**Table of Contents**



# Semana 1

## Descripción de la inferencia

¿Cómo usar la representación declarativa para responder a las consultas reales.?

### consultas de probabilidad condicional

Evidencia: E = e, conjunto de observaciones.

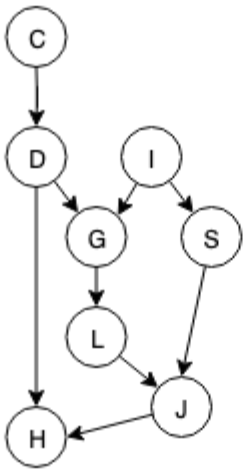
Consulta: un subconjunto de variables y (queremos saber su probabilidad?

Tarea: Calcular P (y | E = e) (NP-hard, son muy caros computacionalmente de resolver)

**Sum-Product en Bayes Network y Markov Network**

Dado un PGM $P_\phi$ (definido por un conjunto de variables X y factores), Valor de $x \in \mathrm{Val}(X)$ y necesitamos calcular $P_\phi(X = x)$

Ejemplo:




**Joint Distribution**

$$\phi_C(C)\phi_D(C,D)\phi_I(I)\phi_G(G,I,D)\phi_S(S,I)\phi_L(L,G)\phi_J(J,L,S)\phi_H(H,D,J)$$

Tenga en cuenta que cada uno de los CPD convierte un factor sobre el alcance de la familia (por ejemplo, $p(g|i,d) -> \phi_G(g,i,d))$.

Si necesitamos saber $P(J)$, necesitamos calcular los siguientes.

$$p(J) = \sum_{C,D,I,G,S,L,H} \phi_C(C)\phi_D(C,D)\phi_I(I)\phi_G(G,I,D)\phi_S(S,I)\phi_L(L,G)\phi_J(J,L,S)\phi_H(H,D,J)$$

Para el Sum-Product de Markov Networks:

unnormalised measure $\longrightarrow$ $\tilde{p}(D) = \sum_{A,B,C} \phi_1(A,B)\phi_2(B,C)\phi_3(C,D)\phi_4(A,D)$

$$p(D) = \frac{1}{Z}\tilde{p}(D)$$ re-normalisation

**Evidencia: Factores reducidos**

$$p(Y|E=e) = \frac{P(Y,E=e)}{P(E=e)}$$

Definamos $W = \{X_1, X_2, \ldots, X_n\} - Y - E$

$$P(Y,E=e) = \sum_w P(Y,W,E=e) = \sum_w \frac{1}{Z}\prod_k \phi_k(D_k, E=e) = \sum_w \frac{1}{Z}\prod_k \phi'_k(D'_k) \propto \sum_w \prod_k \phi'_k(D'_k)$$

we can renormalise the probability at the end

$\prod_k \phi_k(D_k, E=e)$ is the product of factors only thosese which are only the components of those factors that are consistant with the evidence E=e

ejemplo

$$p(J, I=i, H=h) = \sum_{C,D,I,G,S,L,H} \phi_C(C)\phi_D(C,D)\phi_I(I=i)\phi_G(G,I=i,D)\phi_S(S,I=i)\phi_L(L,G)\phi_J(J,L,S)\phi_H(H=h,D,J)$$

Si queremos calcular $P(J|i=i,h=h)$, podemos calcular la ecuación anterior y volver a normalizar.

**Resumen del algoritmo de sum-producto**

$p(Y|E=e) = \frac{P(Y,E=e)}{P(E=e)}$ Y es la consulta

numerados : $p(Y,E=e) = \sum_w \frac{1}{Z}\prod_k \phi'_k(D'_k)$

denominador: $p(E=e) = \sum_Y \sum_w \frac{1}{Z}\prod_k \phi'_k(D'_k)$

En la practica podemos calcular $\sum_w \prod_k \phi'_k(D'_k)$ y renormalizar

Hay muchos algoritmos para computar consultas de probabilidad condicional.

La lista de algoritmos de probabilidad condicional:

Mantener las sumaciones al factor del producto.

- Eliminación variable (programación dinámica)

Mensaje que pasa sobre un gráfico

- Propagación de la creencia (inferencia exacta)
- Aproximaciones variacionales (inferencia aproximada)

Instantámenes aleatorios de muestreo (inferencia aproximada)

- Cadena de Markov Mente Carlo (MCMC)
- Muestreo de importancia

## Mapa (máximo a una inferencia posterior)

Evidencia: E = e, conjunto de observaciones.

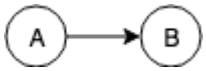
Consulta: un subconjunto de variables y (queremos saber su probabilidad?

Tarea: Calcular $argmax_Y P(Y|E = e)$

Tenga en cuenta que:

1) Puede haber más de una solución posible.

2) Mapa ≠ Max sobre marginales.Ejemplo a continuación:




Los CPD son:

P (A = 0) = 0.4

P (A = 1) = 0.6

P (b = 0 | a = 0) = 0.1

P (b = 1 | a = 0) = 0.9

P (B = 0 | a = 1) = 0.5

P (b = 1 | a = 1) = 0.5

Por lo tanto, podemos obtener la distribución conjunta:

P (B = 0, A = 0) = 0.04

P (B = 1, A = 0) = 0.36

P (b = 0, a = 1) = 0.3

P (b = 1, a = 1) = 0.3

Obviamente, la asignación con la que MAP (A, B) es A = 0, B = 1.Porque su tiene la probabilidad más alta de 0.36.Sin embargo, si observamos las dos variables por separado, la A debería asignarse a A = 1 que es opuesta a la A = 0.Por lo tanto, no podemos vernos por separado en el marginal sobre una y sobre B.

computar MAP es un problema de NP-Hardness.

Dado un PGM $P_\phi$, encuentra una asignación de juntas X con la probabilidad más alta $P_\phi(X)$

**Max-Product**

$$p(J) = argmax_{C,D,I,G,S,L,J,H}\phi_C(C)\phi_D(C,D)\phi_I(I)\phi_G(G,I,D)\phi_S(S,I)\phi_L(L,G)\phi_J(J,L,S)\phi_H(H,D,J)$$

$$p(Y \mid E = e) = \frac{p(Y, E = e)}{p(E = e)} \propto p(Y, E = e), Y = \{X_1, ..., X_n\} - E$$

$p(E = e)$ is constant wrt Y

$$p(Y, E = e) = \frac{1}{Z}\prod_k \phi'_k(D'_k) \propto \prod_k \phi'_k(D'_k)$$

$Z$ is constant wrt Y

$$arg\max_Y p(Y \mid E = e) = arg\max_Y \prod_k (\phi'_k(D'_k))$$

Algoritmos: MAP (también hay diferentes algoritmos para resolver el problema)

Empujar la maximización en el producto del factor

- eliminación variable

Mensaje que pasa sobre un gráfico

- Propagación de la creencia máxima del producto

Usando métodos para la programación enteros (una clase general de optimización que está sobre espacios discretos)

- Para algunas redes: métodos de corte de gráficos.

Búsqueda combinatorial (use técnicas de búsqueda estándar sobre espacios de búsqueda combinatorial)

## Eliminación de variable

El algoritmo de eliminación de variables es un algoritmo de adquisición de conocimiento probabilístico a partir de una red bayesiana. Dada una red bayesiana y una serie de valores observados para ciertas variables, denominadas de evidencia, se obtiene las probabilidades esperadas de una variable de consulta.

El algoritmo trata de hacer uso de diversas técnicas para reducir los cálculos en la medida de lo posible.
El nombre de eliminación de variables proviene de desechar del cálculo de la probabilidad de la variable de consulta a aquellas variables que no tienen ninguna relación de dependencia.

El algoritmo más simple y más fundamental.

## Algrotimo

Eliminación en cadenas


$$p(E) \propto \sum_D \sum_C \sum_B \sum_A \widetilde{P}(A, B, C, D, E)$$ unnormalised measure

$$= \sum_D \sum_C \sum_B \sum_A \phi_1(A, B)\phi_2(B, C)\phi_3(C, D)\phi_4(D, E)$$

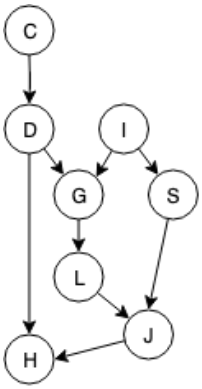
$$= \sum_D \sum_C \sum_B \phi_2(B, C)\phi_3(C, D)\phi_4(D, E) \sum_A \phi_1(A, B)$$

$$= \sum_D \sum_C \sum_B \phi_2(B, C)\phi_3(C, D)\phi_4(D, E)\tau_1(B)$$

...

En última instancia, terminaremos con una expresión que implica solo la variable E.

Eliminación en un BN más complicado.




Goal: p(J)
Eliminate: C,D,I,H,G,S,L

$$= \sum_{L,S,G,H,I,D,C} \phi_J(J, L, S)\phi_L(L, G)\phi_S(S, I)\phi_G(G, I, D)\phi_H(H, G, J)\phi_I(I)\phi_D(C, D)\phi_C(C)$$

$$= \sum_{L,S,G,H,I,D} \phi_J(J, L, S)\phi_L(L, G)\phi_S(S, I)\phi_G(G, I, D)\phi_H(H, G, J)\phi_I(I)\tau_1(D)$$

...

$$= \sum_{L,S,G} \phi_J(J, L, S)\phi_L(L, G)\tau_3(S, G) \sum_H \phi_H(H, G, J)$$

In this case, its equal to 1. $\sum_H p(H \mid G, J) = 1$

...

$$= \sum_{L,S} \phi_J(J, L, S)\tau_5(L, J, S)$$

...

Eliminación variable con evidencia.

Goal: p(J,I=i,H=h)
Eliminate: C,D,G,S,L (no H,I)

reduced factor by evidence

$$= \sum_{L,S,G,L,C} \phi_J(J, L, S)\phi_L(L, G)\phi_S{}'(S)\phi_G{}'(G, D)\phi_H{}'(G, J)\phi_I{}'()\phi_D(C, D)\phi_C(C)$$

then eliminate as before

Si necesitamos calcular P (J | i = i, h = h), simplemente renormalizamos los resultados anteriores.

Eliminación variable en Markov networks

Goal: p(D)
Eliminate: A,B,C

$$= \sum_{A,B,C} \phi_1(A, B)\phi_2(B, C)\phi_3(C, D)\phi_4(A, D)$$

...
at the end, $\tau_3(D) = \tilde{p}(D) \propto p(D)$
we can renormalise to get p(D)

Resumen de algoritmo de eliminación variable

all factors that involve Z

$$\Phi' = \{\phi_i \in \Phi : Z \in Scope[\phi_i]\}$$

then we take all these factors and multiply them:

$$\varphi = \prod_{\phi_i \in \Phi'} \phi_i$$

then we sum out Z:

$$\tau = \sum_{Z} \varphi$$

$$\Phi := \Phi - \Phi' \cup \{\tau\}$$

Reducir todos los factores por evidencia.

- obtener un conjunto de factores $\Phi$

Para cada variable de no consulta z

- Eliminar la variable z de $\Phi$

multiplica todos los factores restantes.

renormalización para obtener distribución

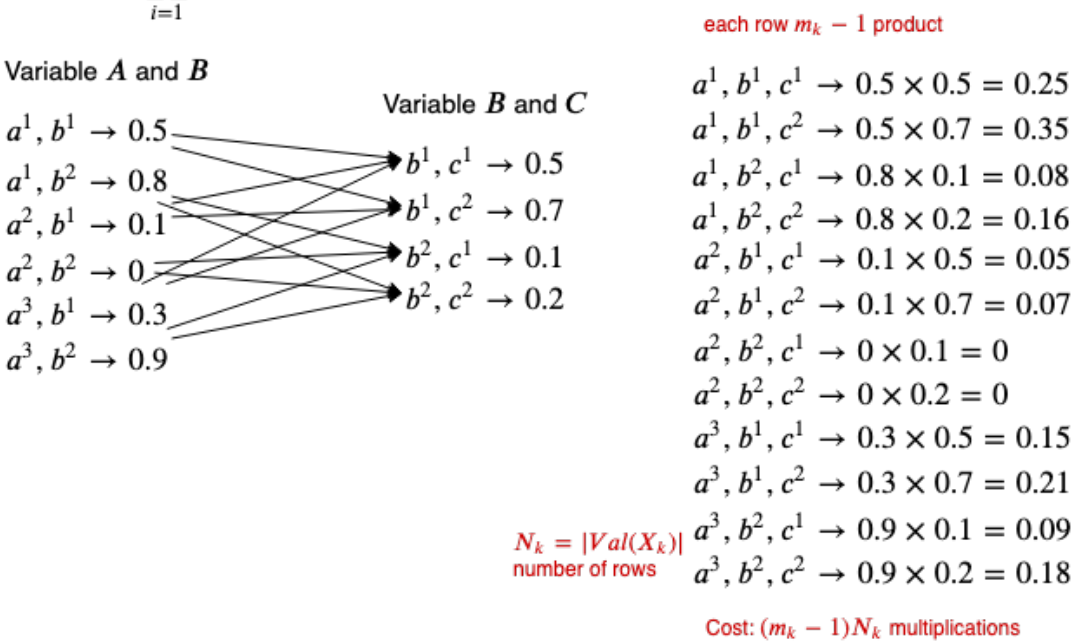
## Complexity of Variable Elimination

$$\varphi_k(X_k) = \prod_{i=1}^{m_k} \phi_i$$ factor product

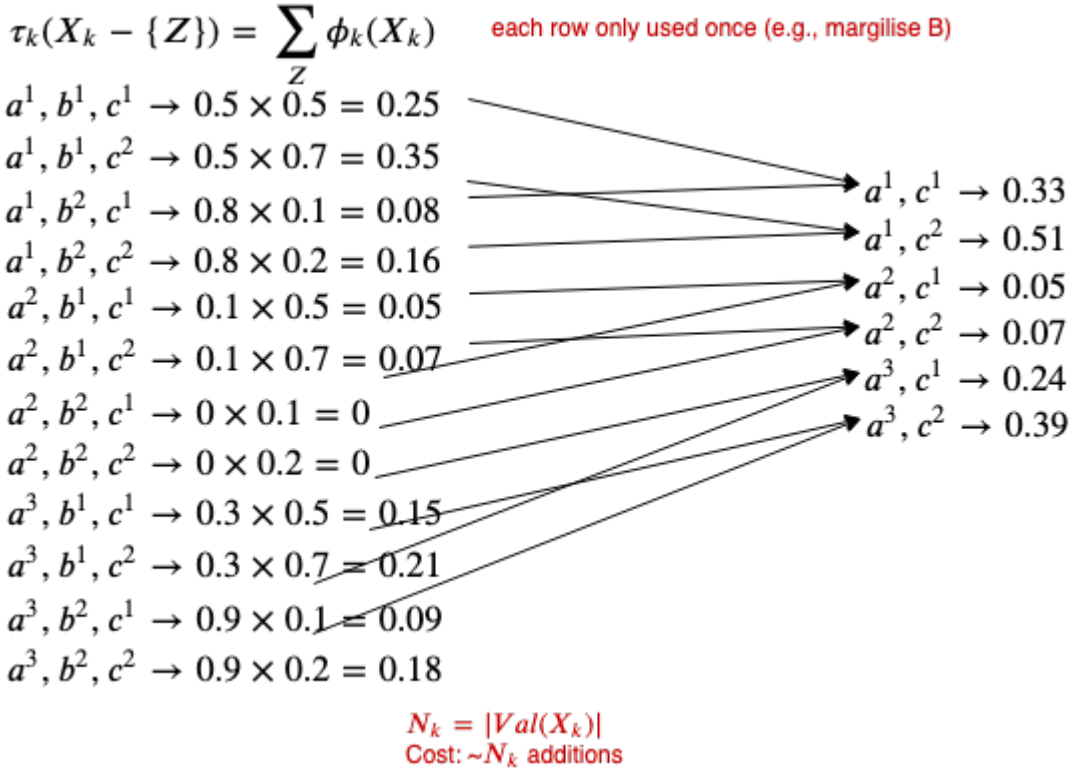
$$\tau_k(X_k - \{Z\}) = \sum_{Z} \phi_k(X_k)$$ margilisation

Producto del factores

$$\varphi_k(X_k) = \prod_{i=1}^{m_k} \phi_i$$




Factor Marginalisation




Detalles:

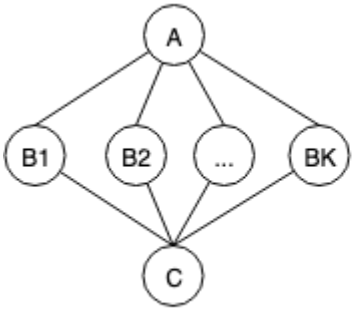
- Comience con M Factores
- - para redes bayesianas, m <= n (el número de variables).m = n cuando un factor para cada variable

- - Para las redes de Markov, M puede ser más grande que el número de variables
- En cada paso de eliminación genera un factor.
- Número total de factores $m^* \leq m + n$
- Operaciones del producto: $\sum_k (m_k - 1) N_k \leq N \sum_k (m_k - 1) \leq N m^*,\ N = \max(N_k)$ =tamaño del factor más grande.Cada factor se multiplica a lo más una vez.
- Operaciones de la suma: $\leq \sum_k N_k \leq N \cdot$Número de pasos de eliminación $\leq$ Nn (El trabajo total es lineal en N y$m^*$)

$N_k = |Val(X_k)| = O(d^{rk})$donde: (explotación exponencial)

- $d = \max(|Val(X_i)|)$D valores en su alcance
- $r_k = |X_k|$ (Número de variables en el factor k-th) = Cardinalidad del alcance del factor K-Th

**La complejidad del algoritmo depende en gran medida de la orden de eliminación**.




**Eliminate A first:**

factor scope: $\{A, B_1, B_2, ..., B_k\}$

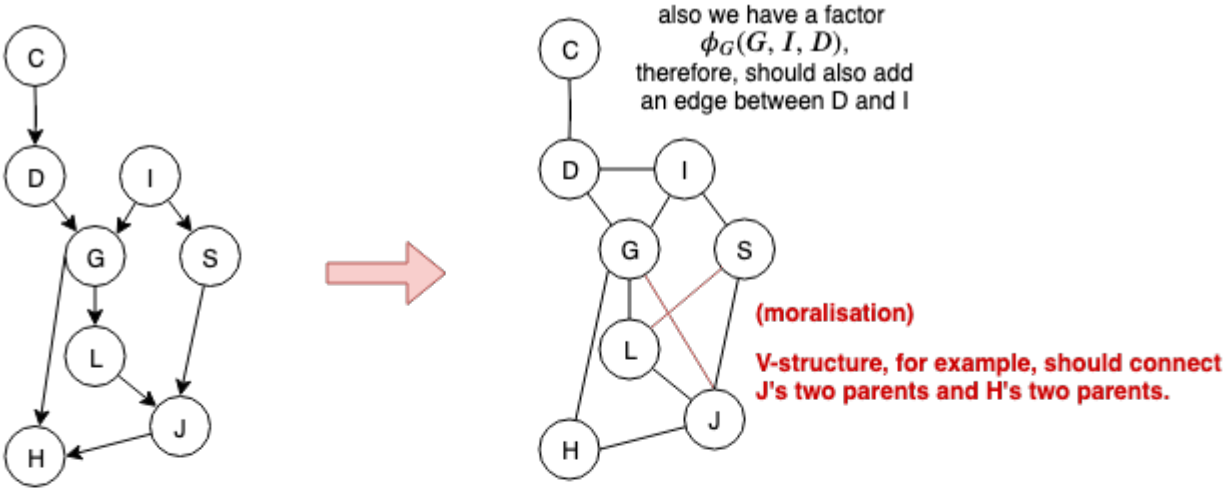
size of factor is exp in k

**Eliminate $B_i$'s first:**

$$\phi_{A_1}(A, B_1)\phi_{B_1}(B_1, C) \rightarrow \tau_1(A, C) \qquad \text{factor scope } \{A, B_1, C\}$$
$$\phi_{A_1}(A, B_2)\phi_{B_2}(B_2, C) \rightarrow \tau_2(A, C)$$
$$\ldots$$
$$\prod_i \tau_i(A, C)$$

## Perspectiva basada en gráficos sobre eliminación variable

Ver un BN como un gráfico no dirigido:




$$\phi_J(J, L, S)\phi_L(L, G)\phi_S(S, I)\phi_G(G, I, D)\phi_H(H, G, J)\phi_I(I)\phi_D(C, D)\phi_C(C)$$

**Eliminate C:**

$$\tau_1(D) = \sum_C \phi_C(C)\phi_D(C, D)$$

$$\phi_J(J, L, S)\phi_L(L, G)\phi_S(S, I)\phi_G(G, I, D)\phi_H(H, G, J)\phi_I(I)\tau_1(D)$$

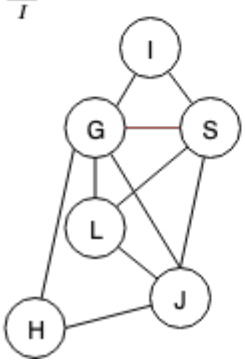
**Eliminate D:**

$$\tau_2(G, I) = \sum_D \phi_G(G, I, D)\tau_1(D)$$

$$\phi_J(J, L, S)\phi_L(L, G)\phi_S(S, I)\phi_H(H, G, J)\phi_I(I)\tau_2(G, I)$$

**Eliminate I:**

$$\tau_3(S, G) = \sum_I \phi_S(S, I)\phi_I(I)\tau_2(G, I)$$




We introduce a new factor $\tau_3(S, G)$
which does not have an edge in the current graph.
We need to introduce this edge between G and S.

The edge is necessary because S and G are together in the same factor.

Todas las variables conectadas a I se conectan directamente directamente para maneter la dependencia

$$\phi_J(J, L, S)\phi_L(L, G)\phi_H(H, G, J)\tau_3(S, G)$$

**Eliminate H:**

$$\tau_4(G, J) = \sum_H \phi_H(H, G, J)$$

$$\phi_J(J, L, S)\phi_L(L, G)\tau_3(S, G)\tau_4(G, J)$$

**Eliminate G:**

$$\tau_5(L, J, S) = \sum_G \phi_L(L, G)\tau_3(S, G)\tau_4(G, J)$$

$$\phi_J(J, L, S)\tau_5(L, J)$$

**Eliminate L,S ...**

Es una orden de eliminación bastante buena en este caso.

Para resumir:

**Gráfico inducido**

El gráfico inducido $I_{\Phi,\alpha}$ sobre factores $\Phi$ y ordenando $\alpha$

- gráfico no dirigido
- $X_i$ y $X_j$ están conectados si aparecen en el mismo factor en una ejecución de la eliminación de la variable (eliminan los factores, generan nuevos factores) algoritmo utilizando $\alpha$ como el ordenamiento

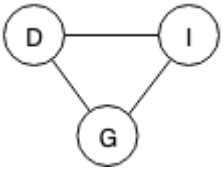
Cliques en el gráfico inducido

Teorema: Cada factor producido durante la Eliminación de  Variables (VE) es una camarilla en el gráfico inducido.

$$\tau_1(D) = \sum_C \phi_C(C)\phi_D(C, D)$$
$$\tau_2(G, I) = \sum_D \phi_G(G, I, D)\tau_1(D)$$
$$\tau_3(S, G) = \sum_I \phi_S(S, I)\phi_I(I)\tau_2(G, I)$$
$$\tau_4(G, J) = \sum_H \phi_H(H, G, J)$$
$$\tau_5(L, J, S) = \sum_G \phi_L(L, G)\tau_3(S, G)\tau_4(G, J)$$
$$\tau_6 = \sum_{L,S} \phi_J(J, L, S)\tau_5(L, J, S)$$

Una camarilla es un subgrafo máximo totalmente conectado.

Ejemplo:




(completamente conectado entre sí)

Es máximo porque no podemos agregar ninguna otra variable a esto y aún así, tener esa propiedad.Por ejemplo, agregue nodo S, pero no se conecta D.

Teorema: Cada clama (máxima) en el gráfico inducido es un factor producido durante la VE.

Ancho inducido:

- El ancho de un gráfico inducido es el número de nodos en la camarilla más grande de la gráfica menos 1.
- Ancho inducido mínimo de un gráfico K es $min_{\alpha}(width(I_{k,\alpha})), \alpha$ sobre todas las posibles órdenes de eliminación.
- Proporciona un límite inferior en el mejor desempeño (la mejor complemento alcanzable) de VE a un modelo de foradura de K (gráfico k)

## Encontrar ordenamientos de eliminación

*Señele codicioso usando la función de costo heurístico

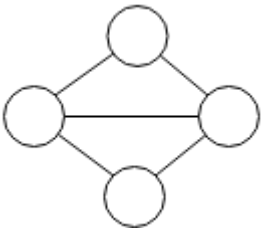
- En cada punto, elimine el nodo con un costo más pequeño.

*Posibles funciones de costo:

- Min-Neighbors: Número de vecinos en gráfico actual
- - Elija el nodo que  tiene el número más pequeño / mínimo de vecinos
- - El correspondiente al factor más pequeño
- - el que cuya cardinalidad (el número de variables en este factor) es más pequeña
- Min-Peso: Peso (por ejemplo, número de valores) de factor formado
- Min-Relling: Número de nuevos bordes de relleno
- Min-Relleno ponderado: peso total de nuevos bordes de relleno (peso del borde = producto de pesos de los 2 nodos)

Teorema: El gráfico inducido está triangulado.

- No hay bucles de longitud > 3 sin un "puente"



- Puede encontrar el orden de eliminación al encontrar una triangulación de gráficos originales de la gráfica original $H_\phi$

## Eliminación variable (resumen)

- Encontrar la ordenación de eliminación óptima es NP-HARD.
- Heurísticas simples que intentan mantenerse inducido el gráfico pequeño a menudo proporcionan un rendimiento razonable.