Chapter 1

Redes Bayesianas

Luis Enrique Sucar
INAOE
Sta. María Tonantzintla, Puebla, 72840, México
Correo electrnico: esucar@inaoep.mx

1.1 Introducción

Las redes bayesianas modelan un fenómeno mediante un conjunto de variables y las relaciones de dependencia entre ellas. Dado este modelo, se puede hacer inferencia bayesiana; es decir, estimar la probabilidad posterior de las variables no conocidas, en base a las variables conocidas. Estos modelos pueden tener diversas aplicaciones, para clasificación, predicción, diagnóstico, etc. Además, pueden dar información interesante en cuanto a cmo se relacionan las variables del dominio, las cuales pueden ser interpretadas en ocasiones como relaciones de causa–efecto.

Incialmente, estos modelos eran construidos 'a mano' basados en un conocimiento experto, pero en los últimos años se han desarrollado diversas técnicas para aprender a partir de datos, tanto la estructura como los parámetros asociados al modelo. También es posible el combinar conocimiento experto con los datos para aprender el modelo.

A continuación, veremos una introducción general a redes bayesianas y los principales métodos de inferencia. Después, introducimos una estructura particular, los clasificadores bayesianos, y veremos cmo aprenderlos de datos. Porteriormente tratamos el tema de aprendizaje en general de redes bayesianas, tanto paramétrico como estructural. Concluimos hablando de la redes bayesianas dinámicas y cmo se pueden aprender estas estructuras. Al final se dan referencias y lecturas adicionales para cada tema.

1.2 Redes bayesianas

Las redes bayesianas son una representación gráfica de dependencias para razonamiento probabilstico, en la cual los nodos representan variables aleatorias y los arcos representan relaciones de dependencia directa entre las variables. La Figura 1.1 muestra un ejemplo hipotético de una red bayesiana (RB) que representa cierto conocimiento sobre medicina. En este caso, los nodos representan enfermedades, síntomas y factores que causan algunas enfermedades. La variable a la que apunta un arco es dependiente de la que está en el origen de éste, por ejemplo $fiebre$ depende de $tifoidea$ y $gripe$ en la red de la Figura 1.1. La topología o estructura de la red nos da información sobre las dependencias probabilísticas entre las variables. La red también representa las independencias condicionales de una variable (o conjunto de variables) dada(s) otra(s) variable(s). Por ejemplo, en la red de la Figura 1.1, *reacciones* es cond. indep. de C, G, F, D dado $tifoidea$. (Donde: C es comida, T es tifoidea, G es gripe, R es reacciones, F es fiebre y D es Dolor). Esto es:

$$P(R|C,T,G,F,D) = P(R|T) \tag{1.1}$$

Esto se representa gráficamente por el nodo T *separando* al nodo R del resto de las variables.

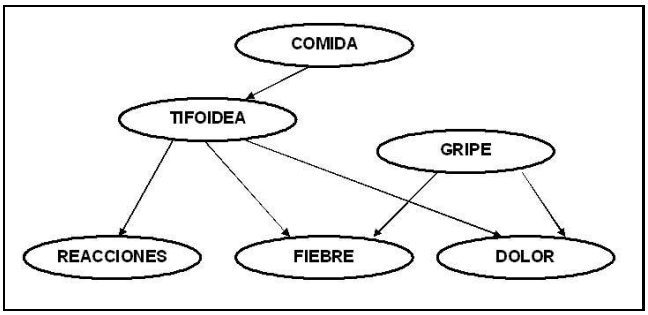

Figure 1.1: Ejemplo de una red bayesiana. Los nodos representan variables aleatorias y los arcos relaciones de dependencia.

En una RB todas la relaciones de independencia condicional representadas en el grafo corresponden a relaciones de independencia en la distribución de probabilidad. Dichas independencias simplifican la representación del conocimiento (menos parámetros) y el razonamiento (propagación de las probabilidades). Una red bayesiana representa en forma gráfica las dependencias e independencias entre variables aleatorias, en particular las independencias condicionales. Lo anterior se representa con la siguiente notación, para el caso de X independiente de Y dado Z:

- Independencia en la distribución: $P(X|Y,Z) = P(X|Z)$.
- Independencia en el grafo: $I < X \mid Z \mid Y >$.

La independencia condicional se verifica mediante el criterio de separación-D. Antes de definir formalmente la separación-D, es necesario distinguir tres tipos de nodos de acuerdo a las direcciones de los arcos que inciden en el nodo:

- Nodos en secuencia: $X \rightarrow Y \rightarrow Z$.
- Nodos divergentes: $X \leftarrow Y \rightarrow Z$.
- Nodos convergentes: $X \rightarrow Y \leftarrow Z$.

Separación D

El conjunto de variables A es independiente del conjunto B dado el conjunto C, si no existe trayectoria entre A y B en que:

1. Todos los nodos convergentes están o tienen descendientes en C.
2. Todos los demás nodos están fuera de C.

Por ejemplo, en la Figura 1.1, R es independiente de C dado T, pero T y G nos son independientes dado F.

Dada una distribución de probabilidad o modelo (M) y una representación gráfica de dependencias o grafo (G) debe existir una correspondencia entre las independencias representadas en ambos. En una RB, cualquier nodo X es independiente de todos los nodos que no son sus descendientes dados sus nodos padres, $Pa(X)$, denominado el contorno de X. La estructura de una RB se especifica indicando el contorno (padres) de cada variable. La estructura de la RB en la Figura 1.1 se especifica de la siguiente manera:

1. Pa(C) = $\emptyset$
2. Pa(T) = C
3. Pa(G) = $\emptyset$
4. Pa(R) = T
5. Pa(F) = T,G
6. Pa(D) = T,G

La cobija de Markov (manto de Markov, *Markov Blanquet*) de un nodo es el conjunto de nodos que lo hacen independiente del resto de la red. Para una RB, la cobija de Markov está formada por:

- Nodos padre.
- Nodos hijo.
- Otros padres de los hijos.

Complementa la definición de una red bayesiana las probabilidades condicionales de cada variable dados sus padres:

- Nodos raíz: vector de probabilidades marginales.
- Otros nodos: matriz de probabilidades condicionales dados sus padres.

La Figura 1.2 ilustra un ejemplo de algunas de las matrices de probabilidad asociadas al ejemplo de la Figura 1.1.

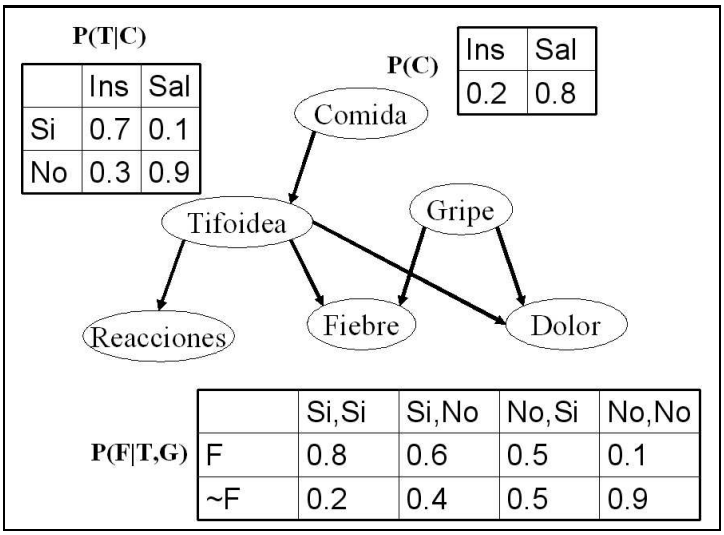

Figure 1.2: Parámetros asociados a una red bayesiana. Se muestran las tablas de probabilidad condicional de algunas de las variables de la red bayesiana de la Figura 1.1: probabilidad *a priori* de Comida, $P(C)$; probabilidad de Tifoidea dada Comida, $P(T \mid C)$; y probabilidad de Fiebre dada Tifoidea y Gripe, $P(F \mid T, G)$. En este ejemplo se asume que todas las variables son binarias.

Dado que los contornos (padres) de cada nodo especifican la estructura, mediante las probabilidades condicionales de dichos nodos podemos especificar también las probabilidades requeridas. Aplicando la regla de la cadena y las independencias condicionales, se puede verificar que con dichas probabilidades se puede calcular la probabilidad conjunta. En general, la probabilidad conjunta se especifica por el producto de las probabilidades de cada variable dados sus padres:

$$P(X1, X2, ..., Xn) = \prod_{i=1}^{n} P(Xi|Pa(Xi)) \tag{1.2}$$

El tamaño de la tabla de probabilidad condicional crece exponencialmente con el número de padres de un nodo, por lo que puede crecer demasiado. Una forma de reducir este problema es utilizando ciertos modelos para representar las tablas sin requerir especificar todas las probabilidades, utilizando lo que se conoce como *modelos canónicos*. Los principales tipos de modelos canónicos son:

- Modelo de interacción disyuntiva (Noisy OR).
- Modelo de interacción conjuntiva (Noisy AND).
- Compuerta Max (Noisy Max gate).
- Compuerta Min (Noisy Min gate).

El modelo canónico más común es el Noisy-OR, que se aplica cuando varias causas pueden ocasionar un efecto cada una por s sola, y la probabilidad del efecto no disminuye si se presentan varias causas. Por ejemplo, este modelo se puede aplicar cuando varias enfermedades pueden producir el mismo síntoma. En este caso sólo se especifica un parámetro por cada nodo padre, considerando variables binarios, en vez de 2^n, donde n es el número de padres.

Otras formas compactas de representar las tablas de probabilidad condicional son mediante árboles de decisión y redes neuronales.

1.2.1 Inferencia

El razonamiento probabilístico o propagación de probabilidades consiste en propagar los efectos de la evidencia a través de la red para conocer la probabilidad *a posteriori* de las variables. Es decir, se le dan valores a ciertas variables (evidencia), y se obtiene la probabilidad posterior de las demás variables dadas las variables conocidas (el conjunto de variables conocidas puede ser vacío, en este caso se obtienen las probabilidades *a priori*). Existen diferentes tipos de algoritmos para calcular las probabilidades posteriores, que dependen del tipo de grafo y de si obtienen la probabilidad de una variable a la vez o de todas. Los principales tipos de algoritmos de inferencia son:

1. Una variable, cualquier estructura: algoritmo de eliminación (*variable elimination*).
2. Cualquier variable, estructuras sencillamente conectadas: algoritmo de propagación de Pearl.
3. Cualquier variable, cualquier estructura: (i) agrupamiento (*junction tree*), (ii) simulación estocástica, y (iii) condicionamiento.

A continuación, veremos el algoritmo de propagación en árboles y poliárboles, que se ilustran en la Figura 1.3; y después el de agrupamiento o árbol de uniones.

Propagación en árboles

Este algoritmo se aplica a estructuras de tipo árbol, y se puede extender a poliárboles (grafos sencillamente conectados en que un nodo puede tener más de un padre).

Dada cierta evidencia E, representada por la instanciación de ciertas variables, la probabilidad posterior de cualquier variable B, por el teorema de Bayes (ver Figura 1.4):

$$P(Bi|E) = P(Bi)P(E|Bi)/P(E) \tag{1.3}$$

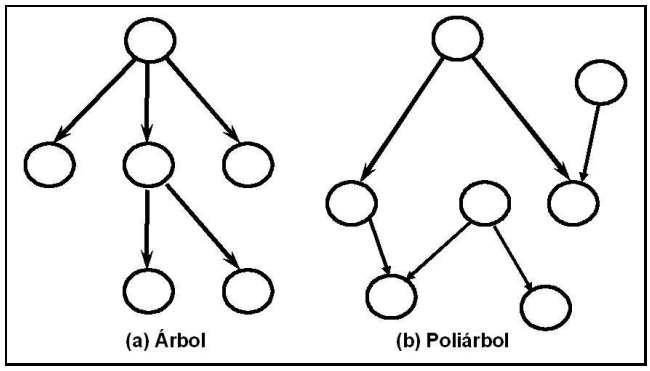

Figure 1.3: Estructuras sencillamente conectadas: (a) árbol, (b) poliárbol.

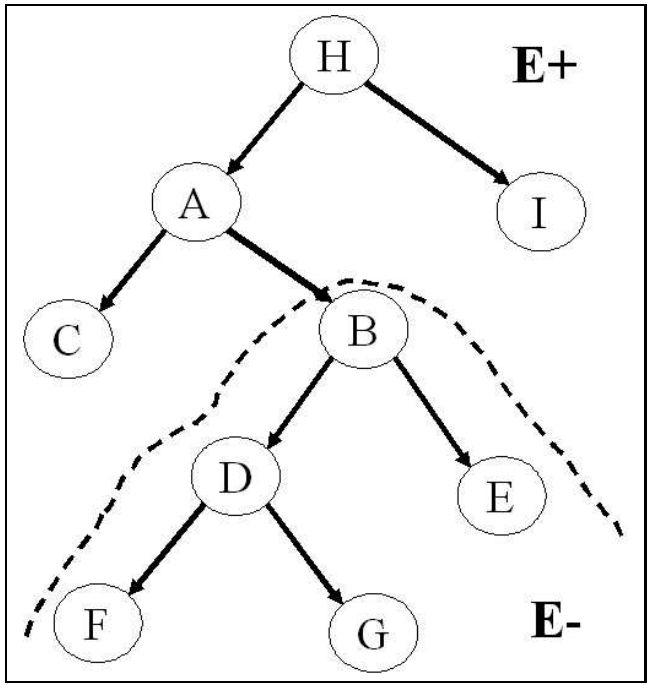

Figure 1.4: Propagación en árboles. En un árbol, cualquier nodo (B) divide la red en dos subgrafos condicionalmente independientes, $E+$ y $E-$.

Ya que la estructura de la red es un árbol, el Nodo B la separa en dos subárboles, por lo que podemos dividir la evidencia en dos grupos:

E-: Datos en el árbol que cuya raíz es B.

E+: Datos en el resto del árbol.

Entonces:

$$P(Bi|E) = P(Bi)P(E-, E+|Bi)/P(E) \tag{1.4}$$

Pero, dado que ambos son independientes y aplicando nuevamente a Bayes:

$$P(Bi|E) = \alpha P(Bi|E+)P(E-|Bi) \tag{1.5}$$

Donde α es una constante de normalización Si definimos los siguientes términos:

$$\lambda(Bi) = P(E-|Bi) \tag{1.6}$$

$$\pi(Bi) = P(Bi|E+) \tag{1.7}$$

Entonces:

$$P(Bi|E) = \alpha\pi(Bi)\lambda(Bi) \tag{1.8}$$

En base a la ecuación anterior, se puede integrar un algoritmo distribuido para obtener la probabilidad de un nodo dada cierta evidencia. Para ello, se descompone el cálculo en dos partes: (i) evidencia de los hijos (λ), y (ii) evidenica de los demás nodos (π). Cada nodo guarda los valores de los vectores π y λ, así como las matrices de probabilidad P. La propagación se hace por un mecanismo de paso de mensajes, en donde cada nodo envía los mensajes correspondientes a su padre e hijos. Mensaje al padre (hacia arriba), nodo B a su padre A:

$$\lambda_B(Ai) = \sum_j P(B_j \mid A_i)\lambda(B_j) \tag{1.9}$$

Mensaje a los hijos (hacia abajo), nodo B a su hijo S_k :

$$\pi_k(Bi) = \alpha\pi(B_j)\prod_{l\neq k}\lambda_l(B_j) \tag{1.10}$$

Al instanciarse ciertos nodos, éstos envían mensajes a sus padres e hijos, y se propagan hasta a llegar a la raíz u hojas, o hasta encontrar un nodo instanciado. Al final de la propagación, cada nodo tiene un vector π y un vector λ. Entonces se obtiene la probabilidad simplemente multiplicando ambos (término por término) de acuerdo a la ecuación 1.8. La propagación se realiza una sla vez en cada sentido (hacia la raíz y hacia las hojas), en un tiempo proporcional al diámetro (distancia de la raíz a la hoja más lejana) de la red.

Este algoritmo se puede extendar fcilmente para poliárboles, pero no se aplica en redes multiconectadas. En este caso hay varios algoritmos, en la siguiente sección analizaremos el de 'árbol de uniones'.

Propagación en redes multiconectadas

El algoritmo general más común en redes bayesianas es el de agrupamiento o 'árbol de uniones' (*junction tree*). El método de agrupamiento consiste en transformar la estructura de la red para obtener un árbol, mediante agrupación de nodos usando la teoría de grafos. Para ello, se hace una transformación de la red a un árbol de uniones (grupos de nodos) mediante el siguiente procedimiento:

1. Eliminar la direccionalidad de los arcos.
2. Ordenamiento de los nodos por máxima cardinalidad.
3. Moralizar el grafo (arco entre nodos con hijos comunes).
4. Triangular el grafo.
5. Obtener los cliques y ordenar.
6. Construir árbol de cliques.

Un *clique* es un subconjunto de nodos completamente conectados máximo, de forma que hay un arco entre cada par de nodos, y no existe un conjunto completamente conectado del que éste sea subconjunto. La Figura 1.5 ilustra esta transformación para una red sencilla. Para los detalles del algoritmo vanse las referencias al final del capítulo.

Un vez transformado el grafo, la propagación es mediante el envío de mensajes en el árbol de uniones o cliques (en forma similar a árboles). Inicialmente se calcula la probabilidad conjunta (potencial) de cada clique, y la condicional dado el padre. Dada cierta evidencia se recalculan las probabilidades de cada clique. La probabilidad individual de cada variable se obtiene de la del clique por marginalización.

En el peor caso, la propagación en redes bayesianas es un problema NP-duro. En la práctica, en muchas aplicaciones se tienen redes no muy densamente conectadas y la propagación es eficiente aún para redes muy grandes (función del clique mayor). Para redes muy complejas (muchas conexiones), la mejor alternativa son técnicas de simulación estocástica o técnicas aproximadas.

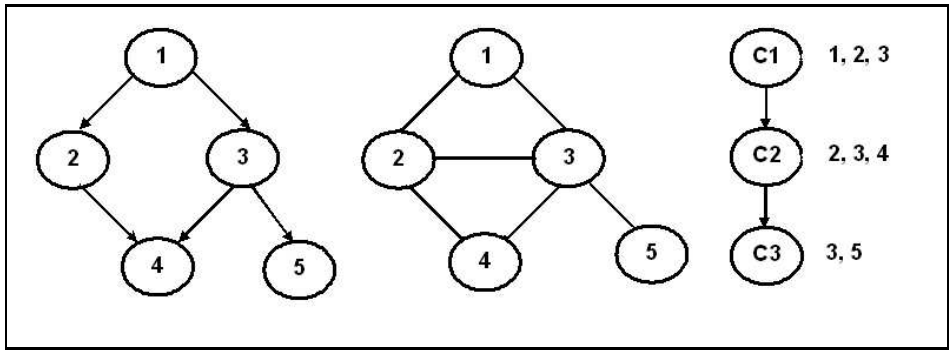

Figure 1.5: Transformación de una red a un árbol de uniones: (a) red original, (b) red moralizada y triangulada, (c) árbol de uniones.

1.3 Aprendizaje de clasificadores bayesianos

Un clasificador, en general, suministra una función que mapea (clasifica) un dato (instancia), especificado por una serie de características o atributos, en una o

diferentes clases predefinidas. Los clasificadores bayesianos son ampliamente utilizados debido a que presentan ciertas ventajas:

1. Generalmente, son fáciles de construir y de entender.
2. Las inducciones de estos clasificadores son extremadamente rápidas, requiriendo sólo un paso para hacerlo.
3. Es muy robusto considerando atributos irrelevantes.
4. Toma evidencia de muchos atributos para realizar la predicción final.

Un clasificador bayesiano se puede ver como un caso especial de una red bayesiana en la cual hay una variable especial que es la clase y las demás variables son los atributos. La estructura de esta red depende del tipo de clasificador, como veremos más adelante.

1.3.1 Clasificador bayesiano simple

Un clasificador bayesiano obtiene la probabilidad posterior de cada clase, C_i, usando la regla de Bayes, como el producto de la probabilidad *a priori* de la clase por la probabilidad condicional de los atributos (E) dada la clase, dividido por la probabilidad de los atributos:

$$P(C_i \mid E) = P(C_i)P(E \mid C_i)/P(E) \tag{1.11}$$

El clasificador bayesiano simple (*naive Bayes classifier*, NBC) asume que los atributos son independientes entre sí dada la clase, así que la probabilidad se puede obtener por el producto de las probabilidades condicionales individuales de cada atributo dado el nodo clase:

$$P(C_i \mid E) = P(C_i)P(E_1 \mid C_i)P(E_2 \mid C_i)...P(E_n \mid C_i) \mid C_i)/P(E) \tag{1.12}$$

Donde n es el número de atributos. Esto hace que el número de parámetros se incremente linealmente con el nmero de atributos, en vez de hacerlo en forma exponencial. Gráficamente, un NBC se puede representar como una red bayesiana en forma de estrella, con un nodo de la raíz, C, que corresponde a la variable de la clase, que está conectada con los atributos, $E_1, E_2, ..., E_n$. Los atributos son condicionalmente independientes dada la clase, de tal manera que no existen arcos entre ellos. Esta estructura se ilustra en el Figura 1.6.

Dado que la estructura de una clasificador bayesiano simple está predeterminada, sólo es necesario aprender los parámetros asociados, que son:

$P(C)$**:** vector de probabilidades *a priori* para cada clase.

$P(E_i \mid C)$**:** matriz de probabilidad condicional para cada atributo dada la clase.

Estos parámetros se pueden estimar fcilmente, a partir de los datos, en base a frecuencias. El denominador en la ecuación 1.12 no se requiere, ya que es una constante; es decir, no depende de la clase. Al final se pueden simplemente

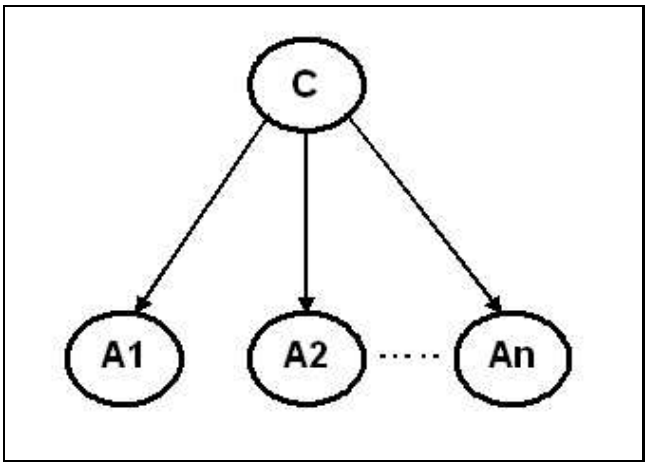

Figure 1.6: Clasificador bayesiano simple. Los atributos $A_1, A_2, ..., A_n$ son condicionalmente independientes dada la clase C.

normalizar las probabilidades posteriores de cada clase (haciendo que sumen uno).

Aunque el clasificador bayesiano simple funciona muy bien (tiene una alta precisión en clasificación) en muchos dominios, en ocaciones su rendimiento decrece debido a que los atributos no son condicionalmente independientes como se asume. En las secciones siguientes veremos dos enfoques para resolver esta limitación.

1.3.2 Extensiones al clasificador bayesiano

Cuando se tienen atributos dependientes, una forma de considerar estas dependencias es extendiendo la estructura básica de NBC agregando arcos entre dichos atributos. Existen dos alternativas básicas:

TAN: clasificador bayesiano simple aumentado con un árbol.

BAN: clasificador bayesiano simple aumentado con una red.

En ambos casos se extiende el NBC agregando una estructura de dependencias entre los atributos. En el TAN, se agrega una estructura de árbol entre los atributos, de forma que se tienen en principio 'pocas' conexiones y no aumenta demasiado la complejidad de la estructura. Para el BAN se agrega una estructura general de dependencia entre atributos, sin limitaciones. Dichas estructuras, tanto la de árbol como la general, se pueden aprender mediante los algoritmos de aprendizaje estructural que veremos más adelante. Una vez obtenida la estructura de dependencia entre atributos, se agregan arcos de la clase a cada uno de los atributos. La Figura 1.7 muestra un ejemplo de BAN y uno de TAN.

En algunos dominios, la precisión de la clasificación aumenta utilizando TAN o BAN, pero no hay uno claramente mejor al otro; y en ciertos casos el NBC

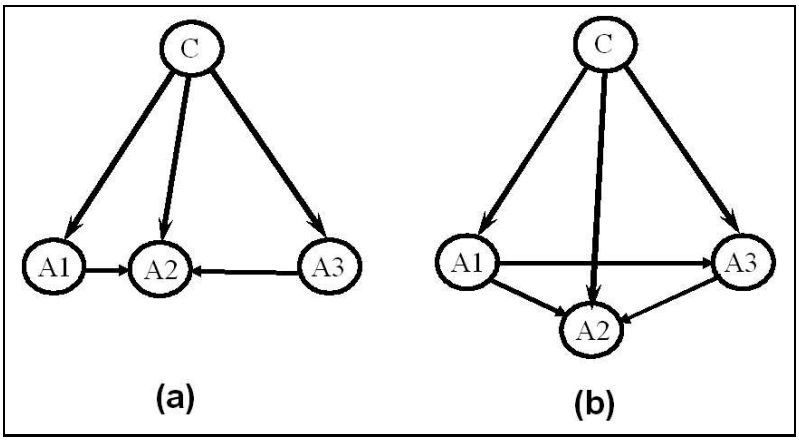

Figure 1.7: Extensiones al clasificador bayesiano simple: (a) TAN, (b) BAN.

da una mayor precisión. La desventaja de estas extensiones es que aumenta la complejidad (y el tiempo) tanto para aprender el modelo como para clasificación. Otra alternativa es tratar de mantener la misma estructura sencilla del NBC pero considerando las dependencias entre atributos, la cual veremos a continuación.

1.3.3 Mejora estructural de un clasificador bayesiano

El clasificador bayesiano simple asume que los atributos son independientes dada la clase. Si esto no es verdad, existen dos alternativas básicas. Una es transformar la estructura del clasificador a una red bayesiana, introduciendo arcos dirigidos entre los atributos dependientes, como vimos en la sección anterior. La desventaja es que la simplicidad del NBC se pierde, ya que aprender el modelo y después clasificar nuevos casos llega a ser más complejo. La otra alternativa es transformar la estructura pero mantener una estructura de estrella o una estructura de árbol. Para esto, se introducen tres operaciones básicas:

1. Eliminar un atributo,
2. Unir dos atributos en una nueva variable combinada,
3. Introducir un nuevo atributo que haga que dos atributos dependientes sean independientes (nodo oculto).

La Figura 1.8 ilustra las tres operaciones.

Para aprender el modelo se hace un proceso iterativo, en que se van probando en forma alternada las tres operaciones, de la más sencilla (eliminar un atributo) hasta la más compleja (introducir un nuevo atributo). Esta búsqueda puede ser guiada midiendo la dependencia entre pares de atributos condicionada a la clase, de forma que los pares de atributos con mayor dependencia sean analizados

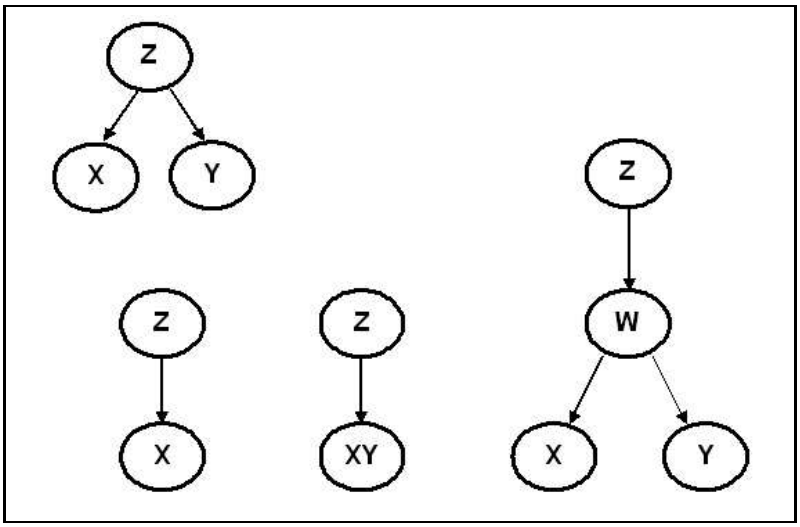

Figure 1.8: Mejora estructural a un clasificador bayesiano simple. Arriba: estructura original. Abajo, de izq. a derecha: eliminaci'on, unión, inserción.

primero. La dependencia se puede medir mediante el cálculo de la información mutua entre pares de atributos X, Y dada la clase C:

$$I(X_i, X_j \mid C) = \sum_{X_i, X_j} P(X_i, X_j \mid C) log(P(X_i, X_j \mid C)/P(X_i \mid C)P(X_j \mid C)) \tag{1.13}$$

En base a lo anterior puede integrarse el siguiente Algoritmo de Mejora Estructural:

1. Obtener la información mutua condicional (IMC) entre cada par de atributos.
2. Seleccionar el par de atributos de IMC mayor.
3. Probar las tres operaciones básicas (i) eliminación, (ii) unión, (iii) inserción.
4. Evaluar las tres estructuras alternativas y la original, y quedarse con la 'mejor' opción.
5. Repetir 2–4 hasta que ya no mejore el clasificador.

Para evaluar las estructuras alternativas pueden usarse dos enfoques. Uno es evaluar el clasificador (con datos de prueba), lo cual en principio es mejor, pero más costoso. El otro enfoque consiste en medir la calidad de la estructura resultante, por ejemplo, basado en el principio de longitud de descripción mínima (MDL), el cual se describe en la siguiente sección. El caso de inserción de un nodo es más complejo (ver sección al final del capítulo de lecturas adicionales),

por lo que puede implementarse el algoritmo usando sólo las opciones de eliminación y unión.

Aunque el proceso de mejora estructural es costoso computacionalmente, se tiene la ventaja de que el clasificador resultante tiene una estructura de árbol, lo cual es muy eficiente para clasificación.

1.4 Aprendizaje de redes bayesianas

El aprendizaje, en general, de redes bayesianas consiste en inducir un modelo, estructura y parámetros asociados, a partir de datos. Este puede dividirse naturalmente en dos partes:

1. Aprendizaje estructural. Obtener la estructura o topología de la red.
2. Aprendizaje paramétrico. Dada la estructura, obtener las probabilidades asociadas.

Veremos primero el aprendizaje paramétrico y luego el estructural.

1.4.1 Aprendizaje paramétrico

Cuando se tienen datos completos y suficientes para todas las variables en el modelo, es relativamente fácil obtener los parámetros, asumiendo que la estructura está dada. El método más común es el llamado *estimador de máxima verosimilitud*, bajo el cual se estiman las probabilidades en base a las frecuencias de los datos. Para una red bayesiana se tienen dos casos:

- Nodos raíz. Se estima la probabilidad marginal. Por ejemplo: $P(A_i) \sim NA_i/N$, donde NA_i es el número de ocurrencias del valor i de la variable A, y N es el número total de casos o registros.
- Nodos hoja. Se estima la probabilidad condicional de la variable dados sus padres. Por ejemplo: $P(B_i \mid A_j, C_k) \sim NB_iA_jC_k/NA_jC_k$, donde $NB_iA_jC_k$ es el número de casos en que $B = B_i$, $A = A_j$ y $C = C_k$ y NA_jC_k es el número de casos en que $A = A_j$ y $C = C_k$.

Dado que normalmente no se tienen suficientes datos, se tiene incertidumbre en las probabilidades estimadas. Esta incertidumbre se puede representar mediante una distribución de probabilidad, de forma que se considere en forma explícita la incertidumbre sobre las probabilidades. Para el caso de variables binarias se modela con una distribución Beta y para variables multivaluadas mediante su extensión, que es la distrubución Dirichlet. Para el caso binario, con una distribución Beta, el valor esperado (promedio) estado por: $P(b_i) = a + 1/a + b + 2$, donde a y b son los parámetros de la distribución.

Esta representación puede utilizarse para modelar la incertidumbre cuando se tienen estimaciones de expertos, cambiando los valores de $a+b$, con el mismo valor estimado. Por ejemplo:

- Ignorancia completa: a=b=0.
- Poco confidente: a+b pequeño (10).
- Medianamente confidente: a+b mediano (100).
- Muy confidente: a+b grande (1000).

También para combinar las estimaciones de expertos con datos. Por ejemplo, para estimar la probabilidad marginal de una variable B:

$$P(b_i) = k + a + 1/n + a + b + 2 \tag{1.14}$$

Donde $a/a+b$ representa la estimación del experto, y k/n la estimación a partir de los datos.

Datos incompletos

En la práctica, en muchas ocasiones los datos no están completos. Hay dos tipos básicos de información incompleta:

Valores faltantes: Faltan algunos valores de una de las variables en algunos casos.

Nodos ocultos: Faltan todos los valores de una variable.

Cuando existen valores faltantes, que es el caso más sencillo, existen varias alternativas, entre ellas:

- Eliminar los casos (registros) donde aparecen valores faltantes.
- Considerar un nuevo valor adicional para la variable, como 'desconocido'.
- Tomar el valor más probable (promedio) de la variable.
- Considerar el valor más probable en base a las otras variables
- Considerar la probabilidad de los diferentes valores en base a las otras variables.

Las primeras dos alternativas pueden ser adecuadas cuando se tienen muchos datos, pero si no, se est de alguna manera desaprovechando información. La tercera alternativa no considera las demás variables del modelo, por lo que normalmente no da los mejores resultados. La cuarta y quinta alternativa son las más interesantes y en general las mejores.

Para el caso del valor más probable se parte de una red bayesiana inicial construida en base a los datos completos. Posteriormente, se complementa el modelo usando los registros con datos incompletos, en base al siguiente algoritmo:

1. Asignar todas las variables observadas en el registro.

2. Propagar su efecto y obtener las probabilidades posteriores de las no observables.

3. Para las variables no observables, asumir el valor con probabilidad mayor como observado.

4. Actualizar las probabilidades previas y condicionales en el modelo.

5. Repetir 1 a 4 para cada observación.

Esto se puede mejorar si, en vez de tomar el valor más probable, se considera cada caso como varios casos *parciales*, en base a la probabilidad posterior de cada valor de la variable faltante.

Cuando existen nodos ocultos, se pueden también estimar las tablas de probabilidad condicional faltantes. El método más comnmente utilizado es el de la 'maximización de la expectación' (EM, por sus siglas en inglés, *expectation maximization*).

El algoritmo EM es un método estadístico muy utilizado para estimar probabilidades cuando hay variables no observables. Consiste básicamente de dos pasos que se repiten en forma iterativa:

Paso E: se estiman los datos faltantes en base a los parámetros actuales.

Paso M: se estiman las probabilidades (parámetros) considerando los datos estimados.

Para el caso de nodos ocultos en redes bayesianas, el algoritmo EM es el siguiente:

1. Iniciar los parámetros desconocidos (probabilidades condicionales) con valores aleatorios (o estimaciones de expertos).

2. Utilizar los datos conocidos con los parámetros actuales para estimar los valores de la variable(s) oculta(s).

3. Utilizar los valores estimados para completar la tabla de datos.

4. Re-estimar los parámetros con los nuevos datos.

5. Repetir 2–4 hasta que no haya cambios significativos en las probabilidades.

La principal limitación de EM es que puede caer en óptimos locales, por lo que los valores finales obtenidos pueden depender de la inicialización.

Discretización de variables continuas

Normalmente las redes bayesianas consideran variables discretas o nominales, por lo que si no lo son, hay que discretizarlas antes de construir el modelo. Aunque existen modelos de redes bayesianas con variables continuas, estos están limitados a variables gaussianas y relaciones lineales.

Los métodos de discretización se dividen en dos tipos principales: (i) no supervisados y (ii) supervisados (Vese el captulo dedicado a discretizacin de variables).

Los métodos de no supervisados no consideran la variable clase, así que los atributos continuos son discretizados independientemente. El método más simple es dividir el rango de valores cada atributo, $[Xmin; Xmax]$, en k intervalos, donde k es dado por el usuario u obtenido usando una cierta medida de información sobre los valores de los atributos.

Los métodos supervisados consideran la variable clase, es decir los puntos de división para formar rangos en cada atributo son seleccionados en función del valor de la clase. El problema de encontrar el número óptimo de intervalos y de los límites correspondientes se puede considerar como un problema de búsqueda. Es decir, podemos generar todos los puntos posibles de división para formar intervalos sobre la gama de valores de cada atributo (donde hay un cambio en la clase), y estimamos el error de clasificación para cada partición posible (usando por ejemplo, *cross validation*). Desafortunadamente, la generación y la prueba de todas las posibles particiones es impráctica, por lo que normalmente se realiza una búsqueda heurística.

El enfoque supervisado se aplica directamente al caso de los clasificadores bayesianos. Por ejemplo, para un atributo continuo, A, se comienza con un número inicial de particiones. Entonces hace una búsqueda iterativa para encontrar una mejor partición, uniendo o particionando intervalos, y probando la exactitud del clasificador después de cada operación. Esta es básicamente una búsqueda 'glotona' (*hill-climbing*), que se detiene cuando la exactitud ya no puede ser mejorada.

Para el caso general de una red bayesiana, existe un método para la discretización de atributos continuos, mientras se aprende la estructura de la red bayesiana. La discretización esta basada en el principio de MDL, considerando el número de intervalos de una variable con respecto a sus vecinos en la red. Para una estructura dada, un procedimiento de búsqueda local encuentra la discretización de una variable que reduce al mínimo la longitud de la descripción referente a los nodos adyacentes en la red, y éste se repite en forma iterativa para cada una de las variables continuas.

1.4.2 Aprendizaje estructural

El aprendizaje estructural consiste en encontrar las relaciones de dependencia entre las variables, de forma que se pueda determinar la topología o estructura de la red bayesiana. De acuerdo al tipo de estructura, podemos dividir los métodos de aprendizaje estructural en:

- Aprendizaje de árboles.
- Aprendizaje de poliárboles.
- Aprendizaje de redes multiconectadas.

Para el caso más general, que es el de redes multiconectadas, existen dos clases de métodos:

1. Métodos basados en medidas y búsqueda.
2. Métodos basados en relaciones de dependencia.

A continuación veremos el método para aprendizaje de árboles y su extensión a poliárboles, para después ver los dos enfoques para aprender redes multiconectadas.

1.4.3 Aprendizaje de árboles

El aprendizaje de árboles se basa en el algoritmo desarrollado por Chow y Liu para aproximar una distribución de probabilidad por un producto de probabilidades de segundo orden (árbol). La probabilidad conjunta de n variables se puede representar como:

$$P(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} P(X_i \mid X_{j(i)}) \tag{1.15}$$

donde $X_{j(i)}$ es el padre de X_i.

Para obtener el árbol se plantea el problema como uno de optimización: obtener la estructura de árbol que más se aproxime a la distribución 'real'. Esto se basa en una medida de la diferencia de información entre la distribución real (P) y la aproximada ($P*$):

$$DI(P, P*) = \sum_{X} P(X) log(P(X)/P*(X)) \tag{1.16}$$

El objetivo es minimizar DI. Se puede definir dicha diferencia en función de la información mutua entre pares de variables, que se define como:

$$I(X_i, X_j) = \sum_{X_i, X_j} P(X_i, X_j) log(P(X_i, X_j)/P(X_i)P(X_j)) \tag{1.17}$$

Se puede demostrar que la diferencia de información es una función del negativo de la suma de las informaciones mutuas (pesos) de todos los pares de variables que constituyen el árbol. Entonces, encontrar el árbol más próximo equivale a encontrar el árbol con mayor peso.

Podemos entonces encontrar el árbol óptimo mediante el siguiente algoritmo, que es equivalente al conocido problema del *maximum weight spanning tree*:

1. Calcular la información mutua entre todos los pares de variables (que para n variables, son $n(n-1)/2$).
2. Ordenar las informaciones mutuas de mayor a menor.
3. Seleccionar la rama de mayor valor como árbol inicial.

4. Agregar la siguiente rama mientras no forme un ciclo, si es así, desechar.

5. Repetir 4 hasta que se cubran todas las variables (n-1 ramas).

El algoritmo NO provee la dirección de los arcos, por lo que ésta se puede asignar en forma arbitraria o utilizando semántica externa (experto).

Para ilustrar el algoritmo consideremos el clásico ejemplo del jugador de golf (o de tenis, segn distintas versiones), en el cual se tienen cuatro variables: *juega, ambiente, humedad y temperatura.* Obtenemos entonces las informaciones mutuas de cada par de variables (10 en total), que se muestran en la Tabla 1.1.

No.	Var 1	Var 2	Info. mutua
1	temp.	ambiente	.2856
2	juega	ambiente	.0743
3	juega	humedad	.0456
4	juega	viento	.0074
5	humedad	ambiente	.0060
6	viento	temp.	.0052
7	viento	ambiente	.0017
8	juega	temp.	.0003
9	humedad	temp.	0
10	viento	humedad	0

Table 1.1: Información mutua entre pares de variables para el ejemplo del golf.

En este caso seleccionamos las primeras 4 ramas y generamos el árbol que se ilustra en la Figura 1.9. Las direcciones de las ligas han sido asignadas en forma arbitraria.

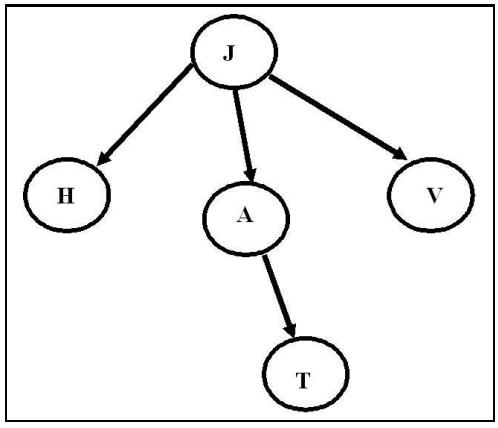

Figure 1.9: Ejemplo de golf. Árbol obtenido mediante el algoritmo de aprendizaje de árboles. J es juega, A es ambiente, H es humedad y T es temperatura.

1.4.4 Aprendizaje de poliárboles

Una forma de darle direcciones al 'esqueleto' aprendido con el algoritmo de Chow y Liu, es mediante pruebas de independencias no sólo entre dos variables, sino entre grupos de tres variables o tripletas. Mediante este esquema se genera un algoritmo que aprende poliárboles, ya que al signar las direcciones puede ser que la estructura generada sea un árbol o un poliárbol (en realidad, un árbol es un caso especial de poliárbol).

El algoritmo parte del esqueleto (estructura sin direcciones) obtenido con el algoritmo de Chow y Liu. Después se determinan las direcciones de los arcos utilizando pruebas de dependencia entre tripletas de variables. Dadas tres variables, existen tres casos posibles:

1. Arcos secuenciales: $X \rightarrow Y \rightarrow Z$.
2. Arcos divergentes: $X \leftarrow Y \rightarrow Z$.
3. Arcos convergentes: $X \rightarrow Y \leftarrow Z$.

Los primeros dos casos son indistinguibles en base a pruebas de independencias; es decir, son equivalentes. En ambos, X y Z son independientes dado Y. Pero el tercero es diferente, ya que las variables X y Z son marginalmente independientes. Este tercer caso lo podemos usar para determinar entonces las direcciones de los dos arcos que unen estas tres variables, y a partir de éstos, es posible encontrar las direcciones de otros arcos utilizando pruebas de independencia. De acuerdo a lo anterior, se establece el siguiente algoritmo para aprendizaje de poliárboles:

1. Obtener el esqueleto utilizando el algoritmo de Chow y Liu.
2. Recorrer la red hasta encontrar una tripleta de nodos que sean convergentes, donde la variable a la que apuntan los arcos la llamaremos *nodo multipadre*.
3. A partir de un nodo multipadre, determinar las direcciones de otros arcos utilizando la prueba de dependencia de tripletas, hasta donde sea posible (base causal).
4. Repetir 2-3 hasta que ya no se puedan descubrir más direcciones.
5. Si quedan arcos sin direccionar, utilizar semántica externa para obtener su dirección.

Consideremos nuevamente el ejemplo del golf, y el esqueleto obtenido (red sin las direcciones), para ilustrar el método. Supongamos que H, J, V corresponden al caso convergente, entonces los arcos apuntan de H a J y de V a J. Después se prueba la dependencia entre H y V respecto a A dado J. Si resulta que H, V son independientes de A dado J, entonces hay un arco que apunta de J a A. Finalmente, probamos la relación de dependencia entre J y T dado A, y si nuevamente fueran independientes, entonces el arco apunta de A hacia T. La Figura 1.10 muestra la estructura resultante.

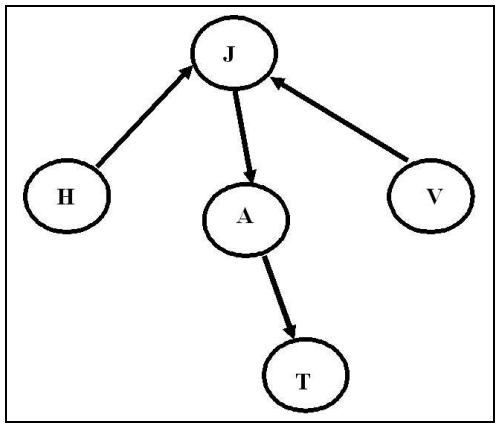

Figure 1.10: Ejemplo de golf. Poliárbol obtenido mediante el algoritmo de aprendizaje.

1.4.5 Aprendizaje de redes

Como comentamos previamente, existen dos clases de métodos para el aprendizaje genérico de redes bayesianas, que incluyen redes multiconectadas. Estos son:

1. Métodos basados en medidas de ajuste y búsqueda.

2. Métodos basados en pruebas de independencia.

A continuación veremos ambos enfoques.

Aprendizaje basado en medidas globales

En esta clase de métodos se tiene una evaluación global de la estructura respecto a los datos. Es decir, se generan diferentes estructuras y se evalúan respecto a los datos utilizando alguna medida de ajuste. Existen diferentes variantes de estos métodos que dependen básicamente de dos aspectos principales: (i) medida de ajuste de la estructura a los datos, y (ii) método de búsqueda de la mejor estructura.

Hay varias posibles medidas de ajuste, las dos más comunes son la medida bayesiana y la medida basada en el principio de longitud de descripción mínima (MDL). La medida bayesiana busca maximizar la probabilidad de la estructura dados los datos, esto es:

$$P(Bs \mid D) \tag{1.18}$$

donde Bs es la estructura y D son los datos. La cual podemos escribir en términos relativos al comparar dos estructuras, i y j, como:

$$P(Bsi \mid D)/P(Bsj \mid D) = P(Bsi, D)/P(Bsj, D) \tag{1.19}$$

Considerando variables discretas y que los datos son independientes, las estructuras se pueden comparar en función del número de ocurrencias (frecuencia) de los datos predichos por cada estructura.

La medida MDL hace un compromiso entre la exactitud y la complejidad del modelo. La exactitud se estima midiendo la información mutua entre los atributos y la clase; y la complejidad contando el número de parámetros. Una constante, α, en [0, 1], se utiliza para balancear el peso de cada aspecto, exactitud contra complejidad. Así, la medida de calidad está dada por:

$$MC = \alpha(W/Wmax) + (1-\alpha)(1 - L/Lmax) \tag{1.20}$$

donde W representa la exactitud del modelo y L la complejidad, mientras que $Wmax$ y $Lmax$ representan la máxima exactitud y complejidad, respectivamente. Para determinar estos máximos normalmente se considera una limitación en cuanto al número de padres máximo permitido por nodo. Un valor $\alpha = 0.5$ da la misma importancia a complejidad y exactitud, mientras que un valor cercano a 0 considera darle mayor importancia a la complejidad y cercano a uno mayor importancia a exactitud.

La complejidad está dada por el número de parámetros requeridos para representar el modelo, la cual se puede calcular con la siguiente ecuación:

$$L = S_i[k_i log_2 n + d(S_i - 1)F_i] \tag{1.21}$$

Donde, n es el número de nodos, k es el número de padres por nodo, S_i es el número de valores promedio por variable, F_i el número de valores promedio de los padres, y d el número de bits por parámetro.

La exactitud se puede estimar en base al 'peso' de cada nodo, en forma análoga a los pesos en el método de aprendizaje de árboles. En este caso el peso de cada nodo, xi, se estima en base a la información mutua con sus padres, Fxi:

$$w(xi, Fxi) = \sum_{xi} P(xi, Fxi) log[P(xi, Fxi)/P(xi)P(Fxi)] \tag{1.22}$$

y el peso (exactitud) total está dado por la suma de los pesos de cada nodo:

$$W = \sum_{i} w(xi, Fxi) \tag{1.23}$$

Una vez establecida una forma de medir la calidad de una estructura, se establece un método para hacer una búsqueda de la 'mejor' estructura entre todas las estructuras posibles. Dado que el número de posibles estructuras es exponencial en el número de variables, es imposible evaluar todas las estructuras, por lo que se hace una búsqueda heurística. Se pueden aplicar diferentes métodos de búsqueda. Una estrategia común es utilizar búsqueda de ascenso de colinas (*hill climbing*), en la cual se inicia con una estructura simple (árbol) que se va mejorando hasta llegar a la 'mejor' estructura. El algoritmo de bsqueda de la mejor estructura es el siguiente:

1. Generar una estructura inicial - árbol.

2. Calcular la medida de calidad de la estructura inicial.
3. Agregar / invertir un arco en la estructura actual.
4. Calcular la medida de calidad de la nueva estructura.
5. Si se mejora la calidad, conservar el cambio; si no, dejar la estructura anterior.
6. Repetir 3 a 5 hasta que ya no haya mejoras.

El algoritmo anterior no garantiza encontrar la estructura óptima, ya que puede llegar a un máximo local. Se pueden utililzar otros métodos de búsqueda como algoritmos genéticos, recocido simulado, búsquedas bidireccionales, etc. La Figura 1.11 ilustra el procedimiento de búsqueda para el ejemplo del golf, inciando con una estructura de árbol que se va mejorando hasta llegar a una estructura final.

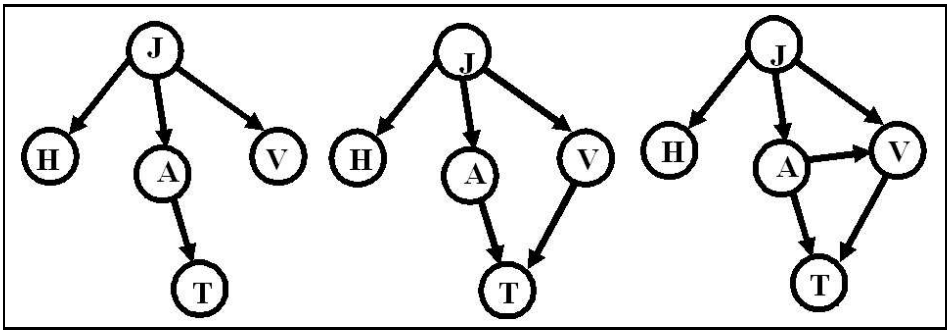

Figure 1.11: Ejemplo de golf. Algunos pasos en la secuencia del aprendizaje de la estructura, partiendo de un árbol (izquierda) hasta llegar a la estructura final (derecha).

Aprendizaje basado en pruebas de independencia

A diferencia del enfoque basado en una medida global, este enfoque se basa en medidas de dependencia local entre subconjuntos de variables. El caso más sencillo es el del algoritmo de Chow y Liu, en el cual se mide la información mutua entre pares de variables. A partir de estas medidas, como se vió previamente, se genera una red bayesiana en forma de árbol. Analizando dependencias entre tripletas de variables, el método se extiende a poliárboles.

Este enfoque se puede generalizar para el aprendizaje de redes multiconectadas, haciendo pruebas de dependencia entre subconjuntos de variables, normalmente dos o tres variables. Por ejemplo, se puede continuar el método de Chow y Liu agregando más arcos aunque se formen ciclos, hasta un cierto umbral mínimo de información mutua. La desventaja es que pueden generarse muchos arcos 'innecesarios', por lo que se incorporan formas de luego eliminar arcos.

Hay diferentes variantes de este enfoque que consideran diferentes medidas de dependencia y diferentes estrategias para eliminar arcos innecesarios.

1.5 Aprendizaje de redes bayesianas dinámicas

1.5.1 Redes bayesianas dinámicas

Las redes bayesianas, en principio, representan el estado de las variables en un cierto momento en el tiempo. Para representar procesos dinámicos existe una extensión a estos modelos conocida como *red bayesiana dinámica* (RBD). Consiste en una representación de los estados del proceso en un tiempo (red estática) y las relaciones temporales entre dichos procesos (red de transición). Se pueden ver como una generalización de las cadenas (ocultas) de Markov.

Para las RBD generalmente se hacen las siguientes suposiciones:

- Proceso markoviano. El estado actual sólo depende del estado anterior (sólo hay arcos entre tiempos consecutivos).
- Proceso estacionario en el tiempo. Las probabilidades condicionales en el modelo no cambian con el tiempo.

Lo anterior implica que podemos definir una red bayesiana dinámica en base a dos componentes: (i) una red base estática que se repite en cada periodo, de acuerdo a cierto intervalo de tiempo predefinido; y (ii) una red de transición entre etapas consecutivas (dada la propiedad markoviana). Un ejemplo de una RBD se muestra en la Figura 1.12.

La inferencia en RBD es en principio la misma que para RB, por lo que aplican los mismos métodos. Sin embargo, la complejidad aumenta, por lo que son más comunes los métodos basados en simulación estocástica, como los métodos MoneCarlo y los filtros de partículas.

1.5.2 Aprendizaje

Dada la representación de una RBD en base a dos componentes, la estructura base y la red de transición, el aprendizaje de RBD puede naturalmente dividirse en el aprendizaje de cada parte por separado:

1. Aprender la estructura base (estática).
2. Aprender la estructura de transición.

Para aprender la estructura base, se consideran los datos de todas las variables en cada tiempo, de forma que sea posible obtener las dependencias entre éstas sin considerar las relaciones temporales. Entonces el problema es equivalente al aprendizaje estructural y paramétrico de una red bayesiana, y se pueden aplicar cualquiera de los métodos vistos anteriormente.

Dada la estructura base, se aprende la red de transición. Esto se puede realizar usando ambos enfoques, tanto el basado en medidas de ajuste y búsqueda,

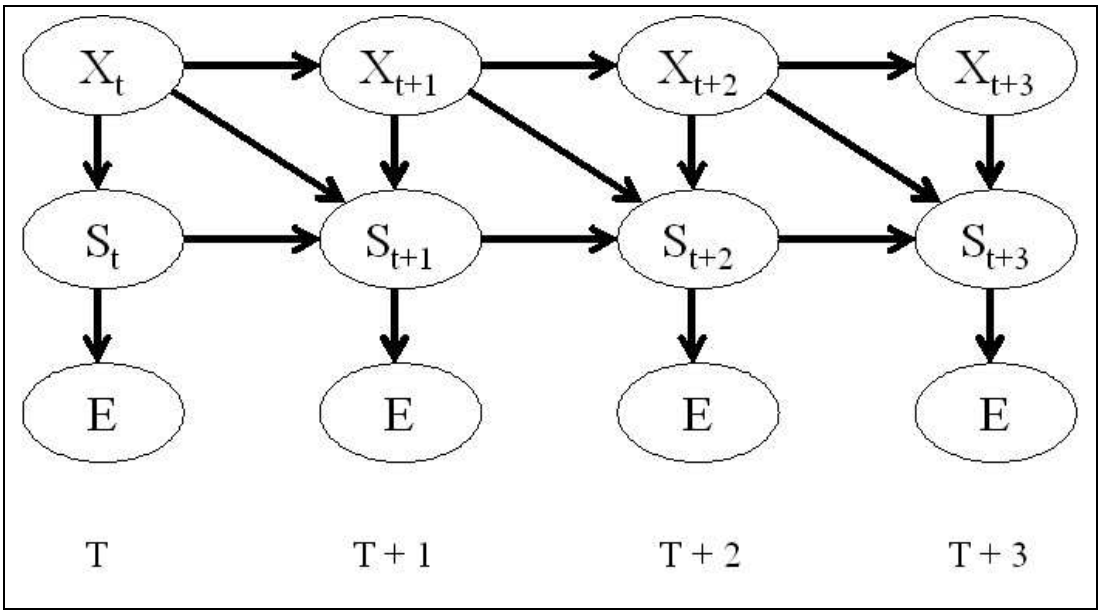

Figure 1.12: Ejemplo de una red bayesiana dinámica. Se muestra la estructura base que se repite en cuatro etapas temporales, así como las relaciones de dependencia entre etapas.

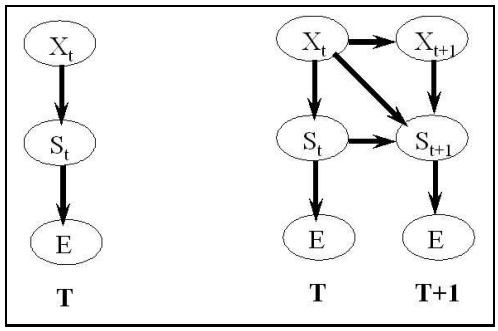

Figure 1.13: Aprendizaje de una red bayesiana dinámica. Primero se obtiene la estructura base (izquierda) y después las relaciones entre etapas (derecha).

como el de medidas locales; con ciertas variantes. Si se utiliza el enfoque basado en búsqueda, se parte de una estructura inicial con dos copias de la red base, y se busca agregar las ligas entre variable en el tiempo T y $T+1$ que optimizen la medida de evaluación. Para ello se consideran los datos de cada variable en un tiempo y el siguiente (de acuerdo al periodo predefinido). Para el enfoque de medidas locales, se aplican éstas a las variables entre etapas para de esta forma determinar los arcos a incluirse en la red de transición. La Figura 1.13 ilustra

el esquema general de aprendizaje de una RBD para un ejemplo sencillo.

1.6 Lecturas adicionales

Las redes bayesianas surgieron en los aos ochenta, y un libro que tuvo un importante impacto es el libro de Judea Pearl [Pearl, 1988], que presenta una muy buena introducción general a redes bayesianas, principalmente en cuanto a representación e inferencia. A partir de entonces se han publicado varios libros de redes bayesianas, entre otros Neapolitan [Neapolitan, 1990] y Jensen [Jensen, 2001]. Recientemente se han publicado algunos libros més enfocados al aprendizaje de redes bayesianas [Borglet and Kruse, 2002, Neapolitan, 2004]. Otra introducción general a aprendizaje en redes bayesianas es el tutorial de Heckerman y otros [Heckerman, 1995].

El algoritmo para propagación en árboles es introducido por Pearl [Pearl, 1986, Pearl, 1988], mientras que el que se basa en la transformación a un árbol de uniones por Lauritzen y Spiegelhalter [Lauritzen and Spiegelhalter, 1988]. Cooper demuestra que el problema de propagación en redes bayesianas es, en general, un problema NP-duro [Cooper, 1990].

Los diferentes enfoques para clasificadores bayesianos se presentan en [Friedman et al., 1997], y una comparación empírica en [Cheng and Greiner, 1999]. La idea de la mejora estructural de clasificadores bayesianos fue introducida originalmente en [Sucar, 1992, Sucar et al., 1993] y posteriormente por [Pazzani, 1996].

En [Pazzani, 1995] se presnta un método para la discertización de atributos en clasificadores bayesianos. Friedman y otros [Friedman and Goldszmidt, 1996] introducen una técnica para disctretizar atributos continuos para al aprendizaje de redes bayesianas en general. Una introducción general a los métodos de discretización se presenta en [Dougherty et al., 1998].

El algoritmo para aprendizaje de árboles fue introducido por Chow y Liu [Chow and Liu, 1968], y posteriormente extendido a poliárboles por Rebane y Pearl [Pearl, 1988]. La idea de usar el principio MDL para aprendizaje de redes bayesianas es originalmente de [Lam and Bacchus, 1994]. Martínez lo aplica para el aprendizaje interactivo de redes bayesianas [Martínez-Arroyo, 1997]. Enfoques alternativos para el aprendizaje estructural de redes bayesianas se presentan en [Cooper and Herskovitz, 1992, Heckerman et al., 1994], entre otros.

Existen diversas herramientas públicas y comerciales para edición, inferencia y aprendizaje de redes bayesianas. Entre éstas se encuentra el sistema Elvira [Elvira Consortium, 2002].

Bibliography

[Borglet and Kruse, 2002] Borglet, C. and Kruse, R. (2002). *Graphical Models: Methdos for Data Analysis and Mining*. Wiley, West Sussex, England.

[Cheng and Greiner, 1999] Cheng, J. and Greiner, R. (1999). Comparing bayesian network classifiers. *Proceedings of the Fifteenth Conference on Uncertainty in Arificial intelligence*, pages 101–108.

[Chow and Liu, 1968] Chow, C. and Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Trans. on Info. Theory*, 14:462–467.

[Cooper, 1990] Cooper, G. (1990). The computational complexity of probabilistic inference using bayesian networks. *Artificial Intelligence*, 42:393–405.

[Cooper and Herskovitz, 1992] Cooper, G. and Herskovitz, E. (1992). A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–348.

[Dougherty et al., 1998] Dougherty, J., Kohavi, R., and Sahami, M. (1998). Supervised and unsupervised discretization of continuous features. *Machine Learning: Proceedings of the Twelfth International Conference on Artificial Intelligence*, pages 194–2002.

[Elvira Consortium, 2002] Elvira Consortium, T. (2002). Elvira: An environment for creating and using probabilistic graphical models. In *Proceedings of the First European Workshop on Probabilistic graphical models (PGM'02)*, pages 1–11, Cuenca, Spain.

[Friedman et al., 1997] Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, 29:131–163.

[Friedman and Goldszmidt, 1996] Friedman, N. and Goldszmidt, M. (1996). Discretizing continuous attributes while learning bayesian networks. *13th International Conference on Machine Learning (ICML)*, pages 157–165.

[Heckerman, 1995] Heckerman, D. (1995). A tutorial on learning with bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, U.S.A.

[Heckerman et al., 1994] Heckerman, D., Geiger, D., and Chickering, D. (1994). Learning Bayesian networks: The combination of knowledge and statistical data. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–94)*, pages 293–301, Seattle, WA.

[Jensen, 2001] Jensen, F. (2001). *Bayesian Networks and Decision Graphs*. Springer-Verlag, New York, U.S.A.

[Lam and Bacchus, 1994] Lam, W. and Bacchus, F. (1994). Learning bayesian belief netwoks: An aproach based on the mdl principle. *Computational Intelligence*, 10:269–293.

[Lauritzen and Spiegelhalter, 1988] Lauritzen, S. and Spiegelhalter, D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society series B*, 50(2):157–224.

[Martínez-Arroyo, 1997] Martínez-Arroyo, M. (1997). *Aprendizaje Interactivo de Redes Bayesianas Multiconectadas*. MSc thesis, Instituto Tecnolgico y de Estudios Superiores de Monterrey - Campus Cuernavaca.

[Neapolitan, 2004] Neapolitan, R. (2004). *Learning Bayesian Networks*. Prentice Hall, New Jersey.

[Neapolitan, 1990] Neapolitan, R. E. (1990). *Probabilistic resoning in expert systems*. John Wiley & Sons, New York, U.S.A.

[Pazzani, 1995] Pazzani, M. J. (1995). An iterative improvement approach for the discretization of numeric attributes in bayesian classifiers. *KDD95*, pages 228–233.

[Pazzani, 1996] Pazzani, M. J. (1996). Searching for atribute dependencies in bayesian classifiers. *In preliminary Papers of the Intelligence and Statistics*, pages 424–429.

[Pearl, 1986] Pearl, J. (1986). Fusion, propagation and structuring in belief networks. *Artificial Intelligence*, 29:241–288.

[Pearl, 1988] Pearl, J. (1988). *Probabilistic reasoning in intelligent systems*. Morgan Kaufmann, Palo Alto, Calif., U.S.A.

[Sucar, 1992] Sucar, L. (1992). *Probabilistic reasoning in knowledge based vision systems*. PhD dissertation, Imperial College of Science, Technology, and Medicine, London U.K.

[Sucar et al., 1993] Sucar, L., Gillies, D., and Gillies, D. (1993). Objective probabilities in expert systems. *Artificial Intelligence*, 61:187–208.