
Embarazo	Tricep	1.000	1.0000
Glucosa	Insulina	1.000	0.9725
Glucosa	Diabetes	1.000	1.0000
Insulina	IMC	0.875	0.9800
IMC	Tricep	1.000	1.0000
IMC	Diabetes	1.000	1.0000
Pedigree	Diabetes	1.000	1.0000
Edad	Embarazo	1.000	1.0000
Edad	Glucosa	0.815	1.0000
Edad	Presion	0.600	1.0000

Tabla 4: Fuerza del arco y dirección, $m = 200$ con restricciones

Se validó el grafo con los expertos, por lo que se continúa con el proceso de inferencia en el que se necesita el aprendizaje paramétrico de la estructura seleccionada. Las tablas de probabilidad condicional son:

Ped	0	1	2	3
	0.2508143	0.2508143	0.2622150	0.2361564

Tabla 5: $P(Pedigree)$

Edad	0	1
	0.781759	0.218241

Tabla 6: $P(Edad)$

Resaltar que hasta el momento únicamente se han utilizado los datos de entrenamiento. Estas tablas fueron creadas con el entrenamiento y muy probablemente cambiarían si el conjunto de entrenamiento cambiara.

Presion	0	1	2
Edad			
0	0.18541667	0.76250000	0.05208333
1	0.05970149	0.78358209	0.15671642

Tabla 7: $P(Presion|Edad)$

Embarazo	0	1	2
Edad			
0	0.3937500	0.3770833	0.2291667
1	0.1343284	0.1119403	0.7537313

Tabla 8: $P(Embarazo|Edad)$

Glucosa	0	1
Edad		
0	0.7750000	0.2250000
1	0.6044776	0.3955224

Tabla 9: $P(Glucosa|Edad)$

IMC	0	1	2
Glucosa			
0	0.18101545	0.24944812	0.56953642
1	0.04968944	0.18012422	0.77018634

Tabla 10: $P(IMC|Glucosa)$

	Tricep	0	1
IMC	**Embarazo**		
0	0	0.97222222	0.02777778
	1	0.87500000	0.12500000
	2	0.36363636	0.63636364
1	0	0.88636364	0.11363636
	1	0.79591837	0.20408163
	2	0.18367347	0.81632653
2	0	0.11811024	0.88188976
	1	0.26086957	0.73913043
	2	0.07857143	0.92142857

Tabla 11: $P(Tricep|IMC, Embarazo)$

6.3. Inferencia.

El paquete estadístico R tiene implementado el algoritmo de propagación en la paquetería *gRain*, misma que se utilizará en el caso de estudio para realizar inferencia de manera supervisada sobre la variable Diabetes.

	Insulina	0	1
Diabetes	**Glucosa**		
0	0	0.83661972	0.16338028
	1	0.16000000	0.84000000
1	0	0.67346939	0.32653061
	1	0.06306306	0.93693694

Tabla 12: $P(Insulina|Diabetes, Glucosa)$

		Diabetes	0	1
Pedigree	**IMC**	**Glucosa**		
0	0	0	1.00000000	0.00000000
		1	0.33333333	0.66666667
	1	0	0.93103448	0.06896552
		1	0.55555556	0.44444444
	2	0	0.78787879	0.21212121
		1	0.28571429	0.71428571
1	0	0	1.00000000	0.00000000
		1	1.00000000	0.00000000
	1	0	0.88461538	0.11538462
		1	0.55555556	0.44444444
	2	0	0.68852459	0.31147541
		1	0.30303030	0.69696970
2	0	0	1.00000000	0.00000000
		1	0.00000000	1.00000000
	1	0	0.83333333	0.16666667
		1	0.66666667	0.33333333
	2	0	0.73684211	0.26315789
		1	0.22580645	0.77419355
3	0	0	0.92307692	0.07692308
		1	0.50000000	0.50000000
	1	0	0.78571429	0.21428571
		1	0.50000000	0.50000000
	2	0	0.49090909	0.50909091
		1	0.17948718	0.82051282

Tabla 13: $P(Diabetes|Pedigree, IMC, Glucosa)$

En otras palabras, se tomará el valor observado de todas las variables diferentes a Diabetes para cada registro, estos representarán la evidencia y con base a dichos valores se obtendrá la probabilidad de que el paciente tenga Diabetes o no.

Este es un problema de clasificación o podría decirse que se utiliza la red bayesiana como un clasificador, la cual no es su función principal, pero es de gran apoyo para el problema a

tratar.

Una de las formas más comunes de cuantificar el desempeño de un clasificador es a través de la *precisión*, la cual mide la proporción de registros o pacientes que fueron clasificados de manera *correcta*. En general, es correcto si los pacientes con diabetes fueron clasificados con diabetes (valor 1 en la variable Diabetes) y los que no tienen diabetes se le asigno 0 en la variable Diabetes.

Para calcular el valor sugerido por la red y el método de inferencia, tenemos que calcular para cada paciente las probabilidades $P(Diabetes = si|E = e)$ y $P(Diabetes = no|E = e)$ donde E es el conjunto de evidencia, es decir $\mathbf{X} \setminus Diabetes$ y e es el valor observado para el conjunto de evidencia de cada uno de los pacientes.

Calculando las probabilidades para los 614 pacientes del conjunto de entrenamiento se tiene:

Paciente	$P(D = no\|E = e)$	$P(D = si\|E = e)$
1	0.3030303	0.69696970
2	0.6885246	0.31147541
3	0.5036869	0.49631310
4	0.8846154	0.11538462
5	0.8613281	0.13867187
⋮	⋮	⋮
610	0.6472196	0.35278042
611	1.0000000	0.00000000
612	0.9049786	0.09502141
613	0.8613281	0.13867187
614	0.9310345	0.06896552

Tabla 14: Probabilidades para diabetes en el conjunto de entrenamiento.

En clasificación binaria suele asignarse como valor inferido al que tenga la mayor probabilidad, es decir, la clase que tenga una probabilidad mayor a 0.5. Aquí, 0.5 se le conoce como punto de corte, pues a partir de este punto se determina el valor de la variable

Si se aplica este criterio para determinar el valor de Diabetes, entonces, el paciente 1 tiene el valor *si* o 1 en la variable Diabetes, puesto que la probabilidad de tener diabetes condicionada a los valores de las demás variables está por arriba de 0.5, mientras que el paciente 614 *no* o 0.

Con 0.5 como el punto de corte la precisión es igual a:

$$Precision = \frac{477}{614} = 0.776873$$

Es decir, se asignó de manera correcta a 477 pacientes de 614 o equivalentemente, en el 77.68 % de los casos se asignó de manera correcta, lo cual pareciera indicar un buen desem-

peño del modelo.

El criterio del punto de corte igual a 0.5 se debe a que se espera que las observaciones que originalmente eran 1 (en este caso si) tengan probabilidades muy grandes y que las observaciones que originalmente eran 0 tengan probabilidades pequeñas.

En la figura 26 se muestran las densidades correspondientes a las probabilidades estimadas bajo el modelo para ambas categorías de la variable Diabetes (valores reales, no predichos bajo el modelo). La curva roja corresponde a valores originales igual 1 y la curva azul a valores originales iguales a 0. Se puede interpretar que los valores originales que indican la presencia de Diabetes son más probables que tengan su probabilidad $P(D = si|E = e)$ más concentrada al rededor de 0.8.

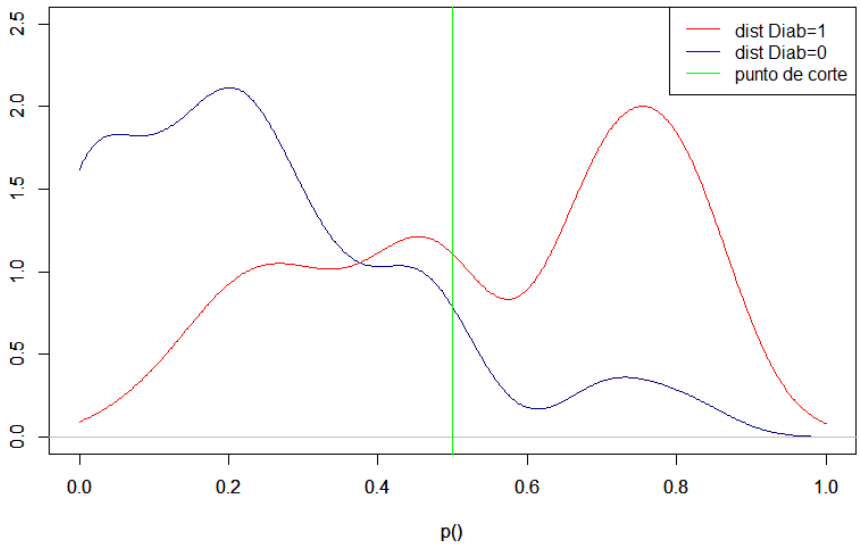


Figura 26: Densidades de los valores originales.

La línea verde indíca el punto de corte igual a 0.5 y toda la curva azul a la derecha del punto de corte esta mal clasificada, así como toda la curva roja a la izquierda de la recta verde.

Se observa que es mayor la cantidad de valores originales iguales a 1 los que están mal clasificados a comparación de los originales iguales a 0.

En los problemas de clasificación binaria puede importar más una categoría que otra, esto depende de que tan costoso sea dar una mala clasificación.

En concreto, para este caso, ¿es más costoso diagnosticar diabetes a un paciente que realmente no lo tiene? ó ¿es más costoso decirle a un paciente que no tiene diabetes cuando en realidad sí la padece?

De aquí en adelante se considerara de mayor interés la clase $D = 1$, pues el costo de una mala clasificación de un paciente con diabetes es mayor. Además de que aquellos clasificados por el modelo con diabetes podrían tener una segunda revisión o prueba más exhaustiva que de más certeza de padecer diabetes, con el objetivo de que aquellos que realmente no la tenían puedan ser descartados.

Gracias a que la categoría 1 es la de mayor interés, se puede desplazar el punto de corte, con el objetivo de reducir la clasificación incorrecta de esta clase. Ahora solo compararemos la probabilidad de que la variable tome el valor de la categoría de interés ($P(D = si|E = e)$), esto es la probabilidad de tener diabetes, y si la probabilidad estimada bajo el modelo es mayor a un nuevo punto de corte distinto a 0.5, de ser así se asigna al individuo a la categoría de *si* para Diabetes.

El punto de corte se desplazará a la izquierda, siendo su nuevo valor más pequeño que 0.5. Esto se puede interpretar también como la relevancia del evento y lo riesgoso que puede ser. Por ejemplo, no es lo mismo asignar 0.2 para el evento ganar cierta cantidad de dinero a comparación de la misma probabilidad a fallecer tras una operación. En el primer caso la probabilidad pareciera ser pequeña, mientras que en el segundo caso es considerablemente alto.

También, debido a la mayor importancia en una clase, surgen nuevas medidas de que tanto se equivoca el modelo en la clasificación de la categoría de mayor relevancia. La *sensibilidad* calcula la proporción de pacientes en la categoría de interés, tener diabetes, que fueron clasificados correctamente.

Por ejemplo, para el punto de corte 0.5 se tiene

$$sensibilidad = \frac{117}{209} = 0.5598086$$

Lo cual se interpreta que de 209 pacientes que originalmente tenían un valor de 1 para Diabetes, solo el 56 % de ellos se clasificó correctamente, lo cual corresponde a 117 pacientes.

A pesar de que la precisión parecía buena, la sensibilidad refleja que se estaba clasificando mal la categoría de interés en una proporción significativa.

La estrategia que se seguirá para modificar el punto de corte será tratar de mejorar la sensibilidad sin que la precisión disminuya más del 10 %. Es decir, tratar de identificar mejor los casos con diabetes y para conseguirlo, el punto de corte se desplazará hasta el máximo de los valores hasta antes del segundo punto de inflexión de la curva de densidad, que equivale al primer punto donde más observaciones se acumulan. Dicho de otra manera, se desplaza hasta el primer punto (de izquierda a derecha) donde deja de ser creciente la función de densidad de las observaciones de los pacientes con diabetes (curva roja).

Tomando esta estrategia, el punto de corte se modificará a 0.2681. La figura 27 muestra

la modificación al punto de corte

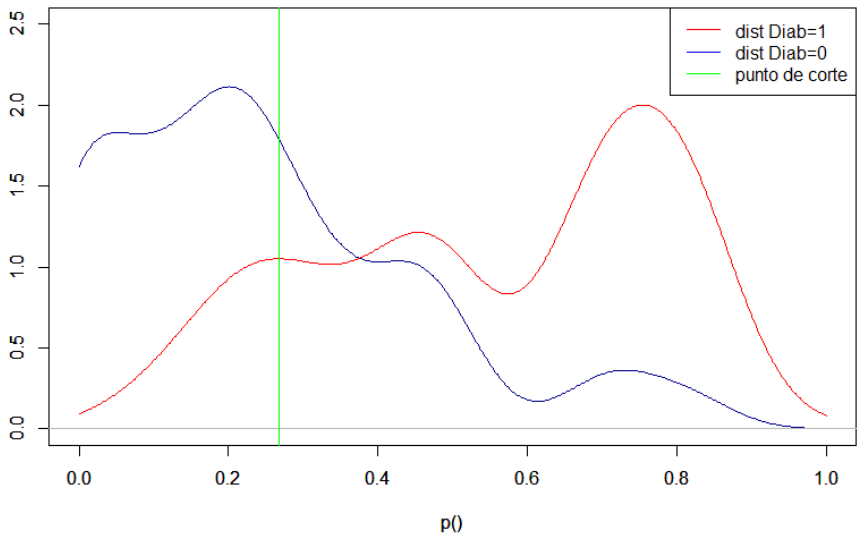


Figura 27: Modificación del punto de corte.

Con este nuevo punto de corte la precisión se modifica a **71.34 %** y la sensibilidad aumenta a **81.34 %**

Si se hubiera realizado el mismo proceso con la red sin restricciones (figura 24), se tendría una precisión del 73.62 % y una sensibilidad igual a 77.51 %. Por lo tanto, podemos decir que el modelo con restricciones obtenido (figura 25) es un mejor modelo, pues además que funciona mejor como clasificador, también modela el problema de forma comprensible desde el punto de vista de un experto, lo cual se traduce en una mayor interpretabilidad de los resultados.

El siguiente paso para el modelo, es verificar su desempeño en nuevas observaciones. Esto se realiza para verificar que se este modelando el problema con la naturaleza original de los datos y los resultados no sean buenos solo para el conjunto de datos de entrenamiento que seleccionamos.

La idea general es obtener las probabilidades $P(D = si|E = e)$ para los pacientes en el conjunto de prueba y con el punto de corte elegido por el conjunto de entrenamiento, determinar si estos pacientes fueron clasificados de manera correcta.

Se esperaría que el desempeño en el conjunto de prueba no difiera tanto del que presentó con el conjunto de entrenamiento.

La figura 28 muestra las densidades para el conjunto de prueba, así como el punto de corte elegido anteriormente, 0.2681.

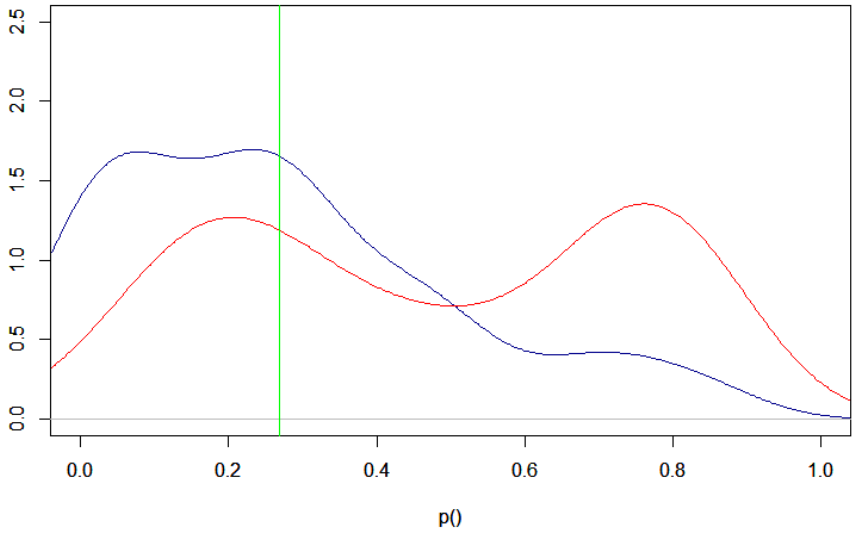


Figura 28: Densidades para el conjunto de prueba.

Para el conjunto de prueba la precisión es igual a **62.34 %** y la sensibilidad es igual a **64.41 %**.

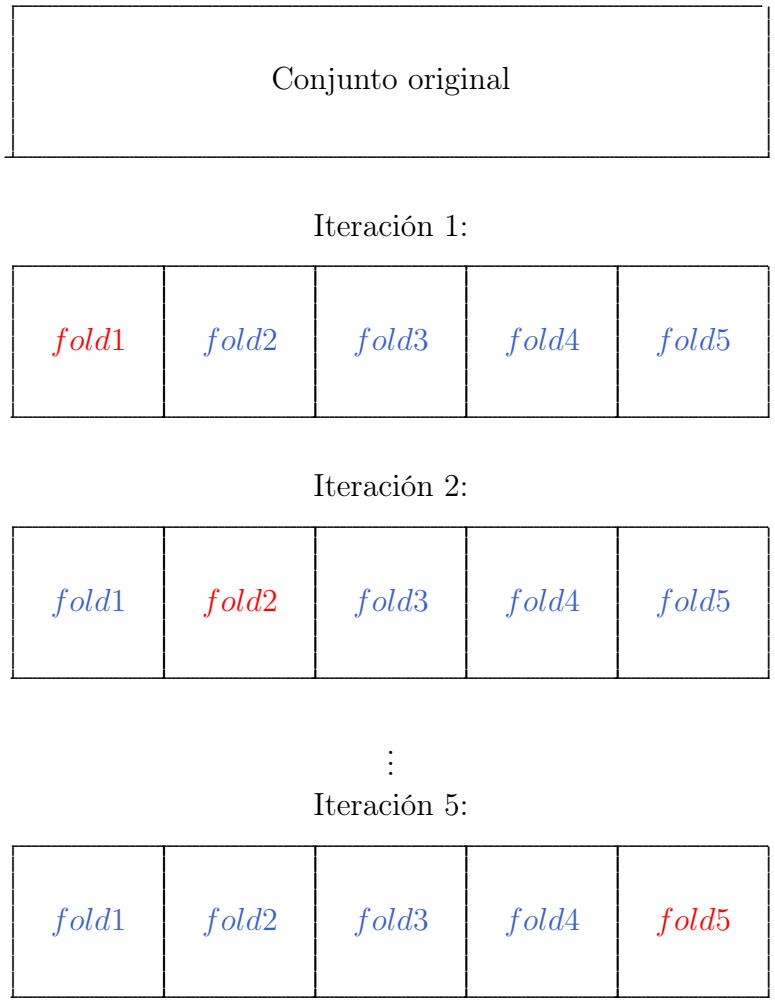
Se puede ver un descenso importante en las métricas y esto puede estar asociado directamente con el conjunto de entrenamiento y el conjunto de prueba seleccionados. Como se puede ver en la figura 28 se acumulan un número importante de observaciones que originalmente padecían diabetes antes del punto de corte, lo cual depende del conjunto de prueba pero también de las variables, pues si se pudiera explicar de mejor manera el fenómeno, la distribución de los pacientes con diabetes debería acumularse a la derecha del punto de corte. Además, para el proceso de aprendizaje se realizó un método de imputación, pero en el entrenamiento no se consideran las variables con valores faltantes.

Para tratar de solucionar esta problemática, se realizará validación cruzada. La validación cruzada consiste en dividir el conjunto original en k subconjuntos llamados *folds*. La técnica realiza k iteraciones y en cada una de ellas toma un *fold* diferente y se considera el conjunto de prueba, el resto de los *folds* se utilizan como conjunto de entrenamiento.

Con esta técnica se busca evaluar que el modelo no dependa de la selección del conjunto de entrenamiento y conjunto de prueba.

En cada iteración los resultados de precisión, sensibilidad y el punto de corte serán diferentes, por lo que el resultado final será el promedio del desempeño de cada uno de los *folds*.

A continuación se presenta una esquema de la implementación de la validación cruzada con 5 *folds*.



En cada iteración los *folds* azules sirven como conjunto de entrenamiento y se prueban sobre el único *fold* rojo.

Para el caso de estudio, se utilizara validación cruzada con 5 *folds*, eligiendo 5 para preservar la proporción 80-20 con la que se ha trabajado.

Los resultados de la validación cruzada se presentan en la tabla 15

En general, el promedio de los resultados se asemejan a los del modelo con restricciones para el primer conjunto de entrenamiento. Enfocandose en los resultados para nuevas observaciones, o conjunto de prueba, en el **77.5 %** de los casos logra clasificar de manera correcta a los pacientes que padecen diabetes, lo que parece representar un desempeño bueno para un modelo cuya función principal no es la de un clasificador.

Fold	Punto de corte	Presición entr	Sensib entr	Presición prueba	Sesns prueba
1	0.2504892	0.7100977	0.8130841	0.6558442	0.6296296
2	0.2681018	0.6894309	0.7892157	0.6993464	0.8593750
3	0.3033268	0.7084691	0.7440758	0.7012987	0.6842105
4	0.3091977	0.6959350	0.7304348	0.7320261	0.8684211
5	0.2739726	0.6758958	0.8403756	0.6428571	0.8363636
Promedio	**0.2810176**	**0.6959657**	**0.7834372**	**0.6862745**	**0.7756000**

Tabla 15: Resultados de la validación cruzada con 5 *folds*.

Como se explicó a lo largo de este trabajo, las redes bayesianas pueden realizar inferencia sobre cualquier conjunto de variables, por lo que podrían desarrollarse diversas consultas de cualquier fenómeno de interés sobre las variables consideradas por la red.

Por ejemplo, un médico podría estar interesado en que tan probable es que un paciente que tiene diabetes tenga la presión alta, motivado por el hecho de que al tener altos índices de glucosa se impide el flujo natural de la sangre por el sistema circulatorio, requiriendo, tal vez, mayor presión.

Para esta tarea, se utiliza la red de la figura 25 y las tablas de probabilidad condicionada mostradas en este capítulo. Propagando la evidencia se muestra que la siguiente tabla de probabilidades para la variable

Presion	0	1	2
Diabetes			
1	0.1530428	0.767929	0.07902818

Tabla 16: $P(Presion|D = 1)$

Contrarío a lo que se pensaba, la presión normal o categoría 1 es la más probable

Consultas como estas se pueden realizar sobre cualquier conjunto de variables. Por ejemplo, se podría esta interesado en calcular la función de probabilidad conjunta de diabetes e IMC cuando se sabe que el paciente ha tenido 2 embarazos y es mayor de 42 años. Propagando la evidencia se tiene el siguiente resultado

		Diabetes	0	1
Embarazo	**Edad**	**IMC**		
1	1	0	0.1163251	0.01274793
		1	0.1703141	0.05171486
		2	0.3098459	0.33905209

Tabla 17: $P(Diabetes, IMC|Em = 1, Ed = 1)$

La tabla anterior la podemos comparar a cuando no tenemos evidencia, dicha tabla se muestra a continuación:

Diabetes	0	1
IMC		
0	0.1370202	0.009559647
1	0.1851693	0.046101047
2	0.3358395	0.286310301

Tabla 18: $P(Diabetes, IMC)$

Se aprecia que conocer que el paciente ha tenido 2 embarazos y es mayor de 42 eleva la probabilidad de que padezca diabetes, pues los valores de la columna 1 son mayores en la tabla 17 comparandolos contra la tabla 18.

Lo anterior es de gran ayuda para interpretar que factores elevan las probabilidades de los eventos en cada uno de los pacientes y como se ha reiterado, esto se puede aplicar en cualquier conjunto de variables de la red y con cualquier conjunto de evidencia o variables ya observadas.

7. Conclusiones.

Las redes bayesianas son una herramienta útil cuando se debe especificar la distribución conjunta de una colección de variables aleatorias, ya que permite incorporar el conocimiento del tema a tratar y visualmente proporciona las relaciones de causalidad existentes.

Son de gran ayuda cuando con frecuencia se realizan consultas de estado de alguna o algunas variables, dado que se conoce el valor de otras. Esto gracias al proceso de inferencia que permite considerar a cualquier subconjunto de variables como variables respuesta, a diferencia de otros modelos en los que la respuesta es fija y cambiarla implicaría entrenar un nuevo modelo.

En general, las ventajas de optar por una red bayesiana son:

- La incorporación de la naturaleza del fenómeno a través de las independencias.
- La reducción en la especificación de parámetros.
- La obtención de manera eficaz y sencilla de probabilidades condicionales.
- Se puede realizar inferencia sobre cualquier subconjunto de variables.
- El modelo permite tener valores no observados en cualquier variable, se puede tomar como que no existe evidencia para esa variable o propagar y asignar un valor. Dicho de otra manera, posee un manejo adecuado de los valores faltantes.
- Si se requiere agregar alguna variable al modelo, no necesariamente se tiene que realizar todo el proceso para la construcción de la red de nuevo, se puede agregar el nodo si se conoce la dependencia con las otras variables y solo agregar o aprender los parámetros los cuales corresponden a las distribuciones del nodo condicionadas a los valores que toman sus padres.
- Visualmente se puede comprender con facilidad.

Estas ventajas cobran sentido a medida que el número de variables crece.

Algunas de las desventajas de este modelo son.

- Encontrar a algún experto que pueda transferir el conocimiento del tema.
- Costo computacional alto a medida que el número de variables crece.
- Al usarlo como clasificador puede no ser tan bueno como algoritmos más sofisticados.
- Dependen de la calidad de los datos, pues se construyen a partir de estos.

En el caso de estudio, se planteo una red bayesiana para ser utilizada como un clasificador, aunque la tarea principal de la red sea representar de manera compacta, visual y sencilla la distribución de las variables aleatorias consideradas.

El desempeño, evaluado a través de las medidas de clasificación correcta, es aceptable y, aunque existan modelos en la actualidad que puedan proporcionar una mejor clasificación, estos no representan las estructuras de dependencia que existen entre las variables y en casi todos los casos se limitan a hacer inferencia o clasificación sobre un conjunto de variables fijo, a diferencia de las redes bayesianas.

Uno de los modelos que poseen un buen desempeño al momento de clasificar y muy usados en los últimos años son las redes neuronales. Brownlee, J. (2019) desarrolla la misma tarea de clasificación para el mismo conjunto de datos, con una red neuronal con 3 capas ocultas, logrando una precisión del 77.73 %, resultado que no está tan alejado del que se presentó usando la red bayesiana.

A pesar de considerar bueno el desempeño, aún se puede perfeccionar más. El problema reside en las distribuciones de los pacientes que realmente tienen diabetes. La distribución de los pacientes que no tienen diabetes aparentemente no tiene problema, pues se acumula en una gran proporción a la izquierda del grafo (figura 25), dicho de otra manera, tienen probabilidades bajas de padecer diabetes.

Una posible solución a esta problemática es solicitar más datos a los pacientes que puedan ser relevantes para explicar el fenómeno. Si bien el IMC se calcula con base en el peso y la estatura, posiblemente sirva más para explicar el problema tener estas dos medidas por separado, pues incluso para algunos grupos de edad el IMC puede ser engañoso y sería mejor considerar el porcentaje de masa corporal, pues, por ejemplo, un deportista podría tener un alto IMC pero no así el porcentaje de grasa corporal.

Así como se puede descomponer la variable IMC, también se podría descomponer la variable Pedigree, considerando ahora los antecedentes familiares pero para cada una de las personas que se pueda relacionar en un nivel que se considere más prudente, por ejemplo, solo considerar a los padres y hermanos del paciente.

Otras variables que interesarían para poder explicar el fenómeno son los hábitos de alimentación y ejercicio de cada paciente.

Considerando estas variables, tal vez, podría acumularse la distribución de los que realmente tienen diabetes en las probabilidades altas o a la derecha de la gráfica y con esto lograr un mejor punto de corte que aumente las medidas de calidad de la clasificación.

La representación de la distribución de probabilidad conjunta se logró a través de 55 parámetros no redundantes, cuando haberlo hecho directamente de la distribución hubiese implicado asignar 3455 $((3*2*3*2*2*3*4*2*2)-1)$ parámetros, obtenidos al considerar los niveles de cada una de las 9 variables del modelo.

Un posible trabajo futuro será llevar esto al caso mexicano, obteniendo las variables que consideramos y otras que puedan ayudar a explicar el fenómeno de la diabetes. Para esto, se requerirá poder acceder a todas estas estadísticas en el sistema de salud mexicano. El

trabajo se haría pensando en el impacto que puede representar un mejor diagnóstico de la enfermedad que cuesta tantas vidas y recursos en la actualidad que atraviesa el país y lograr impactar tanto en el bienestar de las personas que padecen diabetes como en el sistema de salud.

A. Códigos.

Los códigos presentados en esta sección se encuentran en el siguiente repositorio: https://github.com/Ricardo27cruz27/Redes_Bayesianas

Código para la carga de datos, categorización de las variables y creación de los conjuntos de prueba y entrenamiento, así como la figura 23:

```
#CARGA DE LIBRERIAS
library(corrplot)
library(sqldf)
library(bnlearn)
library(visNetwork)

#Se carga librería para carga de datos desde github
library(RCurl)

#Carga de la base de datos
url<-"https://raw.githubusercontent.com/Ricardo27cruz27/Redes_
   Bayesianas/master/diabetes.csv"
url_csv<-getURL(url)
datos<-read.csv(text=url_csv,header = F)
#nombres de las variables
names(datos)<-c("Embarazo",
                "Glucosa",
                "Presion",
                "Tricep",
                "Insulina",
                "IMC",
                "Pedigree",
                "Edad",
                "Diabetes")
attach(datos)


#Análisis exploratorio
summary(datos[,1])
sort(unique(datos[,1]))
#hist(datos[,1])

summary(datos[,2])
sort(unique(datos[,2]))
#hist(datos[,2])

#NA'S
datos.limp<-datos
datos.limp[(which(Glucosa==0)),2]=NA
```

```
datos.limp[(which(Presion==0)),3]=NA
datos.limp[(which(Tricep==0)),4]=NA
datos.limp[(which(Insulina==0)),5]=NA
datos.limp[(which(IMC==0)),6]=NA


#Categorización.
#Embarazos
datos.limp[(which(datos.limp[,1]<=1)),1]=0
datos.limp[(which(datos.limp[,1]>1 & datos.limp[,1]<=4)),1]=1
datos.limp[(which(datos.limp[,1]>4)),1]=2

#Glucosa
datos.limp[(which(datos.limp[,2]<140)),2]=0
#datos.limp[(which(datos.limp[,2]>70 & datos.limp[,2]<140)),2]=1
datos.limp[(which(datos.limp[,2]>=140)),2]=1

#Presión
datos.limp[(which(datos.limp[,3]<=60)),3]=0
datos.limp[(which(datos.limp[,3]>60 & datos.limp[,3]<90)),3]=1
datos.limp[(which(datos.limp[,3]>=90)),3]=2

#Tricep
datos.limp[(which(datos.limp[,4]<=25)),4]=0
datos.limp[(which(datos.limp[,4]>25)),4]=1

#Insulina
datos.limp[(which(datos.limp[,5]<=120)),5]=0
datos.limp[(which(datos.limp[,5]>120)),5]=1

#IMC
datos.limp[(which(datos.limp[,6]<=25)),6]=0
datos.limp[(which(datos.limp[,6]>25 & datos.limp[,6]<=30)),6]=1
datos.limp[(which(datos.limp[,6]>30)),6]=2

#Pedigree
datos.limp[(which(datos.limp[,7]>quantile(Pedigree,.75))),7]=3
datos.limp[(which(datos.limp[,7]>quantile(Pedigree,.5) &
                    datos.limp[,7]<=quantile(Pedigree,.75))),7]=2
datos.limp[(which(datos.limp[,7]>quantile(Pedigree,.25) &
                    datos.limp[,7]<=quantile(Pedigree,.5))),7]=1
datos.limp[(which(datos.limp[,7]<=quantile(Pedigree,.25))),7]=0

#Edad
datos.limp[(which(datos.limp[,8]<=41)),8]=0
#datos.limp[(which(datos.limp[,8]>41 & datos.limp[,8]<=61)),8]=1
datos.limp[(which(datos.limp[,8]>41)),8]=1

```

```
#Correlación
correlacion<-cor(as.matrix(datos.limp),use = "pairwise.complete.obs")
corrplot.mixed(correlacion, upper="square",lower="number")
#corrplot(correlacion)

#D como el conjunto de datos
D<-datos.limp
#como factores
D[names(datos)]<-lapply(D[names(datos)],as.factor)
sapply(D, class)


#Conjunto de entrenamiento y prueba
set.seed(27)
muestra<-sample(1:nrow(D),size = 614)
D.train<-D[muestra,]
D.test<-D[-muestra,]

nrow(D.train); nrow(D.test)
```

Listing 1: Carga de datos

Creación de la red sin restricciones, incluyendo la figura 24 y la tabla 3:

```
#Aprendizaje estructural
#Generacion de un modelo a través de la
#muestra de entrenamiento

#Función para plotear redes:
plot.network <- function(structure, ht = "400px",title,subtitle){
  nodes.uniq <- unique(c(structure$arcs[,1], structure$arcs[,2]))
  nodes <- data.frame(id = nodes.uniq,
                      label = nodes.uniq,
                      color = "darkturquoise",
                      shadow = TRUE,
                      shape="circle")

  edges <- data.frame(from = structure$arcs[,1],
                      to = structure$arcs[,2],
                      arrows = "to",
                      smooth = TRUE,
                      shadow = TRUE,
                      color = "black")

  return(visNetwork(nodes, edges, height = ht, width = "100%",main=
     title,submain=subtitle
  ) %>% visLayout(randomSeed = 27)
  )
```

```
}

#RB sin ninguna restricción
RB.sinrest<-structural.em(D.train,return.all = TRUE,maximize.args =
   list(restart=100))
plot.network(RB.sinrest$dag,title = "",subtitle = "")

#función para fuerza de los arcos
fuerza.bootsrap<-function(RB,args=list()){
  arcs = boot.strength(RB$imputed, algorithm = "hc",algorithm.args =
     args)
  print(arcs[(arcs$strength > 0.5) & (arcs$direction >= 0.5), ])
  #prueba<-(averaged.network(arcs,threshold = .5))
  return(arcs)
}
boot.RB.sinrestricciones<-fuerza.bootsrap(RB.sinrest)
boot.RB.sinrestricciones[which(boot.RB.sinrestricciones$from=="
   Diabetes"),
                         ]
```

Listing 2: Red sin restricciones

Creación de la red con restricciones, incluyendo la figura 25 y las tablas de la 4 a la 13

```
#RED CON RESTRICCIONES
## ARCOS PROHIBIDOS:
blacklist<-data.frame(from=names(D)[-8],to=rep("Edad",8))
whitelist4=data.frame(from=c("IMC","Glucosa","Pedigree","IMC","
   Embarazo"),
                      to=c("Diabetes","Diabetes","Diabetes","Tricep",
                         "Tricep"))
RB.b1w4<-structural.em(D.train,
                       maximize.args = list(blacklist=blacklist,
                          restart=100,whitelist=whitelist4),
                       return.all = TRUE)
plot.network(RB.b1w4$dag,title = "",subtitle = "")
#bootsrap:
boot.RB.b1w4<-fuerza.bootsrap(RB.b1w4,list(whitelist=whitelist4,
   blacklist=blacklist))


#Tablas de probabilidad condicionada:
RB.b1w4$fitted$Pedigree
RB.b1w4$fitted$Edad
RB.b1w4$fitted$Presion
RB.b1w4$fitted$Embarazo
RB.b1w4$fitted$Glucosa
RB.b1w4$fitted$IMC
```

```
RB.b1w4$fitted$Tricep
RB.b1w4$fitted$Insulina
RB.b1w4$fitted$Diabetes
```

Listing 3: Red con restricciones

Creación de la tabla 14:

```
#INFERENCIA

#carga de librerías
library(BiocManager)
library(caret)
BiocManager::install("RBGL")
library(gRbase)
library(gRain)

#Redes como grain
gr.RB1<-as.grain(RB.sinrest$fitted)
gr.RB10<-as.grain(RB.b1w4$fitted)

#redes a probar
redes<-list(gr.RB1,gr.RB10)
#Funcion de propagacion
prop<-function(red){
  propagacion=red
  prediccion.train<-predict(propagacion,response = "Diabetes",newdata
     = D.train,predictors = names(D)[-9],type="distribution")
  pred.train<-as.integer(prediccion.train$pred$Diabetes)
  ac.train<-sum(Diab.train==pred.train)/nrow(D.train)

  prediccion.test<-predict(propagacion,response = "Diabetes",newdata =
      D.test,predictors = names(D)[-9],type="distribution")
  pred.test<-as.integer(prediccion.test$pred$Diabetes)
  ac.test<-sum(Diab.test==pred.test)/nrow(D.test)
  return(list(prediccion.train$pred$Diabetes,prediccion.test$pred$
     Diabetes))
}

ACC.PRU<-lapply(redes, prop)

#Probabilidades
ACC.PRU[[2]][1]
```

Listing 4: Probabilidades

Creación de la figura 26 y cálculo de la precisión y sensibilidad del modelo sin restricciones:

```
#red
rn<-2
#punto de corte
co<-.5
r.train<-as.integer(ACC.PRU[[rn]][[1]][,2]>co)
#matriz de confusion
cm.train<-table(r.train,D.train$Diabetes)
cm.train<-confusionMatrix(cm.train,positive = "1")
cm.train$overall[1]
cm.train$byClass[1]

#gráfico
his.train<-ACC.PRU[[2]][[1]][,2]
ceros.train<-his.train[which(Diab.train==0)]
unos.train<-his.train[which(Diab.train==1)]
dc.train<-density(ceros.train,from = 0,to = 1)
du.train<-density(unos.train,from = 0,to = 1)
x11()
plot(dc.train,xlim=c(0,1),ylim=c(0,2.5),main="",xlab="p()",ylab="",
   col="darkblue")
par(new=T)
plot(du.train,xlim=c(0,1),ylim=c(0,2.5),main="",xlab="p()",ylab="",
   col="red")
abline(v=.5,col="green")
legend(x="topright",legend=c("dist Diab=1","dist Diab=0","punto de
   corte"),
       lty=1,col=c("red","blue","green"))
```

Listing 5: modelo sin restricciones

Creación de la figura 27, figura 28 y cálculo de la precisión y sensibilidad del modelo con restricciones:

```
#Cambio del punto de corte
his.train<-ACC.PRU[[2]][[1]][,2]
ceros.train<-his.train[which(Diab.train==0)]
unos.train<-his.train[which(Diab.train==1)]
dc.train<-density(ceros.train,from = 0,to = 1)
du.train<-density(unos.train,from = 0,to = 1)
m<-max(du.train$y[which(du.train$x<.35)])
cutoff<-du.train$x[which(du.train$y==m)]
x11()
plot(dc.train,xlim=c(0,1),ylim=c(0,2.5),main="",xlab="p()",ylab="",
   col="darkblue")
par(new=T)
plot(du.train,xlim=c(0,1),ylim=c(0,2.5),main="",xlab="p()",ylab="",
   col="red")
```

```
abline(v=cutoff,col="green")

his.test<-ACC.PRU[[2]][[2]][,2]
ceros.test<-his.test[which(Diab.test==0)]
unos.test<-his.test[which(Diab.test==1)]
dc.test<-density(ceros.test)
du.test<-density(unos.test)
x11()
plot(dc.test,xlim=c(0,1),ylim=c(0,2.5),main="",xlab="p()",ylab="",col
   ="darkblue")
par(new=T)
plot(du.test,xlim=c(0,1),ylim=c(0,2.5),main="",xlab="p()",ylab="",col
   ="red")
abline(v=cutoff,col="green")
legend(x="topright",legend=c("dist Diab=1","dist Diab=0","punto de
   corte"),
       lty=1,col=c("red","blue","green"))

#precisión y sensibilidad con diferente punto de corte
rn<-2
co<-cutoff
r.train<-as.integer(ACC.PRU[[rn]][[1]][,2]>co)
r.test<-as.integer(ACC.PRU[[rn]][[2]][,2]>co)

cm.train<-table(r.train,D.train$Diabetes)
cm.train<-confusionMatrix(cm.train,positive = "1")

cm.test<-table(r.test,D.test$Diabetes)
cm.test<-confusionMatrix(cm.test,positive = "1")

cm.train
cm.test
```

Listing 6: modelo con restricciones

Cross validation y creación de la tabla 15:

```
#validación cruzada
estructura<-RB.b1w4$dag
#tamaños de los folds
tamaños<-c(154,153,154,153,154)
#entrenamiento de cada iteración
sets.test<-list(B1=D[0:cumsum(tamaños)[1],],
                B2=D[(cumsum(tamaños)[1]+1):cumsum(tamaños)[2],],
                B3=D[(cumsum(tamaños)[2]+1):cumsum(tamaños)[3],],
                B4=D[(cumsum(tamaños)[3]+1):cumsum(tamaños)[4],],
                B5=D[(cumsum(tamaños)[4]+1):cumsum(tamaños)[5],])
#prueba de cada iteración
sets.train<-list(B1=D[-(0:cumsum(tamaños)[1]),],
```

```
                  B2=D[-((cumsum(tamaños)[1]+1):cumsum(tamaños)[2]),],
                  B3=D[-((cumsum(tamaños)[2]+1):cumsum(tamaños)[3]),],
                  B4=D[-((cumsum(tamaños)[3]+1):cumsum(tamaños)[4]),],
                  B5=D[-((cumsum(tamaños)[4]+1):cumsum(tamaños)[5]),])
#Función de cada iteración en el cv
prueba.cv<-function(red,entrenamiento,pruebas){
  propagacion=red
  prediccion.train<-predict(propagacion,response = "Diabetes",newdata
     = entrenamiento,predictors = names(D)[-9],type="distribution")
  #pred.train<-as.integer(prediccion.train$pred$Diabetes)
  #ac.train<-sum(Diab.train==pred.train)/nrow(entrenamiento)

  prediccion.test<-predict(propagacion,response = "Diabetes",newdata =
      pruebas,predictors = names(D)[-9],type="distribution")
  #pred.test<-as.integer(prediccion.test$pred$Diabetes)
  #ac.test<-sum(Diab.test==pred.test)/nrow(pruebas)
  return(list(prediccion.train$pred$Diabetes,prediccion.test$pred$
     Diabetes))
}
#matriz de resumen del cv
Resumen<-matrix(0,5,5)
colnames(Resumen)<-c("Punto de corte",
                     "Acc train",
                     "Sensit train",
                     "Acc test",
                     "Sensit test")
#CV
for(i in 1:5){
  red.ajustada<-bn.fit(x = estructura,
                       data =sets.train[[i]])
  grain.ajustada<-as.grain(red.ajustada)
  probability<-prueba.cv(grain.ajustada,
                         sets.train[[i]],
                         sets.test[[i]])
  his.train<-probability[[1]][,2]
  ceros.train<-his.train[which(sets.train[[i]][,9]==0)]
  unos.train<-his.train[which(sets.train[[i]][,9]==1)]
  dc.train<-density(ceros.train,from = 0,to=1)
  du.train<-density(unos.train,from = 0,to=1)
  m<-max(du.train$y[which(du.train$x<.35)])
  cutoff<-du.train$x[which(du.train$y==m)]
  #m<-which.min(abs(du.train$y-dc.train$y))
  #cutoff<-du.train$x[m[1]]
  x11()
  plot(dc.train,xlim=c(0,1),ylim=c(0,2.5),main=paste("train",i),xlab=
     "p()",ylab="",col="darkblue")
  par(new=T)
  plot(du.train,xlim=c(0,1),ylim=c(0,2.5),main=paste("train",i),xlab=
```

```
      "p()",ylab="",col="red")
  abline(v=cutoff,col="green")

  his.test<-probability[[2]][,2]
  ceros.test<-his.test[which(sets.test[[i]][,9]==0)]
  unos.test<-his.test[which(sets.test[[i]][,9]==1)]
  dc.test<-density(ceros.test)
  du.test<-density(unos.test)
  x11()
  plot(dc.test,xlim=c(0,1),ylim=c(0,2.5),main=paste("test",i),xlab="p
      ()",ylab="",col="darkblue")
  par(new=T)
  plot(du.test,xlim=c(0,1),ylim=c(0,2.5),main=paste("test",i),xlab="p
      ()",ylab="",col="red")
  abline(v=cutoff,col="green")
  #cortes<-c(cortes,cutoff)

  r.train<-as.integer(probability[[1]][,2]>cutoff)
  r.test<-as.integer(probability[[2]][,2]>cutoff)

  cm.train<-table(r.train,sets.train[[i]][,9])
  cm.train<-confusionMatrix(cm.train,positive = "1")

  cm.test<-table(r.test,sets.test[[i]][,9])
  cm.test<-confusionMatrix(cm.test,positive = "1")

  Resumen[i,1]<-cutoff
  Resumen[i,2]<-cm.train$overall[1]
  Resumen[i,3]<-cm.train$byClass[1]
  Resumen[i,4]<-cm.test$overall[1]
  Resumen[i,5]<-cm.test$byClass[1]
}

#Resumen
row.names(Resumen)<-c("fold1","fold2","fold3","fold4","fold5")
Resumen
colMeans(Resumen)
```

Listing 7: Cross validation

Creación de las tablas 16, 17 y 18:

```

#consultas:
diab1<-D[which(D$Diabetes==1),c("Diabetes","Presion")]
pred<-predict(gr.RB10,
              response = "Presion",
              newdata = diab1,
              predictors = "Diabetes",
```

```
                type="distribution")
pred$pred


querygrain(gr.RB10,
           nodes = c("Diabetes", "IMC"),
           type = "joint")
gr.RB10.evi<-setFinding(gr.RB10,
                        nodes=c("Embarazo","Edad"),
                        states=c("1","1"))
querygrain(gr.RB10.evi,
           nodes = c("Diabetes", "IMC"),
           type = "joint")
```

Listing 8: Consultas

Referencias

Borgelt, C., Steinbrecher, M., y Kruse, R. R. (2009). *Graphical models: representations for learning, reasoning and data mining*. John Wiley & Sons.

Brownlee, J. (2019). *How to save and load your keras deep learning model*. Descargado 2019-05-13, de http://machinelearningmastery.com/save-load-keras-deep-learning-models/

Castillo, E., Gutiérrez, J. M., y Hadi, A. S. (1997). Sistemas expertos y modelos de redes probabilısticas. *Academia de Ingenierıa*.

Darwiche, A. (2009). *Modeling and reasoning with bayesian networks*. Cambridge university press.

Frey, B. J., Brendan, J. F., y Frey, B. J. (1998). *Graphical models for machine learning and digital communication*. MIT press.

Gámez, J. A., Mateo, J. L., y Puerta, J. M. (2011). Learning bayesian networks by hill climbing: efficient methods based on progressive restriction of the neighborhood. *Data Mining and Knowledge Discovery*, *22*(1-2), 106–148.

Højsgaard, S., Edwards, D., y Lauritzen, S. (2012). *Graphical models with r*. Springer Science & Business Media.

Jordan, M., Lauritzen, S., Lawless, J., y Nair, V. (s.f.). Statistics for engineering and information science.

Koller, D., y Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Koski, T., y Noble, J. (2011). *Bayesian networks: an introduction* (Vol. 924). John Wiley & Sons.

Luger, G. F., y Stubblefield, W. A. (1990). *Artificial intelligence and the design of expert systems*. Benjamin-Cummings Publishing Co., Inc.

Moral, S. (2014). Una introducción a las redes bayesianas.

Neapolitan, R. E., y cols. (2004). *Learning bayesian networks* (Vol. 38). Pearson Prentice Hall Upper Saddle River, NJ.

Nielsen, T. D., y Jensen, F. V. (2009). *Bayesian networks and decision graphs*. Springer Science & Business Media.

Scutari, M., y Denis, J.-B. (2014). *Bayesian networks: with examples in r*. Chapman and Hall/CRC.

Sucar, L. E. (2015). Probabilistic graphical models. *Advances in Computer Vision and Pattern Recognition. London: Springer London. doi*, *10*, 978–1.