Modèles Graphiques non-orientés - Elimination	**2009/2010**
Cours 4 — 21 Octobre 2009	
Enseignant: Francis Bach	*Scribe: Islem Rekik, Sixin Zhang*

4.1 Modèles Graphiques non-orientés

Définition 4.1 *Soient $G = (V, E)$ un graphe non orienté, $\mathcal{C}$ l'ensemble des cliques (maximales) de G, alors la famille des lois associées à G, $\mathcal{L}(G)$ s'écrit :*

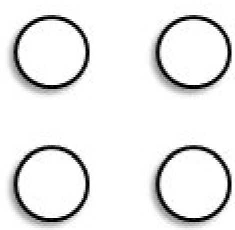
$$\mathcal{L}(C) = \left\{ loi\ p(x) \mid p(x) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C), \psi_C \geqslant 0 \right\}$$

où Z est une constante de normalisation:

$$Z = \sum_x \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

Dans le cas d'indépendance, on a

Proposition 4.2 *Si $E = \emptyset$, alors $p(x) = \prod_{i=1}^n \psi_i(x_i)$.*



Cas d'indépendance

Quand la loi $p(x) \in \mathcal{L}(G)$, on dit que $p(x)$ se factorise dans G. On peut s'étendre cette terminologie et on dit que la fonction $u(x)$ se factorise dans G si

$$u(x) = \prod_{C \in \mathcal{C}} \psi_C(x_C).$$

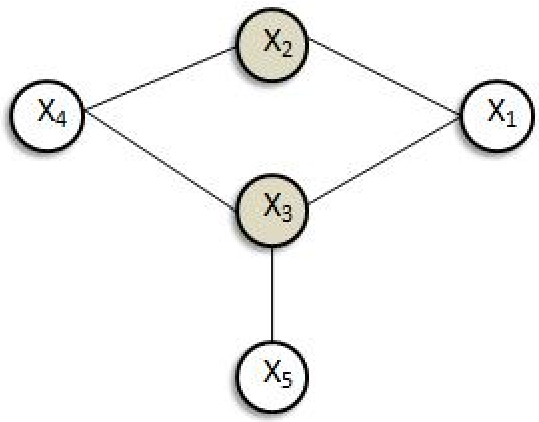


Figure 4.1. Dans ce graphe, $p(x)$ se factorise sous la forme: $\psi_{12}(x_{12})\psi_{24}(x_{24})\psi_{43}(x_{43})\psi_{31}(x_{31})\psi_{35}(x_{35})$.

les ψ_C sont définies à une constante près.

Dans la définition de la factorisation dans le modèle graphique non orienté, on peut prendre pour $\mathcal{C}$, soit l'ensemble des cliques, soit l'ensemble des cliques maximales. En effet, chaque clique C est inclue dans une clique maximale et le potentiel ψ_C associé peut être combiné avec celui de cette clique maximale.

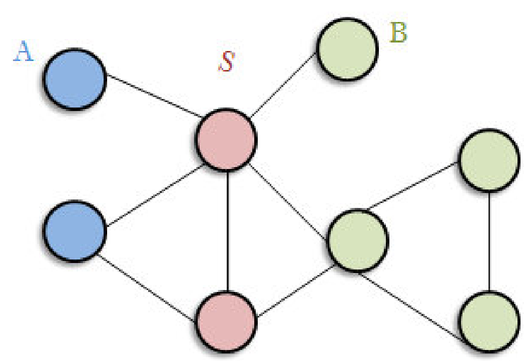
4.2 Séparation et Indépendance

Une relation entre la loi associée à G et la séparation/indépendence est donnée par:

Proposition 4.3 *Soit $G = (V, E)$ un graphe non orienté avec la loi $p(x) \in \mathcal{L}(G)$, alors pour tout $A, B, S \subset V$ tels que S sépare A et B, on a $X_A \perp\!\!\!\perp X_B \,|\, X_S$.*

Démonstration A et B étant séparés par S implique que A, B et S sont disjoints. V peut alors partitionné par $\tilde{A}$, $\tilde{B}$ et S, où $\tilde{A}$ contient tous les éléments de A et tous les éléments dans $V \backslash S$ qui sont connectés à A; $\tilde{B} = V \backslash S \cup \tilde{A}$. Alors $V = \tilde{A} \cup \tilde{B} \cup S$ et S sépare $\tilde{A}$ et $\tilde{B}$. Soit C une clique du graphe, alors il est impossible que $C \cap \tilde{A} \neq \emptyset$ et $C \cap \tilde{B} \neq \emptyset$, car sinon il aura un chemin connectant $\tilde{A}$ et $\tilde{B}$. Donc $C \cap \tilde{A} = \emptyset$ ou $C \cup \tilde{B} = \emptyset$, ce qui est équivalent à:

$$C \subset \tilde{A} \cup S \quad \textbf{ou} \quad C \subset \tilde{B} \cup S$$



S sépare A et B

Alors $p(x)$ peut décomposé comme:

$$p(x) = \prod_{C \subset \tilde{A} \cup S} \psi_C(x_C) \prod_{C \subset \tilde{B} \cup S} \psi_C(x_C) := g(x_{\tilde{A} \cup S}) \cdot h(x_{\tilde{B} \cup S}).$$

Or $A \subset \tilde{A}$ et $B \subset \tilde{B}$, on en déduit que

$$\begin{aligned} p(x_A | x_B, x_S) = \frac{p(x_A, x_B, x_S)}{p(x_B, x_S)} &= \frac{g(x_A, x_S) h(x_B, x_S)}{g(x_S) h(x_B, x_S)} \\ &= \frac{g(x_A, x_S) h(x_S)}{g(x_S) h(x_S)} = \frac{p(x_A, x_S)}{p(x_S)} = p(x_A | x_S). \end{aligned}$$

On peut ainsi conclure que $X_A \perp\!\!\!\perp X_B | X_S$.

■

La réciproque est admise:

Théorème 4.4 *(Hammersley-Cliford) Si* $\forall x, p(x) > 0$ *alors :*

$$[\forall A, B, S \subset V \textit{ t.q. } S \textit{ sépare } A \textit{ et } B \Longrightarrow X_A \perp\!\!\!\perp X_B \mid X_S] \Longrightarrow p(x) \in \mathcal{L}(G)$$

Proposition 4.5 *(Marginalisation) Soit* $G = (V, E)$ *un graphe non orienté avec la loi* $p(x) \in \mathcal{L}(G)$*, alors* $p(x_1, ..., x_{n-1})$ *se factorise dans le graphe* G' *égale à:*

- *au graphe* G *restreit aux noeuds* $X_1, ..., X_{n-1}$,
- *auquel on rajoute tous les connextions entre les voisins de* n.

Démonstration Il suffit de séparer les cliques qui contiennent n:

$$p(x) = \prod_{C \in \mathcal{C}} \psi_C(x_C) = \prod_{n \in C} \psi_C(x_C) \prod_{n \notin C} \psi_C(x_C),$$

$$p(x_1, ..., x_{n-1}) = \sum_{x_n} p(x) = g(x_{\text{voisins de n}}) \prod_{n \notin C} \psi_C(x_C).$$

■

4.3 Comparaisons

	Modèles graphiques orientés	Modèles graphiques non-orientés
Définition	$p(x) = \prod_{i=1}^{n} p(x_i \vert x_{\pi_i})$	$p(x) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C)$
Séparation	d-séparation	séparation
Marginalisation	non close	close

On va s'intérreser à la relation entre le graphe orienté et non-orienté.

Définition 4.6 *Soit $G = (V, E)$ est un* DAG*, le graphe symétrisé $\tilde{G} = (\tilde{V}, \tilde{E})$ est tels que $\tilde{V} = V$, $(u, v) \in \tilde{E}$ ssi. $(u, v) \in E$ ou $(v, u) \in E$.*

Sous cette définition, on a

Proprieté 4.7 *Soit $G = (V, E)$ est un* DAG *et il n'admet pas de v-structure, alors $\mathcal{L}(G) = \mathcal{L}(\tilde{G})$.*

Définition 4.8 *Soit $G = (V, E)$ est un* DAG*, le graphe moralisé $\bar{G}$ est le graphe symetrisé, dont on a connecté les parents entre eux.*

Sous ces définitions, on a

Proprieté 4.9 *Soit $G = (V, E)$ est un* DAG*, alors*

- *G n'a pas de v-structure $\Rightarrow \bar{G} = \tilde{G}$.*
- *$\mathcal{L}(G) \subset \mathcal{L}(\bar{G})$.*

4.4 Algorithme d'élimination

4.4.1 Marginalisation - un example

Définition 4.10 *Soient* $u : (X_1, ...X_n) \to R_+$ *et* $A \subset \{1, ..., n\}$, *la marginale sur* x_A *est définie par:*

$$u(x_A) = \sum_{x_{A^c}} u(x). \tag{4.1}$$

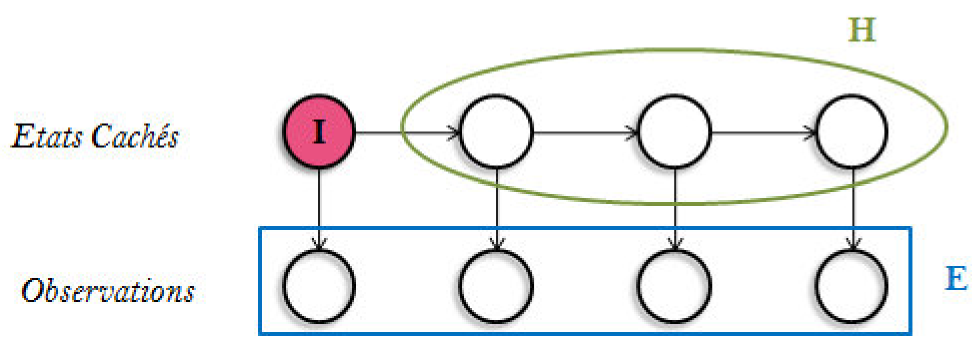


Figure 4.2. $V = I \cup H \cup E$ et l'on veut calculer $p(x_I|x_E)$.

On se ramène à calculer $u(x_A)$ efficacement. Par example, dans le graphe suivant, on calcule $u(x_1)$ selon le principe suivant: sommer de proche en proche n fois (avec peu de termes), au lieu de sommer en une fois (avec 2^n termes si les variables sont toutes binaires):

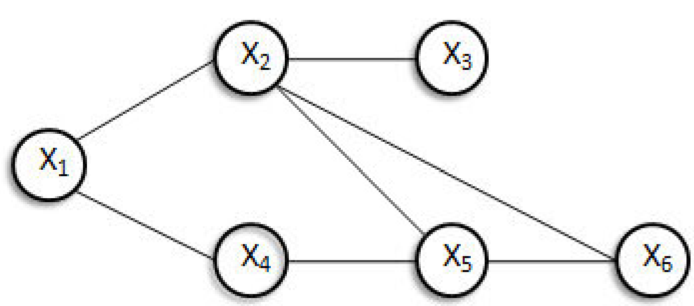


Figure 4.3. $u(x) = \psi_{12}(x_{12})\psi_{14}(x_{14})\psi_{23}(x_{23})\psi_{45}(x_{45})\psi_{256}(x_{256})$.

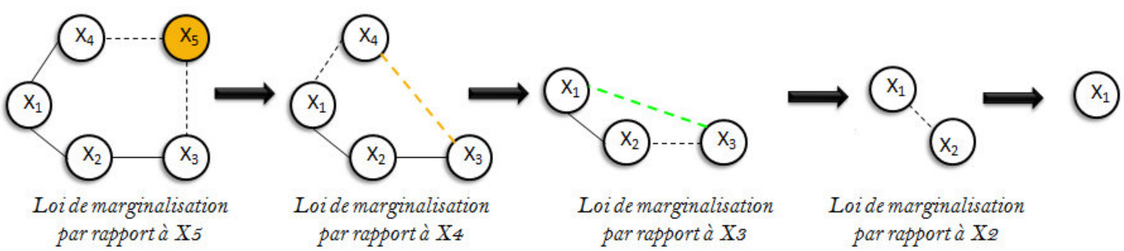
$$
\begin{aligned}
u(x_1) &= \sum_{x_2,x_3,x_4,x_5,x_6} u(x) \\
&= \sum_{x_2,x_3,x_4,x_5,x_6} \psi_{12}(x_{12})\psi_{14}(x_{14})\psi_{23}(x_{23})\psi_{45}(x_{45})\psi_{256}(x_{256}) \\
&= \sum_{x_2} \psi_{12}(x_{12}) \sum_{x_3} \psi_{23}(x_{23}) \sum_{x_4} \psi_{14}(x_{14}) \sum_{x_5} \psi_{45}(x_{45}) \sum_{x_6} \psi_{256}(x_{256}).
\end{aligned}
$$

4.4.2 Algorithme d'élimination

Soit $G = (V, E)$ un graphe non orienté. On considère l'ordre d'élimination $I = (n, n-1, \ldots, 1)$. A chaque étape, l'algorithme élimine le noeud qui suit dans l'ordre I, c'est-à-dire, supprime le noeud du graphe et connecte les voisins restants et on obtient une séquence de graphes $G_n = G, G_{n-1}, ..., G_0 = \emptyset$ tels que $\forall k = n, ..., 1$, $G_k = (V_k, E_k)$ tels que

$$V_k = \{1, \ldots, k\}, \quad E_{k-1} = (E_k \cap V_{k-1} \times V_{k-1}) \cup N_{G_k}(k) \times N_{G_k}(k),$$

où $N_{G_k}(k)$ désigne l'ensemble des voisins du sommet k dans G_k.



Sous ces notations,

Définition 4.11 *Le graphe reconstitué (aussi dit triangulé) est*

$$G^{\triangle} = (V, \cup_k E_k).$$

Définition 4.12 *On appelle clique d'élimination du noeud k l'ensemble des voisins d'un noeud après son élimination, le noeud lui-même y compris, i.e., l'ensemble $N_{G_k}(k) \cup \{k\}$. Il faut noter que*

- *C'est une clique dans $G^{\triangle}$.*
- *Si $u(x)$ se factorise dans G, alors $u(x_1, ..., x_k)$ se factorise dans G_k.*

4.4.3 Complexité finale de l'algorithme

Soit les $X_1, ..., X_n$ prennent Ω valeurs, alors

Proposition 4.13 *la complexité de passer de $u(x_1, ..., x_k)$ à $u(x_1, ..., x_{k-1})$ est*

$$O(\Omega^{|clique\ d'elimination|})$$

.

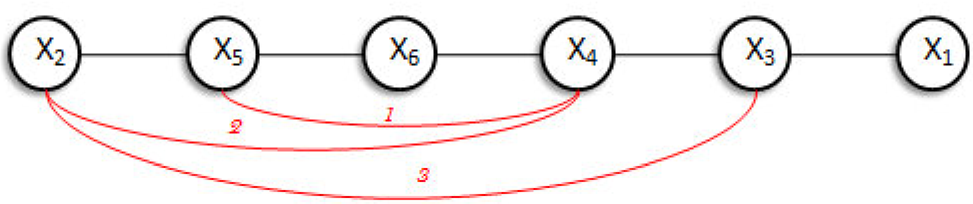
Par suite, la complexité de cet algorithme est au plus

$$O(n\Omega^{T(I)}),$$

où $T(I) = \max_k\{|$*clique d'elimination(k)*$|\}$ est la taille maximale des cliques d'éliminiation.

On s'intéresse à la meilleur ordre I qui minimise $T(I)$, et on introduit ainsi la notion la largeur arborescente (**tree-width**) qui est définie par:

$$TW = \min_I T(I) - 1.$$

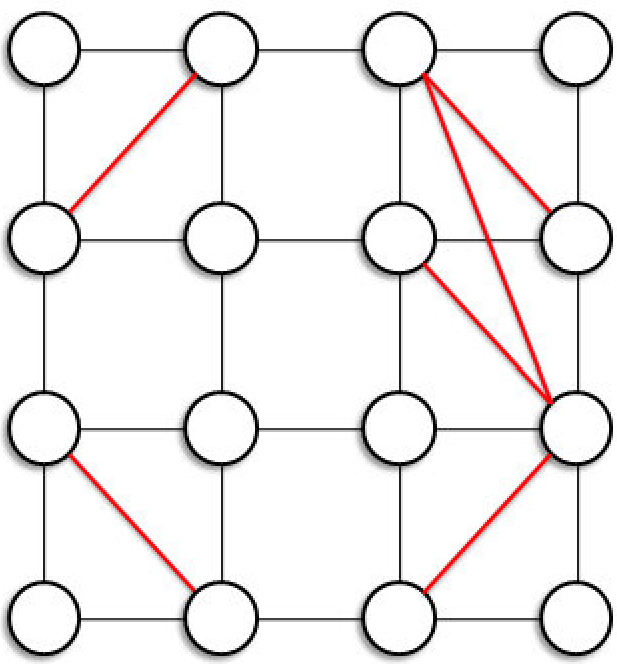


Eliminer les nœuds dans un « mauvais ordre » induira la création de cliques de plus en plus grandes → augmentation de la complexité

Quand on choisi la meilleur ordre, la complexité de cet algorithme est au plus $O(n\Omega^{TW+1})$.

Quelques remarques:

1. La complexité de l'algorithme d'élimination est linéaire en n, à comparer avec Ω^n.
2. Le calcul de TW est NP-difficile, mais l'algorithme glouton nous permet d'approcher un résultat optimum global.
3. Si G est un arbre (graphe <u>connexe</u> et sans cycle), on obtient $TW = 1$,
4. Si G est les grilles TW peu être grand ($\geq c\sqrt{n}$) même si G a peu d'arêtes.



Grille en 2D à n nœuds