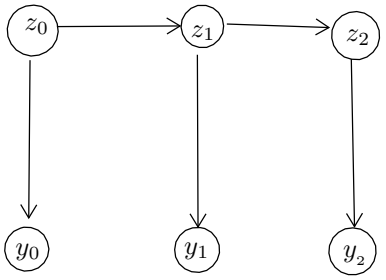
Apprentissage dans les modèles graphiques et Méthodes MCMC 2011/2012

Cours 8 — 17 novembre

Enseignant: Francis Bach *Scribe: Nicolas Pécheux, Rémi Lajugie*

8.1 HMM : Suite et Fin

Ex de HMM :



On rappelle que l'algorithme de propagation de messages pour les HMM s'écrit sous la forme de deux récursions.

L'une d'entre elle est la récursion alpha : $\alpha_{t+1} = p\left(y_{t+1} \mid z_{t+1}\right) \sum_{z_t} p\left(z_{t+1} \mid z_t\right) \alpha_t\left(z_t\right)$.

Computationnellement, comme les probabilités sont généralement petites, on effectue le calcul en logarithme. Par exemple pour la récursion alpha, on écrit matriciellement :

$\alpha_{t+1} = M\alpha_t \Leftrightarrow \exp\left(\tilde{\alpha}_{t+1}\right) = Mexp\left(\tilde{\alpha}_t\right) avec \tilde{\alpha}_t = \log\left(\alpha_t\right)$

Pour un calcul encore plus robuste, il faut écrire plus précisément :

si $y = \sum_1^n x_k$,

on pose $\tilde{y} = \log\left(y\right)$, $\tilde{x} = \log\left(x\right)$

alors on a que $y = \sum_1^n x_k \Leftrightarrow \exp\left(\tilde{y}\right) = \sum_1^n \exp\left(\widetilde{x_k}\right)$

d'où $\tilde{y} = \log\left(\sum_1^n \exp\left(\widetilde{x_k}\right)\right)$

Mais cette dernière expression n'est pas suffisante pour assurer un calcul numérique de bonne qualité. En effet si les $\widetilde{x_k}$ sont négatifs et que parmi eux il y en ait un qui ait le mauvais goût d'être trop proche de 0, la somme des exponentielles de ces nombres risque d'être considérée numériquement comme valant l'exponentielle du plus grand des $\widetilde{x_k}$. C'est pourquoi on doit effectuer une sorte de normalisation des $\widetilde{x_k}$.

On pose $M = \max\left(\widetilde{x_k}\right)_k$ ainsi on peut écrire $\tilde{y} = \log\left(\sum_1^n \exp\left(\widetilde{x_k} - M\right)\right) + M$. cette fois le calcul est robuste.

Remarque 8.1.1 *Pour ce qui concerne les HMM, l'algorithme max-produit va donner la séquence la plus probable pour les états cachés.*

8.2 Apprentissage dans les modèles graphiques

Exemple introductif : Dans un HMM, on suppose que
$p(z_0) \sim Multinomiale(\pi_o)$
$p(y_t | z_t) \sim f(y_t, z_t, \beta)$
$p(z_{t+1} | z_t) \sim A_{zt+1,z_t}$(On suppose que l'on a une chaine homogène.)
On estime les paramètres par EM puisque l'on n'observe pas toutes les variables. Dans notre cas :

E-step : $\mathbb{E}[p(z|y)] = \sum_k p(z_0 = k | , y) \log(\pi_c)_k + \sum_{t=0}^{t=T-1} \sum_k p(z_t = k | y) \log(\pi_c)_k + \sum_{t=0}^{t=T-1} \sum_k p(z_t = k | y) \log$

$\sum_{t=0}^{t=T-1} \sum_{k,k'} p(z_t = k, z_{t+1} = k' | y) \log(A_{k,k'})$

M-step : Pour ce qui concerne π_0 et A c'est classique (comme on l'a fait dans le DM1). Pour les probas d'émissions en revanche, la résolution dépend du problème : on est dans le cas « weighted ML ».

Théorème 8.1 *Soit les trois hypothèses suivantes :*
i) Soit G un DAG
ii) Soit $p_\theta \in \mathcal{L}(G)$tel que $p_\theta(x_i | x_{\pi_i})$ ne dépende que d'un seul paramètre θ_i
iii) Soit n observations complètes et IID des p variables. (On note X$\in \mathbb{R}^{nxp}$)
Alors le Maximum de Vraisemblance se découple.

Démonstration $Vraisemblance(X) = \prod_{j=1}^n p\left(x_k^j, k = 1 \dots p\right)$

Puisque G est un DAG, on a : $= \prod_{k=1}^p \prod_{j=1}^n p\left(x_k^j | x_{\pi_k}\right)$

Vu l'hypothèse ii) $= \prod_{k=1}^p \prod_{j=1}^n p_{\theta_k}\left(x_k^j | x_{\pi_k}\right)$

Ce qui veut bien dire que le maximum de vraisemblance est un problème découplé i.e que l'on peut se contenter de maximiser par rapport à chacun des paramètres θ_k. ■

Remarque 8.2.1 *: Dans un modèle graphique Non Orienté, l'apprentissage n'est pas un problème découplé mais il est convexe (cf chapitre du cours sur la méthode IPF)*

Remarque 8.2.2 *Si les données ne sont pas complètes, on doit employer l'EM pour résoudre le problème.*

Exemple : On considère le modèle dit naïve Bayes (qui n'est ni naïf ni Bayésien pour un sou)

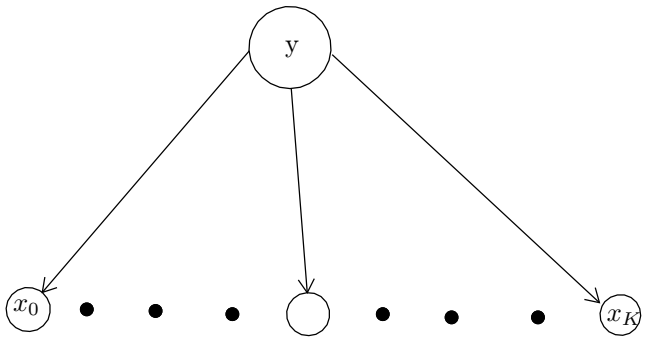
Ce modèle est utilisé par exemple pour classifier des documents en faisant l'hypothèse du sac de mots : il y a M classes de documents, et on veut affecter un document à une classe. Pour cela on regarde la présence de K mots qui vont nous permettre de déterminer à quelle classe de document on a affaire. On appelle x le un vecteur binaire de K bits donnant la présence ou l'absence d'un mot dans le document. Par exemple, pour savoir si un article parle de sport on regarde la présence ou l'absence de termes du type « football », « ballon », « piscine ».

Naïvement on pourrait tenter d'estimer directement $p(\textit{document} \in \textit{classe} \,|x)$. En pratique c'est impossible car on a 2^k possibilités pour le vecteur x, ce qui sera vite rhédibitoire d'un point de vue computationnel.

On utilise donc un autre modèle, dit Naïve Bayes. On fait là l'hypothèse que les descripteurs x sont indépendants les uns des autres conditionnellement à la classe. C'est une hypothèse clairement trop forte (car la présence de mots comme « gardien » et « défenseur » sont par exemple très corrélées) mais dans la pratique elle est relativement pertinente.

On remarque que $p(y\,|x)$ est proportionnel à $p(y\,|x)\,p(y)$.

Ici on suppose que $p(y) \sim \textit{multinomiale}$ et $p(y\,|x) \sim$ K variables independantes. On a donc le modèle graphique suivant



Dans ce modèle graphique, on peut appliquer le théorème précédent puisque les variables sont binaires et n'ont donc qu'un seul paramètre à apprendre. $\mu_{ik} = \mathbb{P}(x = 1\,|y = i)$

On peut améliorer encore ce modèle, en le rendant moins naïf, en ajoutant des arrêtes dans le graphe.

8.2.1 Lien avec la régression logistique

Avec les notations du modèle précédent, on peut écrire :

$p(y)\,p(x\,|y) = p(y)\prod_{k=1}^{K} p(x_k\,|y)$=p(y)$\prod_{k=1}^{K}\prod_{i=1}^{M}\mu_{ik}^{\delta(x_k=1,y_k=i)}\,(1-\mu_{ik})^{\delta(x_k=0,y_k=i)}$

Puisque $p(y)$ est proportionnel à $\prod_{i=1}^{M}\pi_i^{\delta(y=i)}$ (où π est le paramètre de la multinômiale)

on a $\log(p(y\,|x)) = C_1\delta(y=i) + C_2\delta(x_k=1,y_k=i) + C_3\delta(x_k=0,y_k=i)$

ce qui peut se réécrire

$$log\,(p\,(y\,|x)) = \Psi\,(x,y)^T\,\eta$$

avec $\Psi(x,y) = \begin{pmatrix} \delta(y=i) \\ \delta(x_k=1,y_k=i) \\ \delta(x_k=0,y_k=i) \end{pmatrix}$

si M=2,

on a que $log\,(p\,(y=1\,|x)) = \eta^T\Psi\,(x,y)$

ce qui est bien l'expression usuelle de la régression logistique

Le cas M>2 est détaillé dans le chapitre du poly consacré à la Régression SOFTMAX

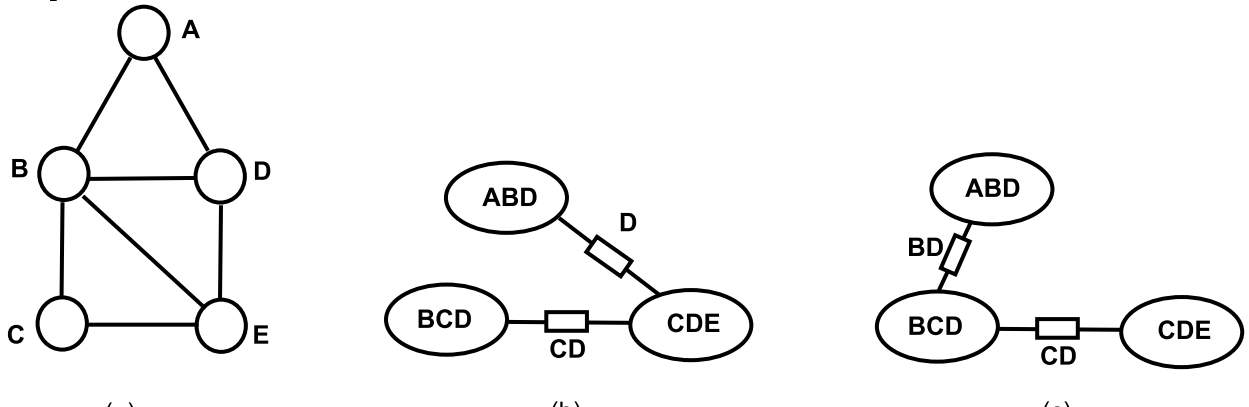
Inférence exacte : (chapitre 16 du poly)

$p \in \mathfrak{L}(G)$. Le but est de calculer les probabilités sur les noeuds et arrêtes du graphe. Dans le cas des arbres on utilisera l'algorithme somme-produit dont la complexité est, si une variable x_i peut prendre r valeurs, en $O(r^2 n)$.

Néanmoins, le problème de l'inférence exacte reste entier si l'on ne considère pas un arbre. La méthode que nous allons proposer est générale et ne nécessite pas d'avoir de structure particulière a priori sur le graphe G.

L'intérêt de cette méthode est d'abord et surtout historique car en pratique, comme on va le voir sur un exemple simple, c'est un problème compliqué (et même NP-difficile).

La méthode d'inférence exacte consiste à transformer un graphe qui n'est pas un arbre en un arbre, en regroupant ses cliques. On travaille sur ce que l'on appelle « l'arbre de cliques ». Il s'agit de l'arbre obtenu en réunissant en un seul noeud une clique entière comme dans l'exemple ci-dessous :

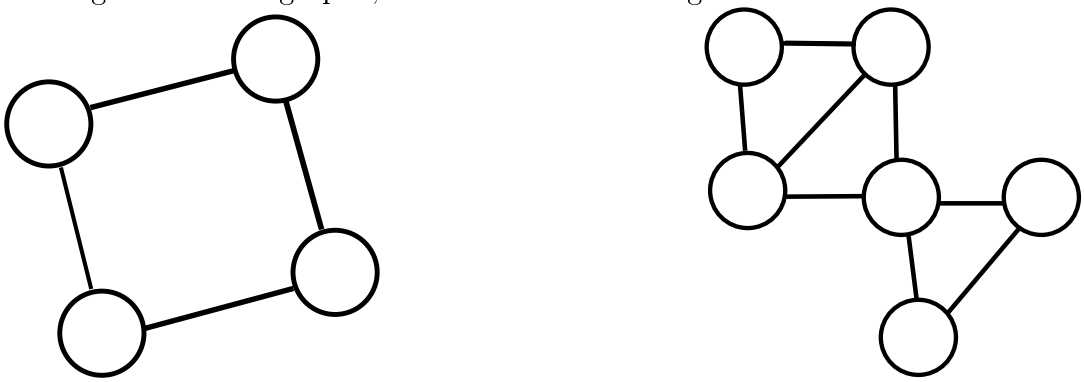


(a) (b) (c)

En regroupant plusieurs noeuds, on fait croitre exponentiellement le nombre de valeurs possibles pour le noeud nouvellement créé.

Ensuite pour faire l'inférence exacte, il suffit d'appliquer l'algorithme somme-produit.

Néanmoins, avant cette opération de regroupement des cliques, il est parfois nécessaire de « triangulariser » le graphe, comme le montre la figure ci dessous



On part d'un graphe avec des cycles ressemblant à celui de la figure de gauche pour finalement arriver à un graphe ayant des cycles « triangulaires » comme dans la figure de droite. Dans un graphe triangularisé on appelle TW ou largeur arborescente, la taille maximale d'une clique après l'opération de triangularisation. On montre alors que l'inférence est en $O\left(n2^{TW}\right)$

En pratique il est très difficile d'utiliser cette inférence exacte, car la largeur arborescente peut dépendre de n, le nombre de sommets. Par exemple pour la simple grille 2D, fort utilisée en traitement des images, la largeur arborescente vérifie : $TW \approx \sqrt{n}$. L'inférence exacte est donc un problème très complexe.

8.3 Inférence approchée

L'inférence exacte étant sans d'espoir pour de nombreux modèles probabilistes, on s'intéresse donc à des méthodes d'inférence approchée.

8.3.1 Méthode d'échantillonnage (*Sampling*)

Dans de nombreux cas on est intéressé par l'espérance d'une certaine fonction f sous une certaine loi de probabilité d'intérêt p que l'on ne peut pas calculer directement.

On considère une variable aléatoire X de loi p et on cherche a calculer $\mu = \mathbb{E}[f(X)]$.

Exemple 8.3.1 $X = (X_1, ..., X_n)$,

$$f(X) = \delta(X = x_A)$$

$$\mathbb{E}[f(X)] = \mathbb{P}(X = x_A)$$

Si on sait échantilloner selon la loi p on peut utiliser la méthode suivant :

1. Échantillonner $X_1, ..., X_n$ i.i.d. de loi p

2. $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} f(X_i)$

Les deux propostitions suivantes justifient la validité de cette méthode.

Proposition 8.2 (Loi des grands nombres)

$$si\ \mathbb{E}[f(X)] < \infty\ alors\ \hat{\mu} \longrightarrow \mu\ p.s.$$

Proposition 8.3 (TCL) *Si* $\sigma^2 = \mathbb{V}ar(f(X))$

$$\hat{\mu} \longrightarrow N(\mu, \frac{\sigma^2}{n})$$

et donc $\mathbb{E}[||\hat{\mu} - \mu||_2^2] = \sigma^2/n$

Comment échantillonner une loi ?

1. Loi uniforme sur $[0,\ 1] \rightarrow$ `rand`
2. Loi de Bernoulli de paramètre $p \rightarrow X = \mathbf{1}_{\{U<p\}}$ avec $U \sim \mathcal{U}([0,1])$
3. Inversion de la fonction de répartition :

$$\forall x \in \mathbb{R} \qquad F(x) = \int_{-\infty}^{x} p(t)dt = \mathbb{P}(X \in [-\infty, x])$$

$$X = F^{-1}(U) \text{ avec } U \sim \mathcal{U}([0,1])$$

Démonstration $\mathbb{P}(X \leq y) = \mathbb{P}(F^{-1}(U) \leq y) = \mathbb{P}(U \leq F(y)) = F(y)$ ■

Exemple 8.3.2 *Loi exponentielle (une des rares pour laquelle on sait calculer analytiquement l'inverse)*

$$p(x) = \lambda e^{-\lambda x}\mathbf{1}_{\mathbb{R}_+}(x)$$

$$X = -\frac{1}{\lambda}\ln(U)$$

8.3.2 Méthode de rejet (*Rejection sampling*)

On suppose que l'on ne connait la loi p qu'à une constante multiplicative près (impossible à calculer), ce qui est souvent le cas en pratique (modèles graphiques non orientés, conditionnement, etc.) :

$$p(x) = \frac{\tilde{p}(x)}{Z_p}$$

On suppose que l'on dispose d'une loi q simple (i.e. on sait échantillonner selon q) telle que

$$\exists k, \forall x, \tilde{p}(x) \leq kq(x)$$

On utilse alors l'algorithme suivant pour échantillonner selon p :

- On tire $X \sim q$
- On accepte X avec une probabilité $\frac{\tilde{p}(X)}{kq(X)} \in [0,1]$, sinon on recommence.

Démonstration $\mathbb{P}(X = j | X \text{ accepté}) \propto \mathbb{P}(X = j, X \text{ accepté}) \propto q(j)\frac{\tilde{p}(j)}{kq(j)} \propto \tilde{p}(j)$ ■

Quelle est la probabilité d'accepter ?

$$\begin{aligned}
\mathbb{P}(X \text{ accepté}) &= \sum_j \mathbb{P}(X \text{ accepté}, X = j) \\
&= \sum_j \mathbb{P}(X = j)\mathbb{P}(X \text{ accepté} | X = j) \\
&= \sum_j q(j)\frac{\tilde{p}(j)}{kq(j)} \\
&= \frac{Z_p}{k}
\end{aligned}$$

En pratique il est souvent très difficile de calculer k. De plus en grande dimension, k peut être très très grand et on risque de ne jamais accepter.

Remarque 8.3.1 *Si $\tilde{p}$ est à queue lourde (par exemple $\sim_\infty 1/x^2$), ce qui arrive souvent en économie, on a aucune chance de pouvoir choisir une loi de proposition q gaussienne.*

8.3.3 Échantillonnage préférentiel (*Importance sampling*)

On revient au problème de calculer l'espérance d'une fonction :

$$\begin{aligned}
\mu &= \int p(x)f(x)dx \\
&= \int \frac{p(x)}{q(x)} f(x)q(x)dx \\
&\approx \frac{1}{n}\sum_{i=1}^{n} g(X_i) \text{ où } X_i \sim q \\
&= \hat{\mu}
\end{aligned}$$

Où l'on a introduit une nouvelle distribution q. On remplacé le problème du calcul de $\mathbb{E}_p[f]$ par celui de $\mathbb{E}_q[g]$, avec une loi que l'on sait échantillonner. Un problème qui peut se poser est que la variance de g peut être très grande.

$$\mathbb{E}[\hat{\mu}] = \mathbb{E}[\mu]$$

si $\forall x \ |f(x)| \leq M$

$$\begin{aligned}
\mathbb{V}ar[\hat{\mu}] &\leq \frac{1}{n}\mathbb{V}ar[\frac{p(X)f(X)}{q(X)}] \\
&\leq \frac{1}{n}\mathbb{E}[\frac{p(X)f(X)}{q(X)}]^2 \\
&\leq \frac{1}{n}\sum \frac{p(x)^2}{q(x)}M^2 dx \\
&\leq \frac{M^2}{n}\sum \frac{p(x)^2}{q(x)}dx
\end{aligned}$$

La méthode fonctionne si on ne connaît p et q qu'à une constante multiplicative près :

$$p(x) = \frac{\tilde{p}(x)}{Z_p} \qquad q(x) = \frac{\tilde{q}(x)}{Z_q}$$

$$\begin{aligned}
\hat{\tilde{\mu}} &= \frac{1}{n}\sum_{i=1}^{n} f(X_i)\frac{\tilde{p}(X_i)}{\tilde{q}(X_i)} \\
&= \frac{Z_q}{Z_p}\left(\frac{1}{n}\sum_{i=1}^{n} f(X_i)\frac{\tilde{p}(X_i)}{\tilde{q}(X_i)}\right) \\
&\to \frac{Z_q}{Z_p}\mu
\end{aligned}$$

(Astuce géniale) on prend $f = 1$:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{\tilde{p}(X_i)}{\tilde{q}(X_i)} \to \frac{Z_q}{Z_p}$$

On a alors

$$\hat{\mu} = \frac{\frac{1}{n}\sum_{i=1}^{n} f(X_i)\frac{\tilde{p}(X_i)}{\tilde{q}(X_i)}}{\frac{1}{n}\sum_{i=1}^{n}\frac{\tilde{p}(X_i)}{\tilde{q}(X_i)}}$$

8.4 MCMC (*Monte Carlo Markov Chains*)

"C'est pas dit que c'est bien, mais c'est beaucoup utilisé"

$x \in \mathcal{X}$ discret.

On cherche à construire une chaîne de Markov $X_1, ...X_p$ telle que "$p(x_p) \to p(x)$" où p est une certaine distribution cible.

8.4.1 Rappels sur les chaînes de Markov

Définition 8.4 (Chaîne de Markov homogène) *On dit que $X = X_1, ..., X_n$ est une chaîne de Markov homogène si $\forall n, \forall x, y$*

$$\mathbb{P}(X_{n+1} = y | X_n = x, X_{n-1}, ..., X_0) = \mathbb{P}(X_{n+1} = y | X_n = x) = S(x, y)$$

S est appelée la matrice de transition de la chaîne et vérifie les propriété suivantes (matrice stochastique) :

- $S \in \mathbb{R}^{k \times k}$ *si* $|\mathcal{X}| = k$
- $S \geq 0$
- $S\mathbf{1} = \mathbf{1}$ *où* $\mathbf{1} = (1, ..., 1)^T$

Définition 8.5 *La loi π sur $\mathcal{X}$ est dite stationnaire si $S^T \pi = \pi$*

$$i.e.\ \forall x, y\ \ \pi(y) = \sum_x \pi(x) S(x, y)$$

Et donc si $X_n \sim \pi$ alors $X_{n+1} \sim \pi$

En effet si $\mathbb{P}(X_{n+1} = y) = \sum_x \mathbb{P}(X_{n+1} = y | X_n = x) \mathbb{P}(X_n = x) = \sum_x S(x, y) \pi(x) = \pi(y)$

Théorème 8.6 (de Perron-Frobenius) *Si S est une matrice stochastique, alors il existe au moins une loi stationnaire.*

Proposition 8.7 *Si $\forall x, y \in \mathcal{X},\ S(x, y) > 0$ alors la chaîne est irréductible.*
Si on note q_n la loi de X_n, alors pour toute loi q_0 on a :

$$q_n \to \pi$$

.

La vitesse de convergence n'est en générale pas connue à l'avance et peut même être exponentiellement lente (ceci dépend de la deuxième valeur propre de S).

$\longrightarrow$ Retenir que d'une manière générale une méthode MCMC est lente.

8.4.2 Matrices de transition

Soit p une loi sur $\mathcal{X}$, comment trouver une matrice de transition telle que $\forall x, y \in \mathcal{X}$

- $S(x, y) > 0$
- $\sum_y p(x) S(x, y) = p(y)$

Proposition 8.8 (Bilan détaillé) *Si $\forall x, y \in \mathcal{X}\ \ p(x)S(x, y) = p(y)S(y, x)$ alors p est stationnaire pour S. La chaîne est alors dite réversible.*

Démonstration $\sum_x S(x, y) p(x) = \sum_x p(y) S(y, x) = p(y) \sum_x S(y, x) = p(y)$ ■

8.4.3 Transition de Metropolis Hasting

La matrice de transition de Metropolis Hasting $S(x, y)$ est donnée par l'algorithme suivant :

1. $z \sim T(x, .)$ où T est une loi de proposition
2. Avec probabilité $\alpha = \min\left(1, \frac{p(z)T(z,x)}{p(x)T(x,z)}\right)$ on accepte $y = z$ sinon on rejette $y = x$

8.4.4 Échantillonnage de Gibbs (*Gibbs Sampling*)

L'échantillonnage de Gibbs revient à faire un choix particulier de la loi de proposition T. Si $x = (x_1, ..., x_n)$, pour $T(x, y)$

1. On prend $k \in \{1, ..., n\}$ au hasard selon une loi uniforme
2. On remplace x_k par un échantillon de loi $p(x_k|x_{reste})$

On comprend l'intérêt dans le cadre des modèles graphiques, grâce à la notion de couverture de Markov. En effet $p(x_k|x_{-k}) = p(x_k|x_{couverture(k)})$. On rappelle que dans le cas des modèles graphiques non orientés, la couverture de Markov est simplement l'ensemble des voisins directs. Pour les modèles graphiques orientés en revanche, ce sont les parents, les enfants et les parents des enfants.

8.4.5 Algorithme MCMC

1. Échantillonner X_0
2. $\forall n$ échantillonner X_{n+1} selon $S(., X_n)$

La loi de X_n converge alors vers p.

En pratique, pour générer $X \sim p$, on simule la chaine de Markov précédent et on prend la variable X_n, avec n suffisement grand, ce que l'on appelle *temps de chauffe* (*burn-in*). Comme on a en général besoin d'un k-échantillon $X^{(1)}, ..., X^{(k)}$, on est ammené à simuler k chaînes. On peut aussi simuler une seul très longue chaîne et attendre un certain temps entre deux prélèvements pour s'assurer (approximativement) de l'indépendance des variables aléatoires.

Avantage : On a bien convergence vers la loi d'intérêt
Inconvénient : Comme signalé, la vitesse peut être assez lente

Comme très souvent, on est face à un compromis entre précision et vitesse.