



UNIVERSIDAD
DE CANTABRIA

Representación del Conocimiento

Práctica 3: Inferencia probabilística exacta

Arahí Fernández Monagas

Iker Martínez Gómez

Carlos Velázquez Fernández

Grado en Ingeniería
Informática
Mención en Computación
Curso 2023-2024

1. Fundamentos teóricos

El objetivo de esta práctica es implementar el algoritmo de eliminación de variables. Se busca resolver problemas como la inferencia marginal ($p(a) = \sum_b p(a, b)$), y la inferencia condicional ($p(a|b) = p(a, b)/p(b)$), los cuales son NP-duros. La complejidad de la inferencia depende de la estructura específica de la red bayesiana (RB), concretamente, de las variables X (discretas) y la topología del grafo G .

En el marco de las RB, se introducen los factores, funciones $\phi(a)$ que mapean el dominio de un conjunto de variables aleatorias A a \mathbb{R}^+ . Estos factores no necesariamente representan distribuciones de probabilidad. Las principales operaciones exploradas son:

Producto: Dados dos factores: $\phi_1(a, b)$ y $\phi_2(b, c)$, el factor resultante del producto $\psi(a, b, c)$ se obtiene como:

$$\psi(a, b, c) = \phi_1(a, b) \cdot \phi_2(b, c)$$

donde $\psi(a, b, c)$ será un nuevo factor cuyo ámbito abarcará todas las variables de ϕ_1 y ϕ_2 .

Marginalización: La marginalización de un factor ψ sobre una variable X consiste en la eliminación de esta variable del factor siguiendo la fórmula:

$$\tau(a, c) = \sum_b \psi(a, b, c)$$

Eliminación de variables (EV): Se trata de un proceso en que se multiplican y marginalizan factores de manera iterativa y que permite razonar con inferencias marginal y condicional. Su complejidad computacional está determinada por el orden en que se eliminan las variables. El resultado nos permite inferir la probabilidad de una variable dada una red bayesiana.

Algorithm 1 Eliminación de variables

```

for  $x_i \in X$  (en orden) do
     $\psi_i(\mathbf{w}, x_i) \leftarrow \prod_{j,k} \phi_j(x_i, \mathbf{y}_j) \cdot \tau_k(x_i, \mathbf{z}_k)$ 
     $\tau_i(\mathbf{w}) \leftarrow \sum_{x_i} \psi_i(\mathbf{w}, x_i)$ 
    Sustituye los  $\phi_j$  y  $\tau_k$  por  $\tau_i(\mathbf{w})$ 
end for

```

▷ $\mathbf{W} = \bigcup_{j,k} \mathbf{Y}_j \cup \mathbf{Z}_k$

▷ Marginalización de x_i

▷ Eliminación de x_i

Mínimos Vecinos: Una forma de determinar el orden de eliminación de las variables es mediante este algoritmo. La forma de escoger cada variables es:

$$\arg \min_{X_i} |Ve(X_i)|$$

es decir, seleccionar primero los nodos que menos vecinos tengan.

Mínimos Valores: otra forma de determinar el orden de eliminación es mediante este algoritmo. La forma de escoger cada variables es:

$$\arg \min_{X_i} \prod_{x_j \in Ve(X_i)} |Va(X_j)|$$

En otras palabras, se trata de elegir primero los nodos para los cuales el resultado del producto del número de valores que puede tomar cada vértice vecino sea mínimo.

2. Producto y marginalización de factores

2.1. Producto

Tomando como ejemplo dos factores $\phi_1(a, b)$ y $\phi_2(b, c)$ tal que

| | | |
|-------|-------|-------|
| | A_0 | A_1 |
| B_0 | 0.4 | 0.6 |
| B_1 | 0.1 | 0.9 |

| | | |
|-------|-------|-------|
| | B_0 | B_1 |
| C_0 | 0.5 | 0.5 |
| C_1 | 0.8 | 0.2 |

Cuadro 1: $\phi_1(a, b)$ y $\phi_2(b, c)$

En primer lugar, se genera una nueva tabla para el factor $\psi(a, b, c)$ que abarca todas las combinaciones posibles de valores para la unión de variables presentes en los factores. La probabilidad de cada combinación comenzará vacía.

| | | |
|-----------|-------|-------|
| | A_0 | A_1 |
| $B_0 C_0$ | | |
| $B_0 C_1$ | | |
| $B_1 C_0$ | | |
| $B_1 C_1$ | | |

Cuadro 2: pseudo- $\psi(a, b, c)$

La fase principal del algoritmo implica la iteración sobre las tablas de ambos factores. Para cada fila en la tabla del primer factor, ϕ_1 , se extrae la probabilidad asociada a esa combinación de valores de las variables. En la nueva tabla se buscan las filas cuyos valores coincidan con los de la combinación de ϕ_1 que está siendo explorada y se actualiza la nueva tabla asignando esa probabilidad. Tras realizar este paso, obtenemos una tabla con todas las variables combinadas y las probabilidades respectivas al primer factor.

| | | |
|-----------|-------|-------|
| | A_0 | A_1 |
| $B_0 C_0$ | 0.4 | 0.6 |
| $B_0 C_1$ | 0.4 | 0.6 |
| $B_1 C_0$ | 0.1 | 0.9 |
| $B_1 C_1$ | 0.1 | 0.9 |

Cuadro 3: pseudo- $\psi(a, b, c)$

Este proceso se repite para el segundo factor ϕ_2 , pero en lugar de simplemente actualizar, se realiza una multiplicación de probabilidades. Como resultado, obtenemos finalmente la tabla con todas las variables combinadas y las probabilidades correspondientes multiplicadas.

Conclusión: La función producto genera una nueva tabla con todas las combinaciones posibles a partir del conjunto completo de variables de ambos factores, y luego actualiza esta tabla añadiendo y multiplicando los valores de probabilidad de los factores originales. El resultado es un nuevo factor que representa la distribución conjunta de las variables en los factores originales.

2.2. Marginalización

Dado un factor $\psi(a, b, c)$ como el resultado del apartado 6, la marginalización con respecto a una variable X , supongamos b para este caso, se lleva a cabo de la siguiente manera:

| | A_0 | A_1 |
|-----------|-----------------|-----------------|
| $B_0 C_0$ | $0,4 \cdot 0,5$ | $0,6 \cdot 0,5$ |
| $B_0 C_1$ | $0,4 \cdot 0,8$ | $0,6 \cdot 0,8$ |
| $B_1 C_0$ | $0,1 \cdot 0,5$ | $0,9 \cdot 0,5$ |
| $B_1 C_1$ | $0,1 \cdot 0,2$ | $0,9 \cdot 0,2$ |

| | A_0 | A_1 |
|-----------|-------|-------|
| $B_0 C_0$ | 0.2 | 0.3 |
| $B_0 C_1$ | 0.32 | 0.48 |
| $B_1 C_0$ | 0.05 | 0.45 |
| $B_1 C_1$ | 0.02 | 0.18 |

Cuadro 4: $\psi(a, b, c)$

Primero se crea una nueva tabla a partir de la original, la cual tendrá una variable menos (la variable a marginalizar). Esta tabla contendrá combinaciones de valores repetidas, puesto que el valor que las diferenciaba ha sido eliminado.

| | A_0 | A_1 |
|-----------|-------|-------|
| $B_0 C_0$ | 0.2 | 0.3 |
| $B_0 C_1$ | 0.32 | 0.48 |
| $B_1 C_0$ | 0.05 | 0.45 |
| $B_1 C_1$ | 0.02 | 0.18 |

| | A_0 | A_1 |
|-------|-------|-------|
| C_0 | 0.2 | 0.3 |
| C_1 | 0.32 | 0.48 |
| C_0 | 0.05 | 0.45 |
| C_1 | 0.02 | 0.18 |

Cuadro 5: $\psi(a, b, c)$ y pseudo- $\tau(a, c)$

Después, se recorre esta tabla guardando en un diccionario cada combinación con su respectiva probabilidad. Si la entrada ya existe, las probabilidades se suman. Al terminar se traslada esta información a una nueva tabla en la que cada llave será una combinación y cada valor una probabilidad. Este nuevo factor $\tau(a, c)$ se encuentra ya marginalizado respecto a la variable X .

| | A_0 | A_1 |
|-------|--------------|---------------|
| C_0 | $0,2 + 0,05$ | $0,3 + 0,45$ |
| C_1 | $0,32 + 0,2$ | $0,48 + 0,18$ |

| | A_0 | A_1 |
|-------|-------|-------|
| C_0 | 0.25 | 0.75 |
| C_1 | 0.34 | 0.66 |

Cuadro 6: $\tau(a, c)$

3. Algoritmo de Eliminación de Variables

La función `variableElimination` implementa el algoritmo de Eliminación de Variables (EV) en una red bayesiana, proporcionando un mecanismo eficiente para realizar inferencias marginales y condicionales. Este método es fundamental en la inferencia probabilística y se emplea para inferir la probabilidad de una variable dada en la RB.

El procedimiento comienza eliminando los nodos hoja del grafo, siempre y cuando estos no sean variables observadas o la variable de la cual queremos inferir la probabilidad. La eliminación de hojas ayuda a simplificar el grafo y optimizar el proceso de eliminación posterior.

A continuación, se recorre el grafo almacenando los factores correspondientes a cada nodo en una lista. Si queremos realizar una inferencia condicional, es decir, dadas unas variables observadas, podemos realizar un proceso de simplificación antes de comenzar a operar. Al asignar un valor constante a una variable, podemos reducir el tamaño de las tablas eliminando todas aquellas posibilidades en las que dicha variable tome valores distintos. Una vez hecho esto, comenzamos con la eliminación.

El eje principal del algoritmo radica en esta parte. Siguiendo el orden especificado en el input, para cada variable:

- Se identifican los factores que tienen dicha variable en común.
- Se multiplican estos factores siguiendo las reglas del producto explicadas anteriormente.
- Se marginaliza la variable del nuevo factor.
- Se reemplazan los factores seleccionados por el resultado obtenido en la lista de factores.

Para la inferencia marginal, el resultado alcanzado al terminar es el que buscamos: la probabilidad de la variable. Sin embargo, para la inferencia condicional deberemos dar algún paso extra. Sabiendo que:

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

y si hemos cumplido los pasos anteriores, llegados a este punto hemos calculado $P(A, B)$, resultado que debemos guardar. Para calcular el denominador debemos eliminar una última variable (siguiendo los mismos pasos), la aquella cuya probabilidad queremos inferir. Para finalizar, realizamos la división y obtenemos así la probabilidad condicionada.

4. Casos de Prueba

Para probar el correcto funcionamiento del algoritmo de eliminación de variables, presentamos varios casos, tanto para la inferencia marginal como para la inferencia condicional. En todos los casos, se calcula de forma manual el resultado y se comparan con los obtenidos por el algoritmo teniendo en cuenta una tolerancia de $1 * 10^{-2}$. Si coinciden, el algoritmo funciona de forma correcta.

La forma de comprobar si los resultados eran correctos ha sido la siguiente:

Tomando de ejemplo: $p(b)$ eliminando C, E, D, A . Para calcular el resultado manualmente, se han seguido los pasos mostrados a continuación.

Eliminar C:

$$\psi(c, e) = \phi_5(c, e) \tag{1}$$

$$\tau_1(e) = \sum_e \psi(c, e) = 1 \tag{2}$$

$$p(a, b, d, e) = \phi_1(a, d, e) \cdot \phi_2(b, a) \cdot \phi_3(d) \cdot \phi_4(e) \cdot \tau_1(e) \tag{3}$$

Eliminar E:

$$\psi_2(a, d, e) = \phi_1(a, d, e) \cdot \phi_4(e) \cdot \tau_1(e) \tag{4}$$

$$\tau_2(a, d) = \sum_e \psi_2(a, d, e) \tag{5}$$

$$p(a, b, d) = \phi_2(b, a) \cdot \phi_3(d) \cdot \phi_4(e) \cdot \tau_2(a, d) \tag{6}$$

Eliminar D:

$$\psi_3(a, d) = \phi_3(d) \cdot \tau_2(a, d) \tag{7}$$

$$\tau_3(a) = \sum_d \psi_2(a, d) \quad (8)$$

$$p(a, b) = \phi_2(b, a) \cdot \tau_3(a) \quad (9)$$

Eliminar A:

$$\psi_4(a, b) = \phi_2(b, a) \cdot \tau_3(a) \quad (10)$$

$$\tau_4(b) = \sum_a \psi_4(a, b) \quad (11)$$

$$p(b) = \tau_4(b) \quad (12)$$

Tras aplicar estos pasos, hemos obtenido la siguiente tabla $\tau_4(b)$ con la probabilidad de b .

| b | $\tau_4(b)$ Resultado |
|----------|---------------------------------|
| 0 | 0,4288 |
| 1 | 0,5712 |

Ahora comparamos con lo obtenido por el algoritmo y comprobamos si se ha obtenido el mismo resultado (o uno que cumpla con el máximo de tolerancia, puesto que al calcular los valores a mano siempre se pueden perder decimales). En caso afirmativo, podemos determinar que lo está realizando de forma correcta. Para seguir probando la inferencia marginal, se ha aprovechado la misma red bayesiana, jugando con el orden de las variables para comprobar que compute el mismo resultado.

En cuanto a la inferencia condicional, el procedimiento es muy similar. Se ha calculado siguiendo los pasos explicados en el algoritmo de EV y se han comprobado que todos los resultados fuese correctos. Además, se han implementado ejemplos para analizar el impacto del orden de eliminación en el algoritmo, como se explica en el próximo apartado.

5. Coste temporal empírico

El orden en que eliminemos las variables tiene un impacto significativo en el rendimiento del algoritmo. Dado que el coste temporal del algoritmo de eliminación de variables (EV) es $O(nk^{\max_i |X_i|+1})$, donde:

- $\tau_1(x_1), \dots, \tau_m(x_m)$ son los factores creados al marginalizar.
- $\max_i |X_i|$ depende del orden en que se eliminan las variables.

Una vez hecho este análisis teórico debemos contrastarlo con el análisis empírico. Para ello, hemos propuesto conjuntos de datos de prueba que representan diferentes casos para evaluar el rendimiento en diversas condiciones. Además, exploramos el impacto de la presencia de variables observadas en la eficiencia del proceso de eliminación de variables así como el orden de este.

Para demostrar el correcto funcionamiento del algoritmo para la inferencia marginal, se ha usado el siguiente grafo:

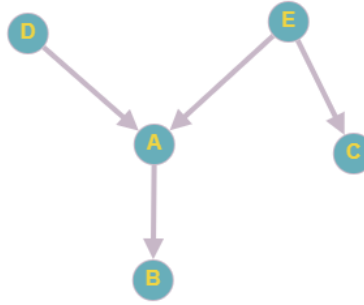


Figura 1: Grafo 1

en donde el número de valores que puede tomar cada variable es

| | | | | |
|---|---|---|---|---|
| A | B | C | D | E |
| 3 | 2 | 3 | 2 | 2 |

Se han probado dos casos cuyos tiempos han sido los siguientes¹:

| Probabilidad | Orden de eliminación | Tiempo (ms) |
|--------------|----------------------|-------------|
| $p(b)$ | D-E-C-A | 9.9183 |
| $p(b)$ | C-E-D-A | 9.7714 |

Cuadro 7: Casos de prueba (inferencia marginal)

Al poder eliminar C por ser un nodo hoja, y comprobar a simple vista que las características de D y E son similares (mismos resultados en los algoritmos de Mínimos Vecinos y Mínimos Valores), podemos entender que la complejidad debería ser la misma, independientemente del orden escogido. Esto se refleja en los tiempos resultantes, puesto que son extremadamente parecidos.

En cuanto a la inferencia condicional, se ha empleado el siguiente grafo:

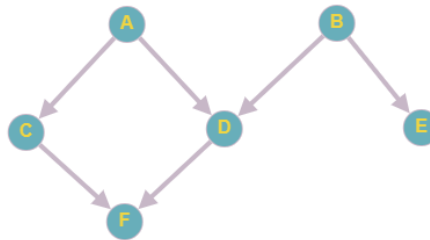


Figura 2: Grafo 2

en donde el número de valores que puede tomar cada variable es

| | | | | | |
|---|---|---|---|---|---|
| A | B | C | D | E | F |
| 2 | 3 | 2 | 3 | 2 | 3 |

¹Tiempo medio después de 100 ejecuciones.

Los resultados obtenidos han sido los siguientes:

| Probabilidad | Orden de eliminación | Tiempo (ms) |
|----------------------|----------------------|-------------|
| $p(f \mid b^2, c^1)$ | E-A-C | 5.9357 |
| $p(f \mid d^1, e^1)$ | B-A-C | 9.6380 |
| $p(f \mid a^0)$ | E-B-C-D | 24.0851 |
| $p(f \mid a^0)$ | E-D-C-B | 78.3170 |

Cuadro 8: Casos de prueba (inferencia condicional)

Los dos primeros ejemplos se han llevado a cabo para verificar el correcto funcionamiento del algoritmo. Tras comprobar que ambos resultados son correctos, podemos suponer que funciona como debería. Los dos siguientes, sin embargo, son más interesantes.

Al inferir la probabilidad de f dado a , podemos eliminar el nodo E puesto que quedaría como vértice hoja. Ahora debemos escoger el orden de eliminación de los nodos B , C y D . Aplicando cualquiera de los dos algoritmos introducidos previamente (Mínimos Valores y Mínimos Vecinos), encontramos que el orden óptimo es B-C-D y, por el contrario, que D-C-B es el peor. El objetivo para no perder eficiencia en el algoritmo de EV es mantener el tamaño de las tablas tan pequeño como sea posible, por ello se busca minimizar las dimensiones de las tablas a multiplicar. Al probar ambos casos, el mejor y peor orden de eliminación, obtenemos una diferencia sustancial en los tiempos resultante. En el peor caso el algoritmo se demora hasta tres veces más que en el mejor. Al tratarse de una RB pequeña, esta diferencia puede parecer insignificante, sin embargo, está claro que para casos reales es algo fundamental a tener en cuenta.