Table of Contents

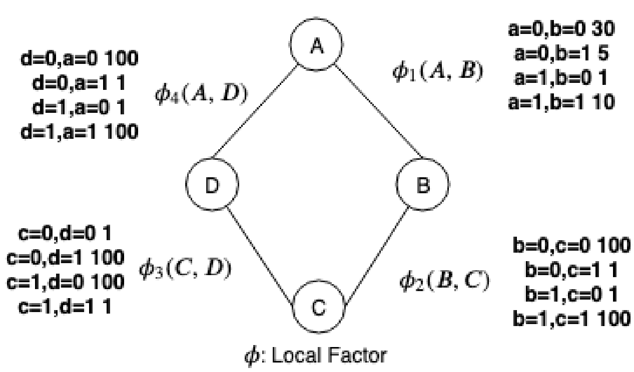
semana 2

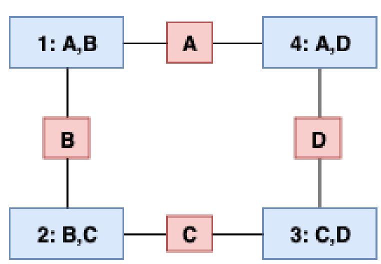
Transmisión de mensajes en gráficos de conglomerados

Algoritmo de propagación de creencias

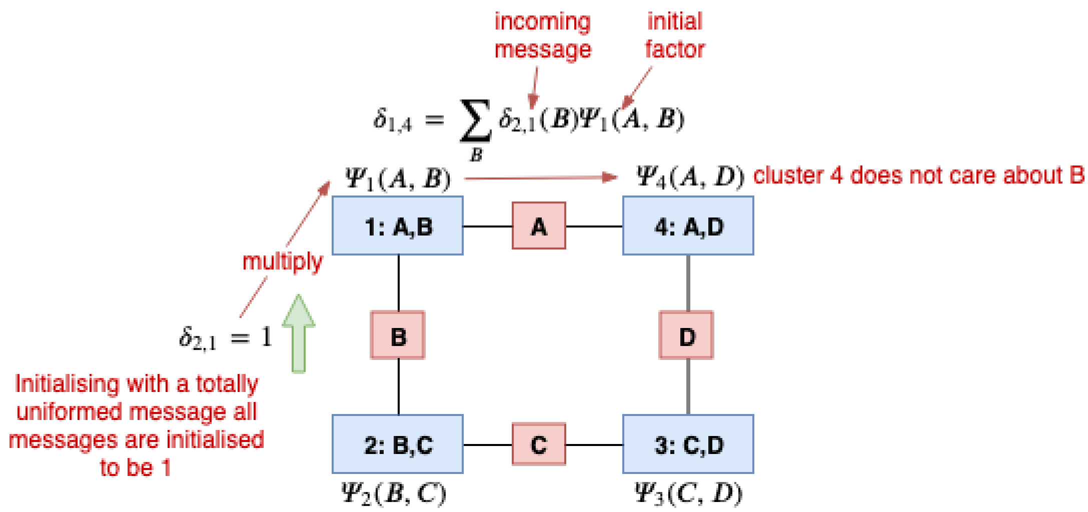
El **algoritmo de propagación de creencias** (en inglés, *belief propagation algorithm*), también conocido como el **algoritmo suma-producto**, es un algoritmo de paso de mensajes para realizar inferencia sobre modelos gráficos tales como redes bayesianas, campos aleatorios de Markov y *factor graph*.

Construye un gráfico de clúster:



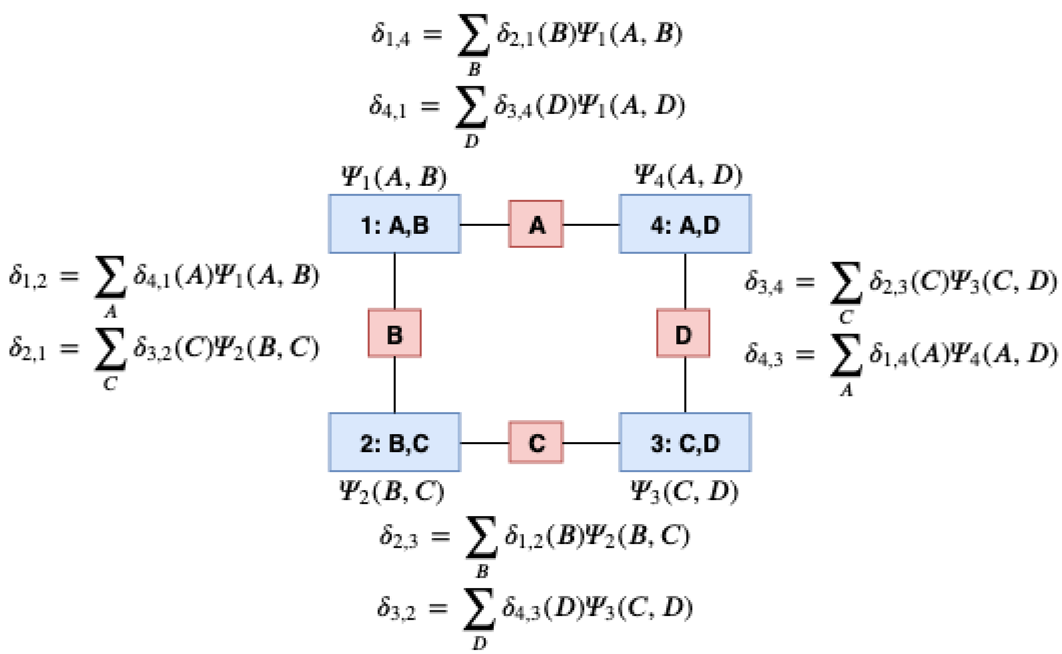


Pasando mensajes:



Clusters: 1,2,3,4

Cada grupo enviará mensajes a sus clústeres adyacentes que reflejen el mismo proceso que arriba.



Gráficos de clúster (definición de generalización)

Gráfico no dirigido

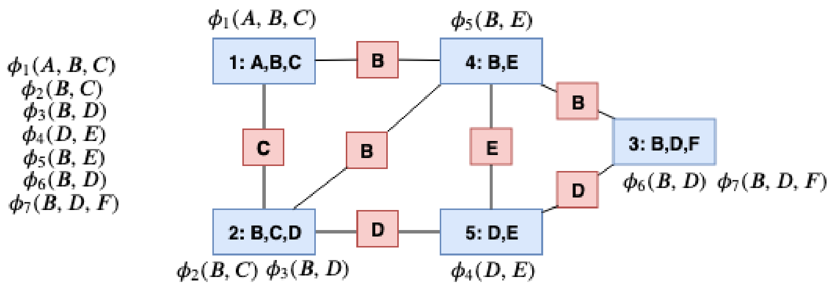
- Nodos: Cluster $C_i \subseteq X_1, \ldots, X_n . C_i$ es un subconjunto de variables
- Bordes: entre C_i y C_j asociados con el subconjunto $S_{i,j} \subseteq C_i \cap C_j, S_{i,j}$ Son las variables que comparten el mensaje.

Dado un conjunto de factores Φ, asigne cada ϕ_k a un clúster, $C_{\alpha(k)}$ (uno y un solo grupo) tenemos que $Alcance[\phi_k] \subseteq C_{\alpha(k)}$

Definamos $\psi_i(C_i) = \prod_{k:\alpha(k)=i} \phi_k$

- Defina una creencia inicial de un clúster en particular el producto de todos los factores que le asignaron
- y algunos grupos podrían no tener factores asignados a ellos, en este caso igual a 1.

ejemplo:

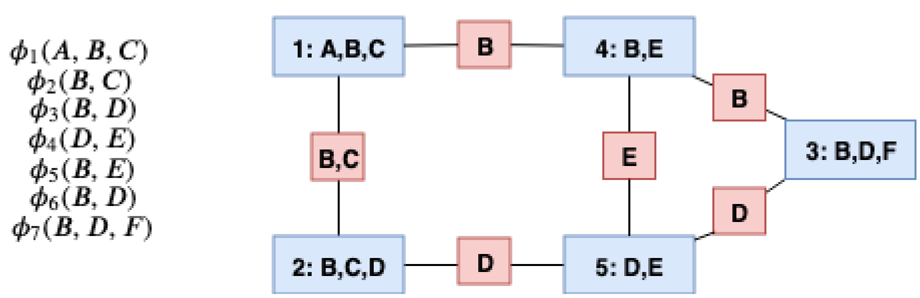


Inicialmente, tenemos que averiguar para cada uno de esos factores un grupo para ponerlo.

- Para a, b, c: solo hay una opción porque solo hay un grupo en el gráfico de enitre que entiende acerca de todos, B y C.
- Para B, C: Sin embargo, tiene dos posiciones.Puede ir al racimo uno o dos.Ambos están bien.
- Para B, D: también tiene dos posiciones.
- Para D, E y B, E: solo tienen una opción.
- Para B, D: Tiene 2 opciones.
- Para B, D, F: solo hay una opción.
- Esta es una forma posible de asignar el clúster, el flujo de factor de influencia probabilística (ensayo activo)

Diferente gráfico de cluster

Para exactamente el mismo conjunto de factores, los grupos no han cambiado, pero el borde cambia.



Paso de mensajes (para el primer gráfico de cluster)

$$\delta_{1\to 4}(B) = \sum_{A,C} \Psi_1(A, B, C)\delta_{2\to 1}(C) \qquad \Psi_1(A, B, C) = \phi_1(A, B, C)$$

$$\delta_{4\to 1}(B) = \sum_{E} \Psi_4(B, E)\delta_{2\to 4}(B)\delta_{5\to 4}(E)\delta_{3\to 4}(B) \qquad \Psi_4(B, E) = \phi_5(B, E)$$

de manera general:

$$S_{i\to j}(S_{i,j}) = \sum_{C_i - S_{i,j}} \Psi_i \times \underbrace{\prod_{k\in(N_i-\{j\})} \delta_{k\to i}}_{\text{all the incoming messages (other from j)}}$$

algoritmo formal

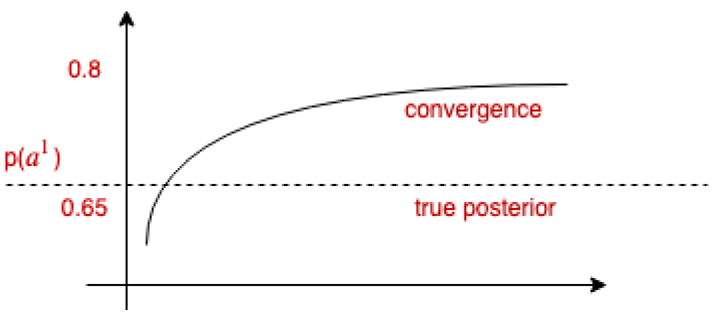
1. asignar cada factor $\phi_k \in \Phi$ a un cluster $C_{\alpha(k)}$
2. Construir potenciales iniciales $\Psi_i(C_i) = \prod_{k:\alpha(k)=i} \phi_k$
3. Inicializa todos los mensajes a 1
4. Repetir: (hasta que la convergencia)
5. * * Seleccione Edge (I, J) y Pase el mensaje. N_i es el vecino.

$$S_{i\to j}(S_{i,j}) = \sum_{C_i - S_{i,j}} \Psi_i \times \underbrace{\prod_{k\in(N_i-\{j\})} \delta_{k\to i}}_{\text{all the incoming messages (other from j)}}$$

6. Cómputar, creencia de este grupo $\beta_i(C_i) = \Psi_i \times \prod_{k\in N_i} \delta_{k\to i}$ N_i es todos los vecinos

Propagación de la creencia (aproximado)

Ejemplo:



Las creencias resultantes son pseudo-marginales, pero el algoritmo realmente se desempeña bien en una gama de aplicaciones prácticas.

Propiedades de los gráficos de clústeres.

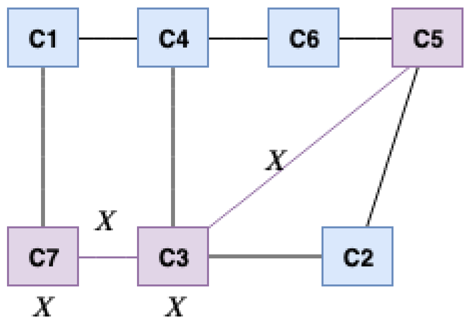
Preservación familiar

Dado un conjunto de factores Φ, asignamos cada ϕ_k a un clúster, $C_{\alpha(k)}$(uno a un solo cluster) tenemos que $Alcance[\phi_k] \subseteq C_{\alpha(k)}$

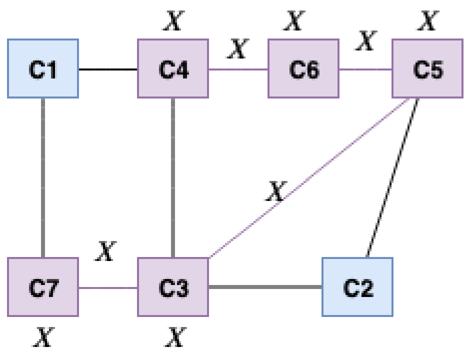
Para cada factor ϕ_k en Φ existe un cluster C_i con $Alcance[\phi_k] \subseteq C_{\alpha(k)}$

Propiedad de intersección de funcionamiento

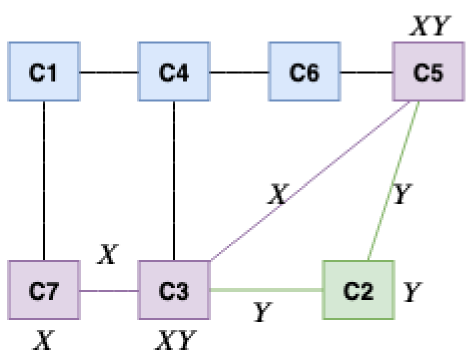
Para cada par de clusters C_i y C_j , y variable $X \in C_i \cap C_j$ existe un camino único entre C_i y C_j para el cual todos los grupos y bordes contienen X.



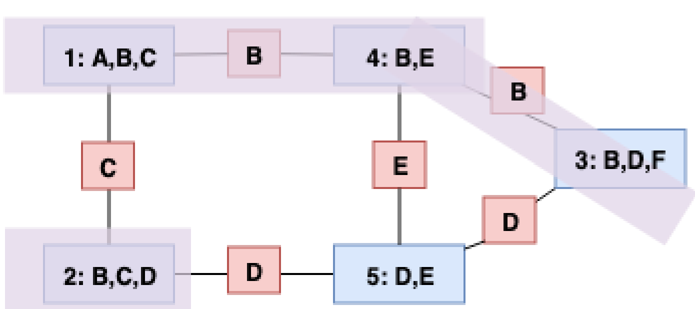
Difusión alternativa de la propiedad de la itersección en ejecución: equivalente: para cualquier variable x, el conjunto de clústeres y sepetes (bordes) que contienen x forman un árbol.(Cada variable indica su propio árbol pequeño en el que la información sobre esa variable fluye en el gráfico).

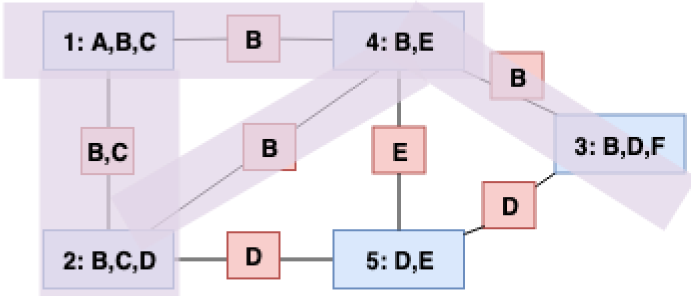


Si X e Y están muy fuertemente correlacionados, BP hace mal el desempeño, debido a los bucles de retroalimentación.Cuanto más sesgó las probabilidades en su modo gráfico, la propagación de creencias más difíciles tiene en términos de los resultados que obtiene.



Ejemplos de gráficos de clústeres ilegales:





Ejemplos de gráficos de clústeres validos:

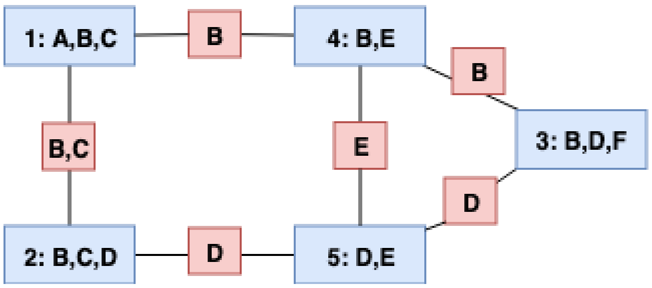
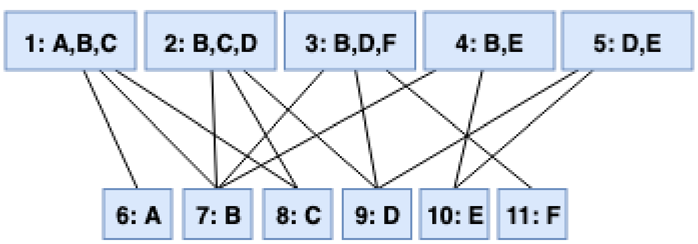


Gráfico de Cluster Bethe

(¿cómo construir un gráfico de clústeres que tiene las propiedades deseadas?)

- Big Cluster: para cada $\phi_k \in \Phi$un factor de cluster $C_k = Alcance[\phi_k]$
- Grupo pequeño: para cada X_i A Singleton Cluster $\{X_i\}$ (variable)
- borde $C_k - X_i$ si $X_i \in C_k$

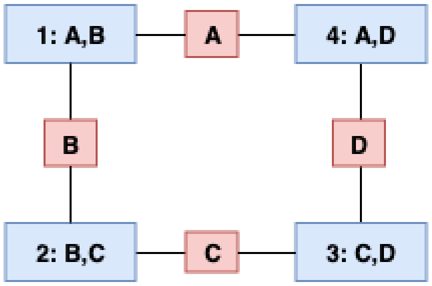


Resumen

- Gráfico de clúster debe satisfacer:
- ** Preservación familiar: permitir que se codifique Φ
- ** Intersección de funcionamiento: conecte toda la información sobre cualquier variable, pero sin los bucles de retroalimentación
- El gráfico de Bethe Cluster es a menudo primero por defecto
- Las estructuras gráficas de grupo más rico (por ejemplo, más bordes) pueden ofrecer diferentes compensaciones. Costo computacional (aumentando la cantidad de tamaños de los mensajes que se pasan que pueden crecer más caros), pero al mismo tiempo implica la preservación de las dependencias, ya que los mensajes se transmiten en el gráfico para que se mantenga más información y no se pierda en este mensaje.proceso de paso.

Propiedades de la propagacion de creencias

Calibración



Cluster Belief:

$$\beta_i(C_i) = \Psi_i \times \prod_{k \in N_i} \delta_{k \to i}$$

For Example:

$$\beta_1(A, B) = \Psi_1(A, B) \times \delta_{4 \to 1}(A) \times \delta_{2 \to 1}(B)$$

Un gráfico de conglomerados se calibra si cada par de conglomerados adyacentes C_i y C_j estan de acuerdo su sepset $S_{i,j}$:

$\sum_{C_i - S_{ij}} \beta_i(C_i) = \sum_{C_j - S_{ij}} \beta_j(C_j)$, en otras palabras, las creencias marginales son iguales.

$\delta_{i \to j}(S_{ij}) = \delta'_{i \to j}(S_{ij})$, igualar el mensaje en el paso de tiempo anterior.

$$\delta'_{i \to j}\left(S_{i,j}\right) = \sum_{C_i - S_{i,j}} \Psi_i \times \prod_{k \in (N_i - \{j\})} \delta_{k \to i} = \sum_{C_i - S_{i,j}} \frac{\beta_i(C_i)}{\delta_{j \to i}}$$

$\delta_{i \to j}\left(S_{i,j}\right)$
(convergence)

$$\beta_i(C_i) = \Psi_i \times \prod_{k \in N_i} \delta_{k \to i}$$

rewrite it:

$$\delta_{j \to i}\left(S_{i,j}\right)\delta_{i \to j}\left(S_{i,j}\right) = \sum_{C_i - S_{ij}} \beta_i(C_i)$$

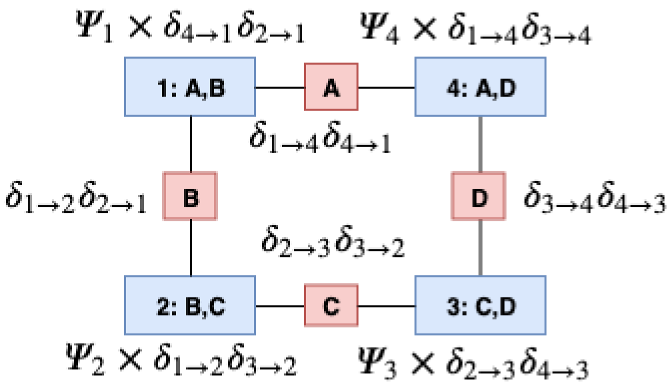
‖ **calibration**

$$\delta_{j \to i}\left(S_{i,j}\right)\delta_{i \to j}\left(S_{i,j}\right) = \sum_{C_j - S_{ij}} \beta_j(C_j)$$

La expresión anterior corresponde a la creencia sepset:

$$\mu_{i,j}(S_{i,j}) = \delta_{j\to i}\delta_{i\to j} = \sum_{C_j - S_{i,j}} \beta_j(C_j)$$

Reparamtrización



Si escribimos de esta forma:

$$\boxed{\frac{\prod_i \beta_i}{\prod_{ij} \mu_{ij}}} = \frac{\prod_i \Psi_i \prod_{j\in N_i} \delta_{j\to i}}{\prod_{ij} \delta_{i\to j}} = \prod_i \Psi_i = \boxed{\tilde{p}(X_1, ..., X_n)}$$

the unnormalised measure

Puede verse que es simplemente la medida no normalizada. Implica que el término / razón es simplemente un conjunto diferente de parámetros que captura la medida original no normalizada que definió nuestra distribución.

Por lo tanto, no hemos perdido información como resultado del algoritmo de propagación de creencias. La distribución sigue ahí, solo un conjunto diferente de parámetros.

Resumen

Las creencias de los gráficos de conglomerados son una parametorización alternativa, calibrada, de la densidad no normalizada original.

- No se pierde información al pasar el mensaje.
- Reparametorizado la distribución original en una forma más conveniente y fácil de usar.

Árboles de clique(camarilla)

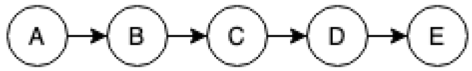
(respuesta más rápida y exacta de la inferencia)

Algoritmo del árbol de camarillas: corrección

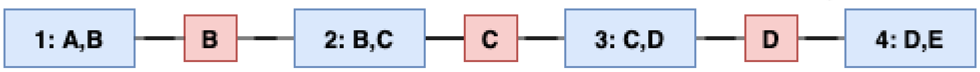
Un caso especial de gráfico de clúster que tiene la garantía de rendimiento. (respuesta de inferencia más rápida y exacta).

Mensaje pasando en árboles

Un árbol de ejemplo simple:



$$\delta_{1\to 2}(B) = \sum_A \Psi_1 \qquad \delta_{2\to 3}(C) = \sum_B \Psi_2 \delta_{1\to 2} \qquad \delta_{3\to 4}(D) = \sum_C \Psi_3 \delta_{2\to 3}$$



$$\Psi_1 \qquad \Psi_2 \qquad \Psi_3 \qquad \Psi_4$$

$$\delta_{2\to 1}(B) = \sum_C \Psi_2 \delta_{3\to 2} \qquad \delta_{3\to 2}(C) = \sum_D \Psi_3 \delta_{4\to 3} \qquad \delta_{4\to 3}(D) = \sum_E \Psi_4$$

Correctness:

Cluster Belief $\beta_3(C, D) = \Psi_3 \delta_{2\to 3} \delta_{4\to 3} = \Psi_3 \left(\sum_B \Psi_2 \sum_A \Psi_1 \right) \sum_E \Psi_4$

product of factors marginalised
out unnecessary variables.

$$\hat{p}(x) = \sum_{x_c \setminus x} \beta_c(x_c).$$

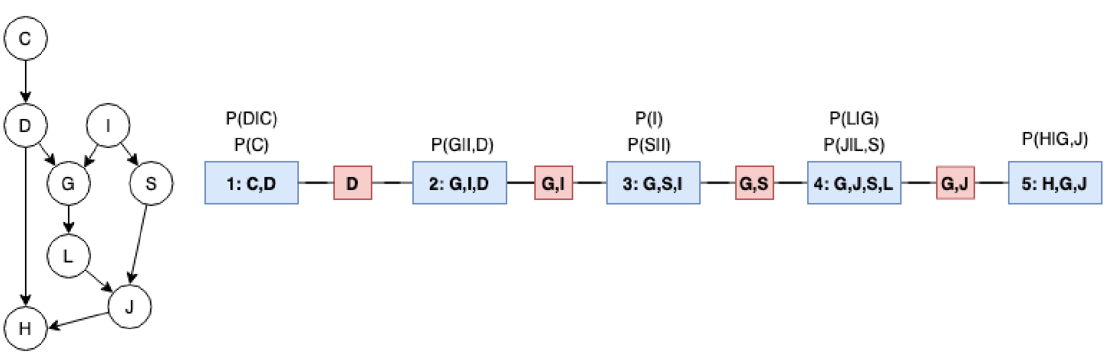
es basicamente eliminación de variables

Como puede ver, nos hemos marginado en un orden legal.

Árbol de camarilla **:** árbol no dirigido tal que:

- los nodos son cluster $C_i \subseteq X_1 \dots X_n$
- cada bode entre C_i y C_j esta asociado con sepset $S_{ij} = C_i \cap C_j$
- Para cada par de grupos C_i y C_j y variable $X \in C_i \cap C_j$ y un camino unico entre estos dos grupos tal que unicamente los grupos intermedios contiene a X

Arbol de camarillas más completo



correccion de arbol de camarilla

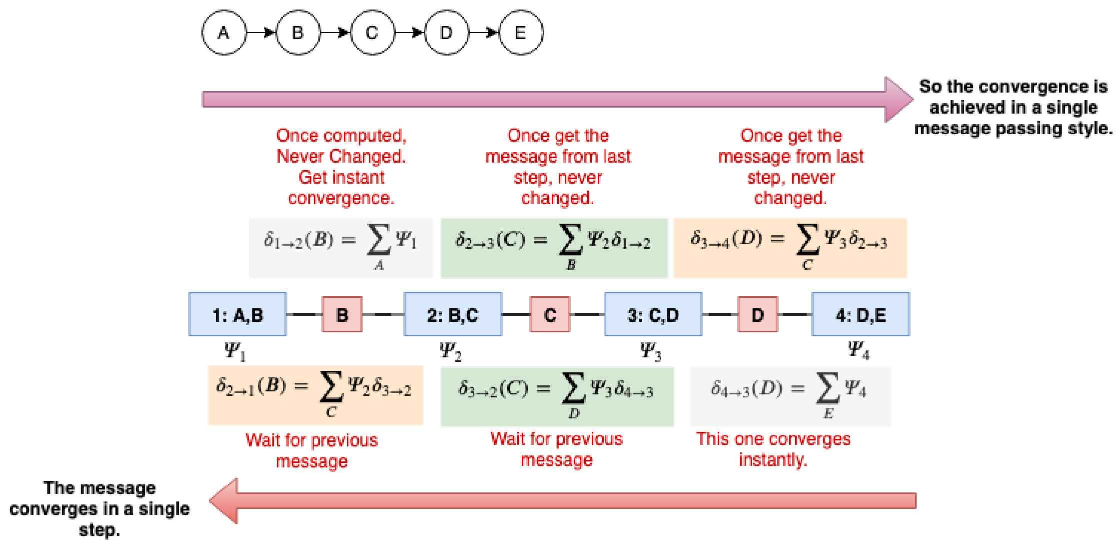
- Si X se elimina cuando pasamos el mensaje $C_i \to C_j$

- Entonces x no aparece en el lado del C_j del árbo

Resumen:

- En este caso, el cálculo es una variante de eliminación variable.
- Se garantiza que las creencias resultantes son los marginales correctos.

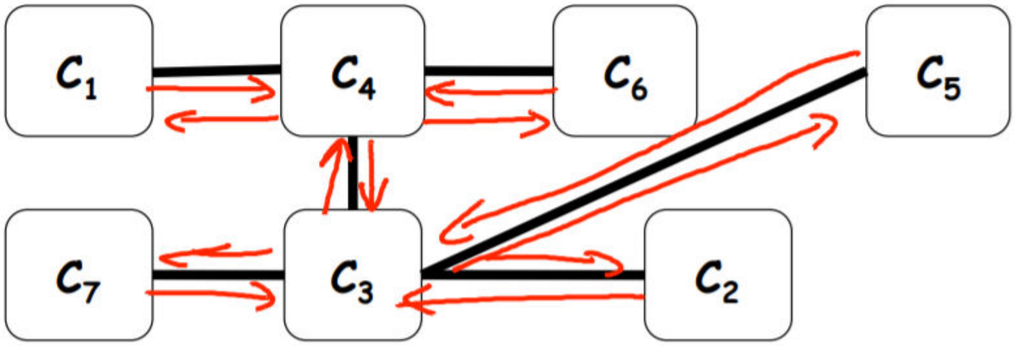
Cálculo



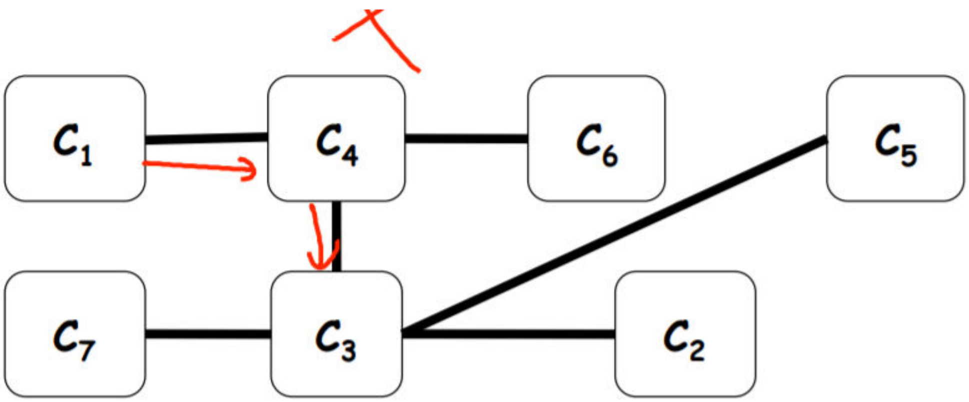
Podemos calcular todos los mensajes de este árbol completo en una ruta en cualquier dirección. Un camino es de izquierda a derecha y un camino de derecha a izquierda. En el contexto de las estructuras de cadena, es posible que vea esto bajo el nombre **de algoritmo hacia adelante y hacia atrás** . Esto se usa muy comúnmente en cosas como el modelo oculto de Markov y otras representaciones estructuradas en cadena similares., el mensaje que se va a enviar tiene que

- Una vez que C_i recibe un mensaje final de todos los vecinos, excepto CJ, entonces $\delta_{i\to j}$ es también final (nunca cambiará)
- Los mensajes de las hojas son inmediatamente definitivas, (las esquinas)
- Puede pasar mensajes de hojas hacia adentro.
- Si los mensajes se pasan en el orden correcto, solo necesita pasar 2(K-1) mensajes

orden correcto



orden incorrecto



Resolviendo consultas

Consultas de distribución posteriores en las variables que aparecen juntas en clique

- resuma variables irrelevantes de cualquier camarilla que contiene esas variables

Presentamos la nueva evidencia Z = z y consulta x

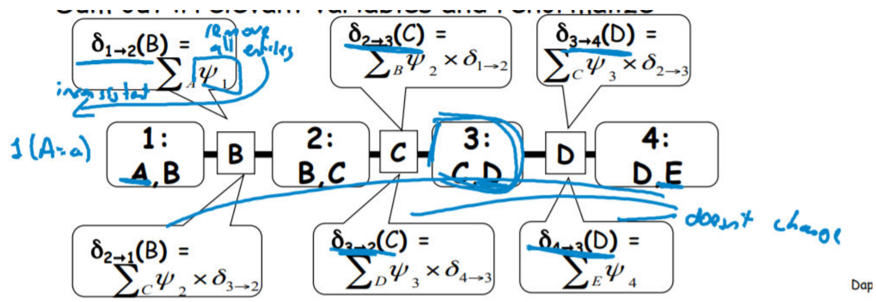
- - Si x aparece en clique con Z
- • Multiplique la clique que contiene x y z con la función indicadora I(Z = z)
- • Sumar las variables irrelevantes y renormalizar

Presentamos la nueva evidencia z = z y consulta x if x no comparte una camarilla con z

- Multiplica 1 (z = z) en alguna camarilla que contiene z

- propagar mensajes a lo largo del camino a la camarilla que contiene x

- resume las variables irrelevantes y renormalizar



independencia

running intersection property (RIP)

Propiedad de intersección de funcionamiento

• Para un borde (i, j) en t, sea:

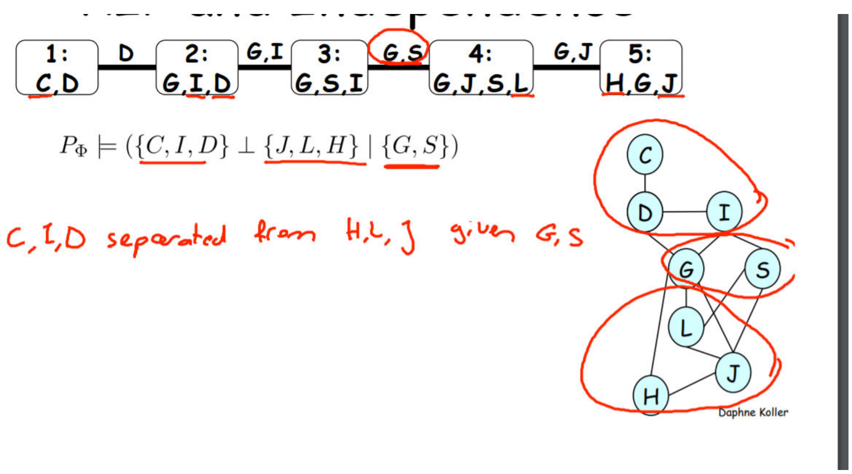
- $W_{<(i,j)}$ = todas las variables que aparecen solo en el lado C_i de t

- $W_{<(j,i)}$ = todas las variables que aparecen solo en el lado de C_j

- Las variables en ambos lados están en el Sepset $S_{i,j}$

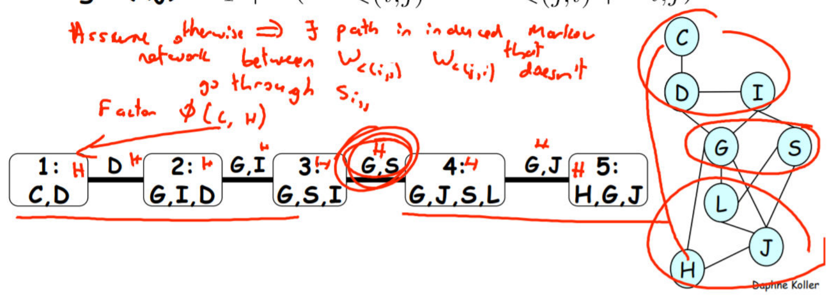
• Teorema: t satisface la RIP si y solo si, para cada (i,j)

$P_\Phi \models (W_{<(i,j)} \perp W_{<(j,i)} | S_{i,j})$



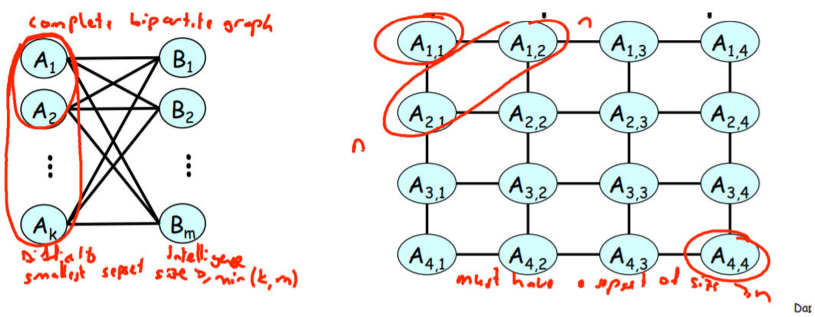
C,I,D estan d-separados de H,L,J dado G,S

- **Theorem:** T satisfies RIP if and only if, for every edge (i,j) $P_\Phi \models (\boldsymbol{W}_{<(i,j)} \perp \boldsymbol{W}_{<(j,i)} \mid \boldsymbol{S}_{i,j})$



implicaciones

- Cada Sepset necesita separar el gráfico en dos partes condicionalmente independientes



Resumen

• La corrección de la inferencia del árbol de la camarilla se basa en la propiedad de intersección de funcionamiento

• La propiedad de intersección en ejecución implica la separación en la distribución original.

• Implica complejidad mínima incurrida por cualquier árbol de clique:

- - relacionado con el ancho inducido por el gráfico mínimo

Arbol de camarilla y eliminación de variable

• eliminación variable

- - Cada paso crea un factor λ_i a través del producto
- - una variable se elimina en λ_i para generar un nuevo factor τ_i
- - π_i se utiliza en computación Otros factores λ_i

• Vista de árbol de clique

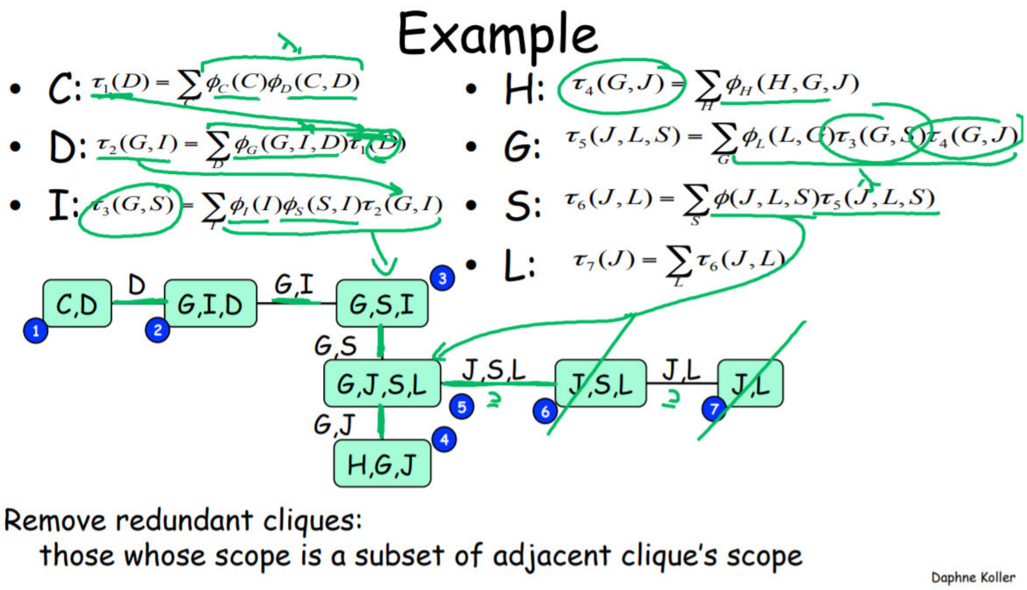
- - Los factores intermedios λ_i son camarillas.

- - τ_i son "mensajes" generados por la clique λ_i y transmitidos a otra camarilla λ_j

Árbol de clique de eliminación de variable

- eliminación de variable define un gráfico

    - - clúster C_i para cada factor λ_i utilizado en el cálculo
    - - dibuja un borde de $C_i - C_j$ si el factor generado a partir de λ_i se usa en el cálculo de λ_j



Propiedades del árbol(a partir de VE)

• El proceso de eliminación de variable induce un árbol.

- - En variable, cada factor intermedio se usa solo una vez
- - Por lo tanto, cada grupo "pasa" un factor (mensaje) a un otro grupo

• El árbol es la preservación familiar:

- - Cada uno de los factores originales debe usarse en algún paso de eliminación.
- - y por lo tanto contenido en el alcance de los λ_i asociados

El árbol obedece la propiedad de intersección en ejecución

- Si $X \in C_i$ y $X \in C_{ij}$, , entocnes X está en cada clúster en la ruta (única) entre C_i y C_j

Propiedad de intersección de funcionamiento

• Teorema: Si T es un árbol de clústeres inducido por VE, entonces t OBEYS RIP

Resumen

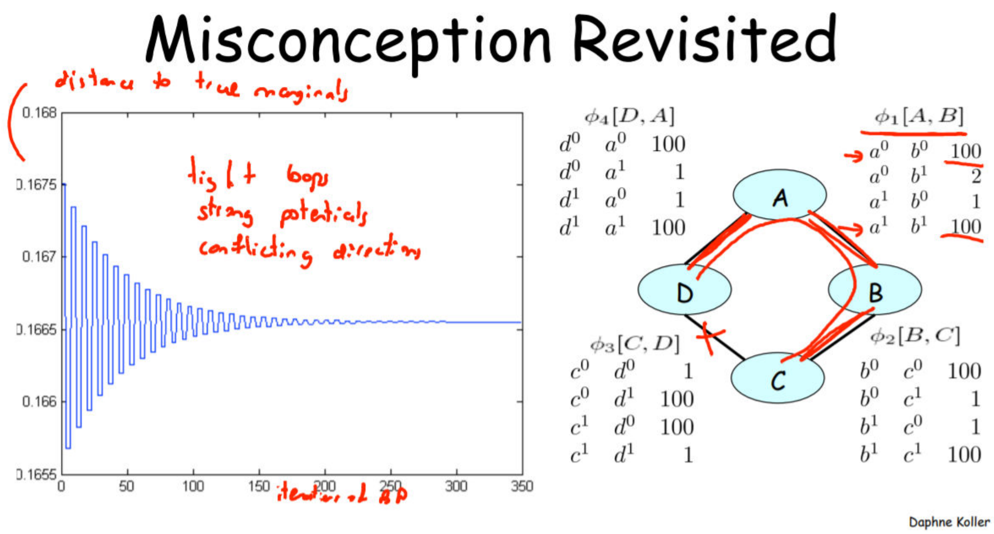
• Una ejecución de eliminación variable define implícitamente un árbol de clique correcto

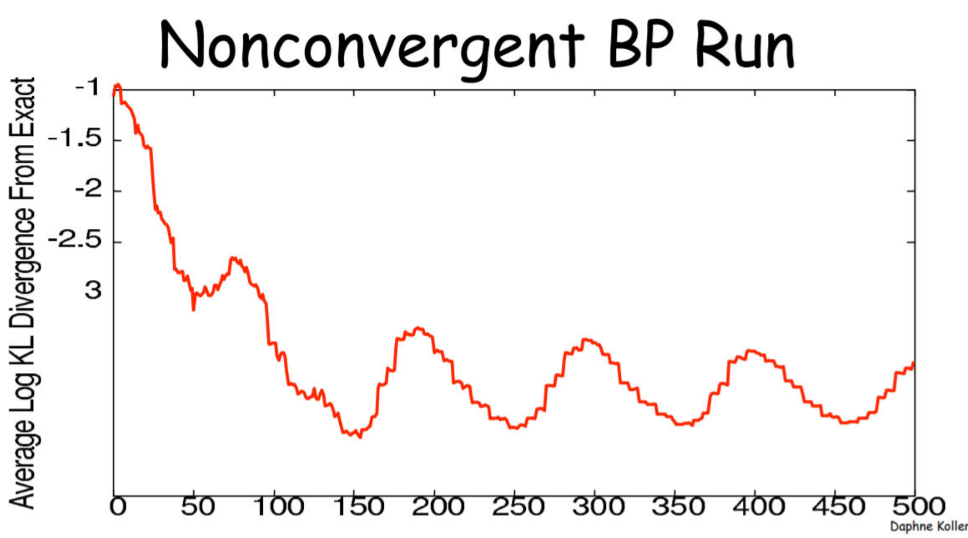
- Podemos "simular" una carrera de VE para definir camarillas y conexiones entre ellos.

• El costo de la eliminación variable es ~ igual que los mensajes que pasan en una dirección en el árbol

• Los árboles de clique utilizan programación dinámica (almacenando mensajes) para calcular los marginales sobre todas las variables a solo el doble del costo de ve

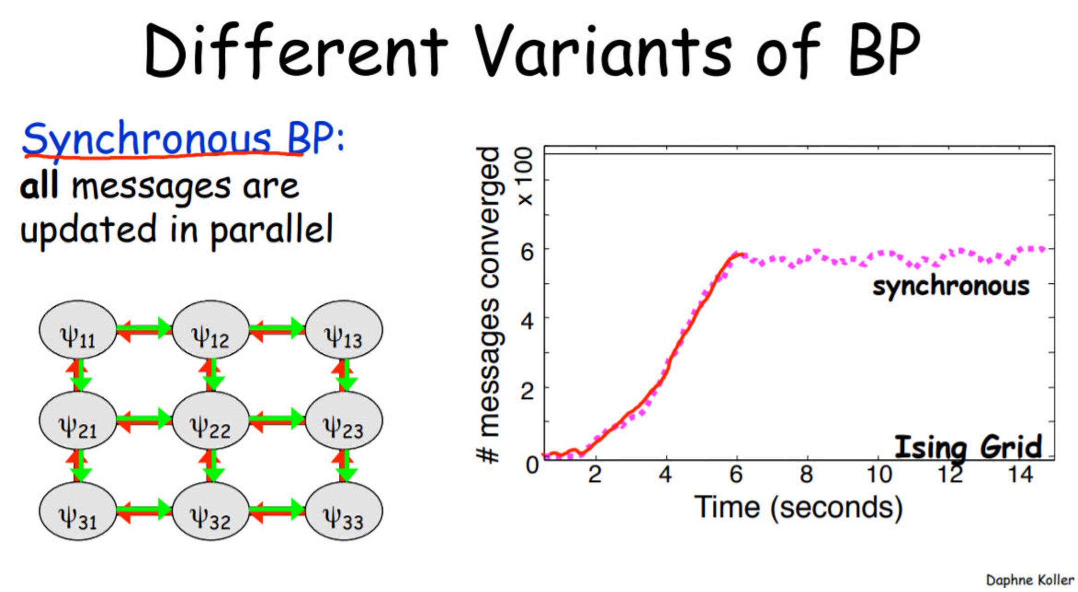
En practica



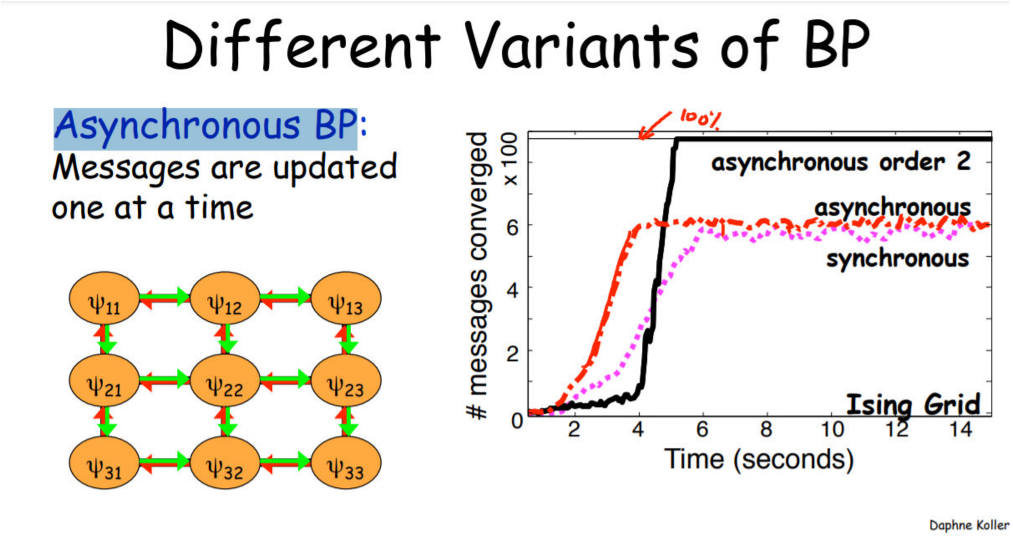


Diferentes variantes de BP

Synchronous BP: Todos los mensajes son actualizados en paralelo



Asynchronous BP: los mensajes son actualizados uno a la vez



Observaciones

• Convergencia es una propiedad local:

- Algunos mensajes convergen pronto.

- otros nunca pueden converger

• Synchronous BP converge considerablemente peor que asynchronous

• El orden de propagacion del mensaje hace una diferencia en el alcance y la velocidad de convergencia

Programación de mensajes informados

• Reparameterización del árbol (TRP)

- Elija un árbol y pase mensajes para calibrar.

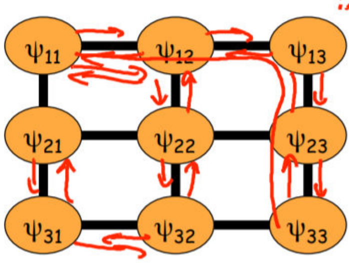
• Propagación de creencias residuales (RBP)

- Pasar mensajes entre dos grupos cuyas creencias sobre el sepset no están de acuerdo.

Informed Message Scheduling

- Tree reparameterization (TRP)
  - Pick a tree and pass messages to calibrate

trees must cover all edges
improves performance if trees are larger (spanning trees)

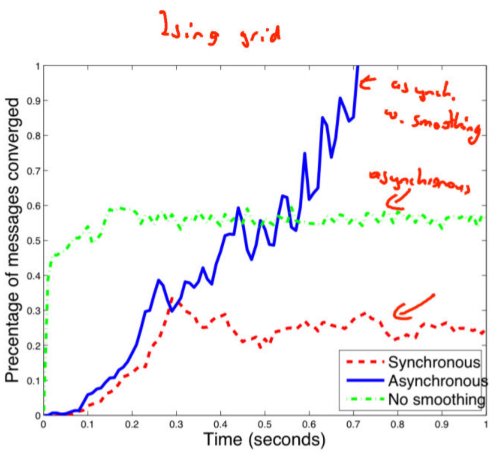


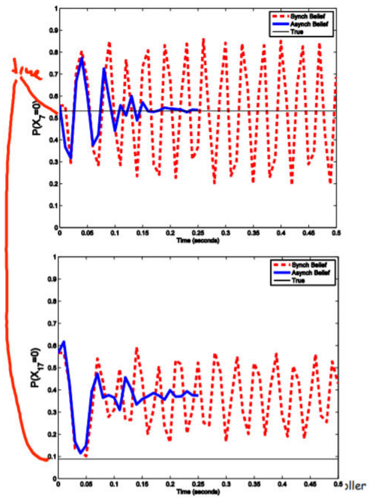
Smoothing (Damping) Messages

$$\delta_{i\to j} \leftarrow \sum_{C_i - S_{i,j}} \psi_i \prod_{k\neq j} \delta_{k\to i}$$

$$\delta_{i\to j} \leftarrow \lambda \underbrace{\left(\sum_{C_i - S_{i,j}} \psi_i \prod_{k\neq j} \delta_{k\to i} \right)}_{\text{new msg}} + (1-\lambda)\underbrace{\delta^{\text{old}}_{i\to j}}_{\text{old msg}}$$

- Dampens oscillations in messages





Resumen

- Para lograr la convergencia de BP, dos trucos principales.
    - - amortiguación
    - - orden de paso de mensajes inteligentes

- La convergencia no garantiza la corrección.
- Malos casos para BP, tanto convergencia como de precisión:
    - - Potenciales fuertes que tiran de diferentes direcciones.
    - - Bucles apretados

- Algunos algoritmos nuevos tienen mejor convergencia:
    - - Optimización basada en la inferencia.