Théorie de l'information : Application aux choix de modèles.

Aurélien Garivier, CMLA ENS Cachan & Université Paris Sud Orsay

Séminaire de statistique - Université Paris-Dauphine.

Ma thèse

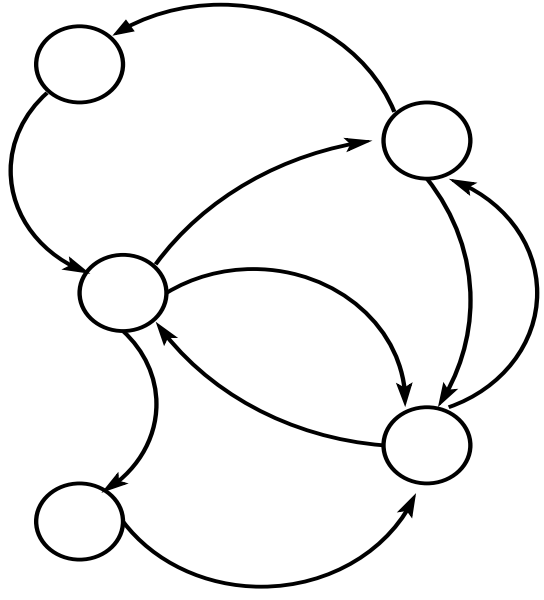
- 2004: Redondance de l'algorithme CTW sur les processus de renouvellement.
- 2005: Consistance de l'estimateur BIC illimité pour l'identification de VLMC.
- 2006: Une nouvelle borne inférieure pour la redondance maximin de motifs.
- 2006: Estimation d'ordre pour les HMM à émission infinie (avec A. Chambaz et E. Gassiat).
- 2006: Codage universel sur les alphabets infinis et classes enveloppes (avec S. Boucheron et E. Gassiat).

Plan de l'exposé

- Présentation des problèmes et modèles
- Théorie de l'information et principe MDL
- Estimateur d'arbre de contexte
- Estimateurs d'ordre de HMM

Chaînes de Markov cachées

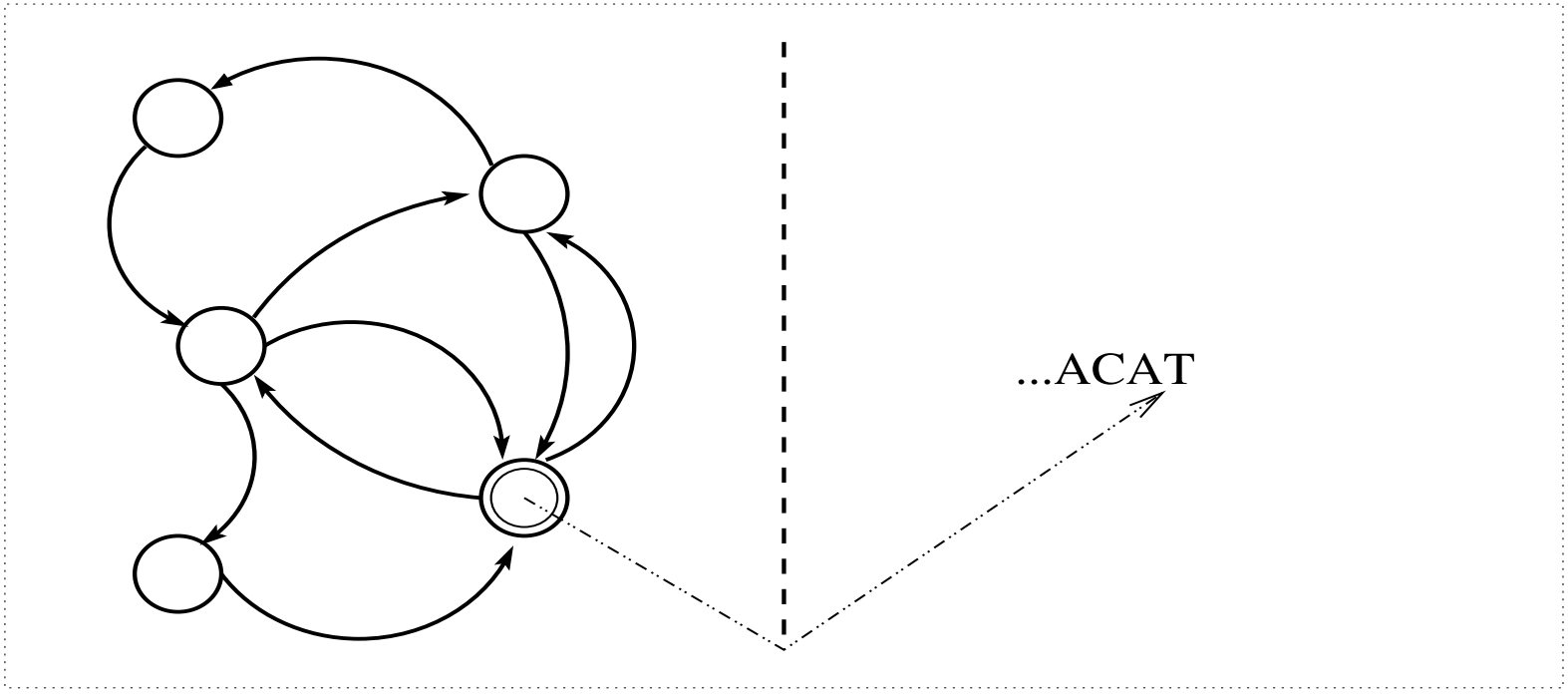
- chaque état caché a une loi d'émission differente.
- on passe d'un état caché à l'autre par un processus markovien.
- à chaque étape, on émet un caractère.



$\Longrightarrow$ on veut estimer l'ordre = le nb d'états cachés

Chaînes de Markov cachées

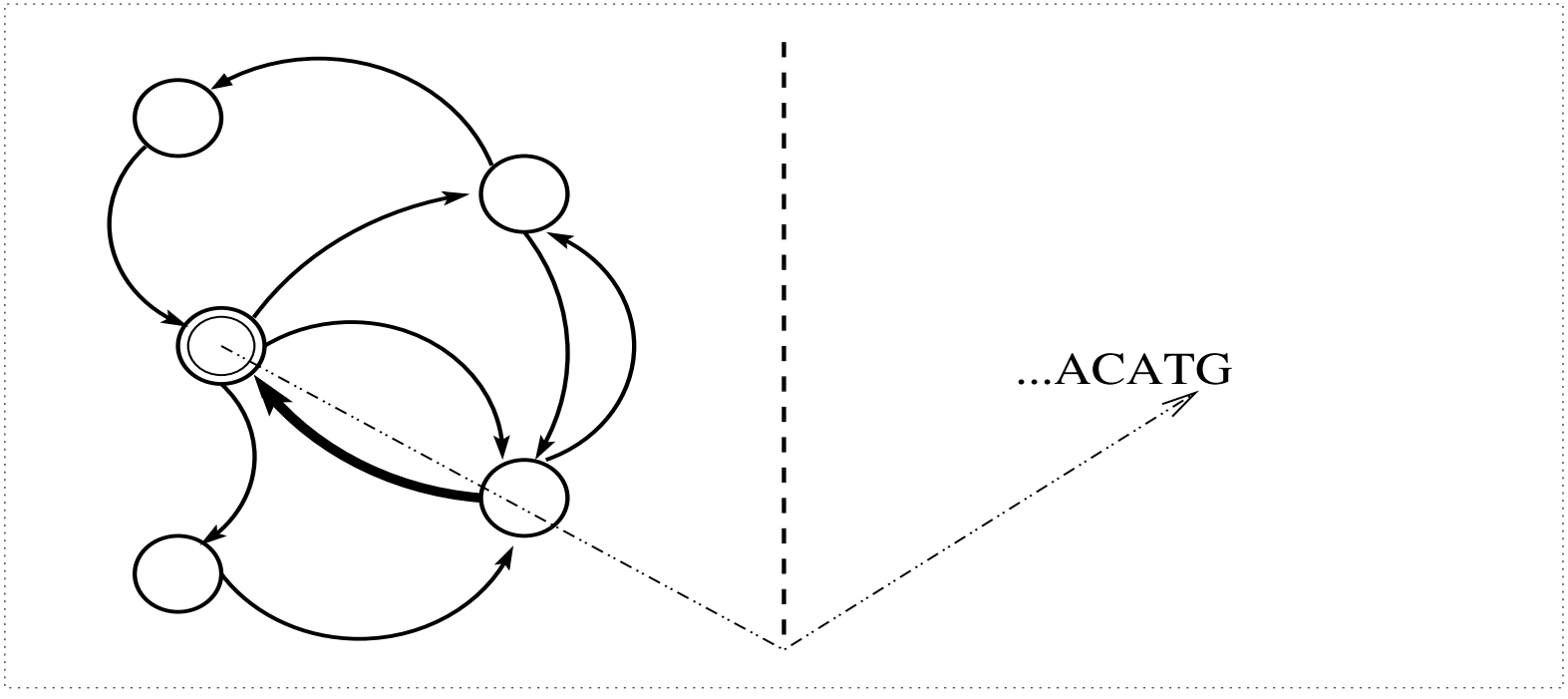
- chaque état caché a une loi d'émission differente.
- on passe d'un état caché à l'autre par un processus markovien.
- à chaque étape, on émet un caractère.



$\Longrightarrow$ on veut estimer l'ordre = le nb d'états cachés

Chaînes de Markov cachées

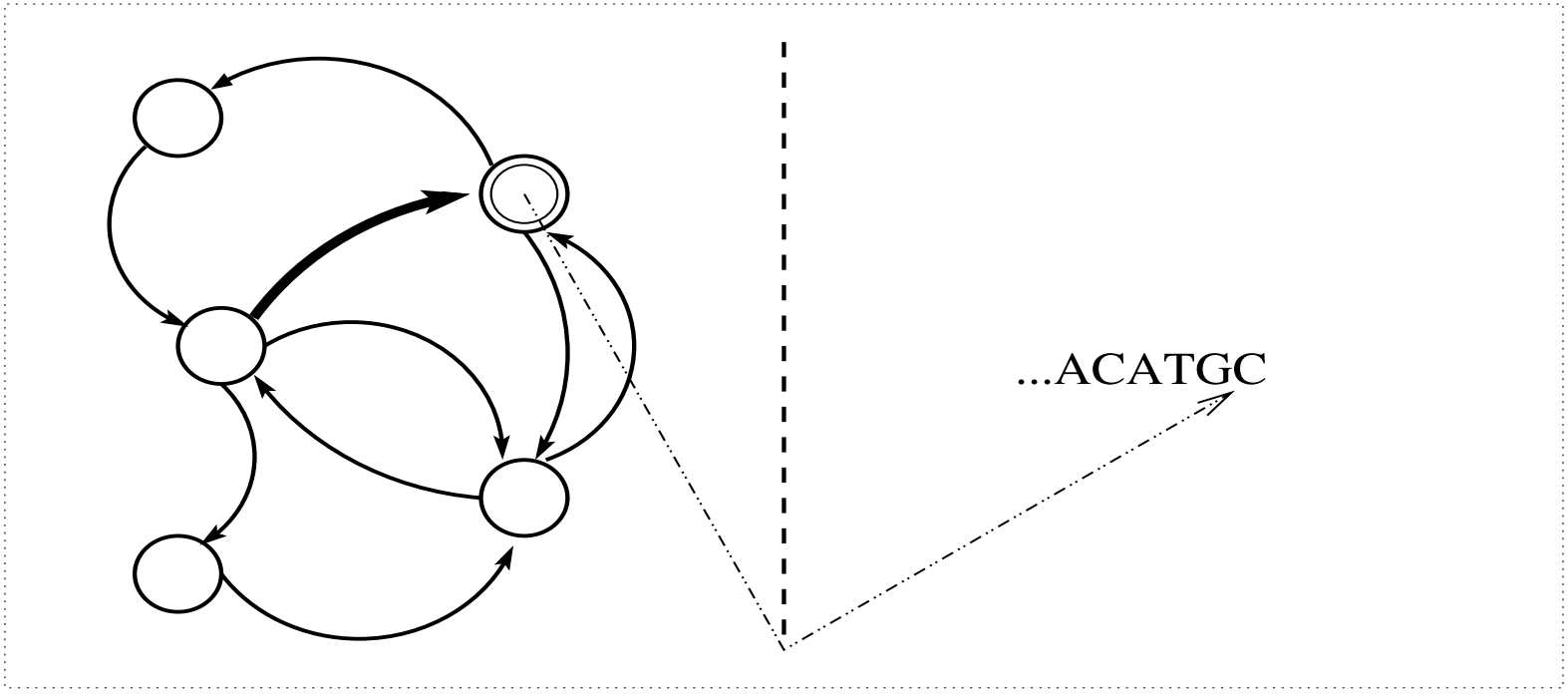
- chaque état caché a une loi d'émission differente.
- on passe d'un état caché à l'autre par un processus markovien.
- à chaque étape, on émet un caractère.



$\Longrightarrow$ on veut estimer l'ordre = le nb d'états cachés

Chaînes de Markov cachées

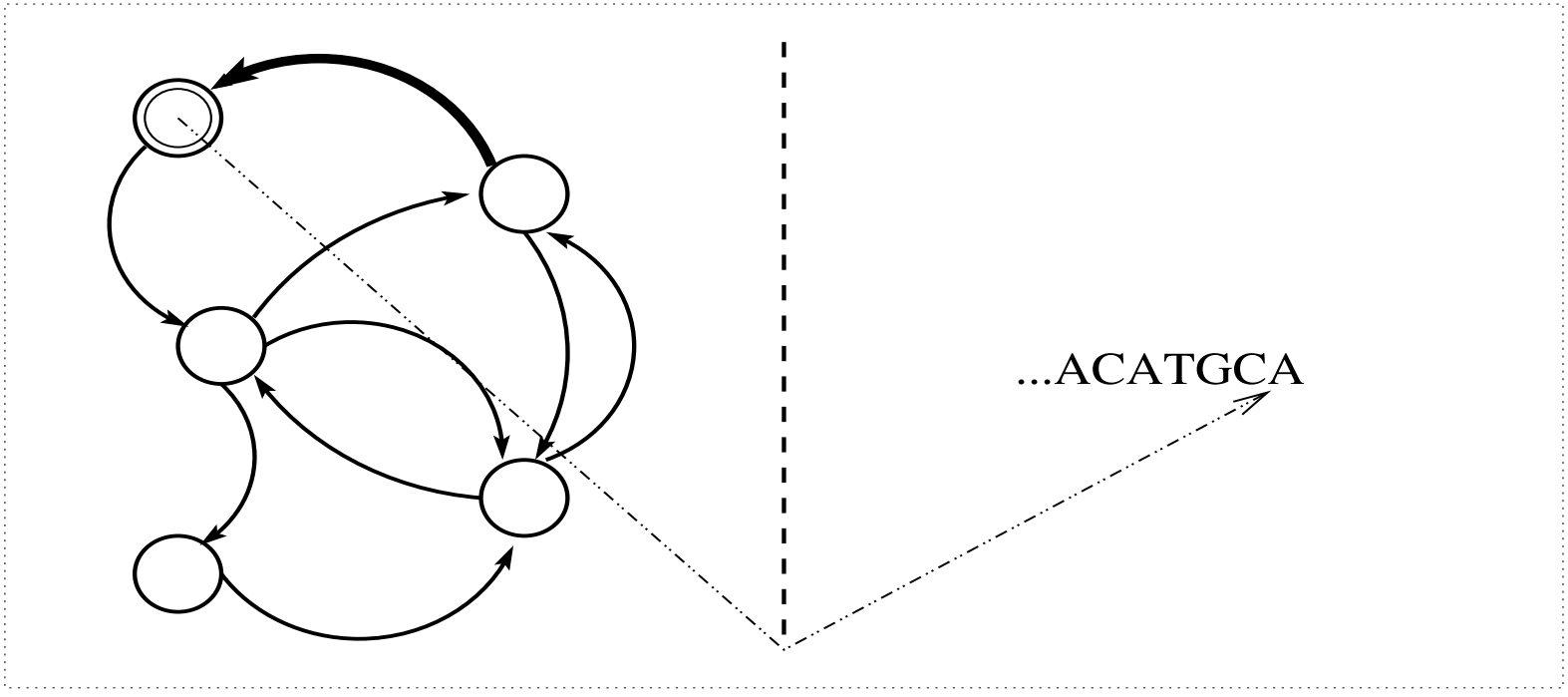
- chaque état caché a une loi d'émission differente.
- on passe d'un état caché à l'autre par un processus markovien.
- à chaque étape, on émet un caractère.



$\Longrightarrow$ on veut estimer l'ordre = le nb d'états cachés

Chaînes de Markov cachées

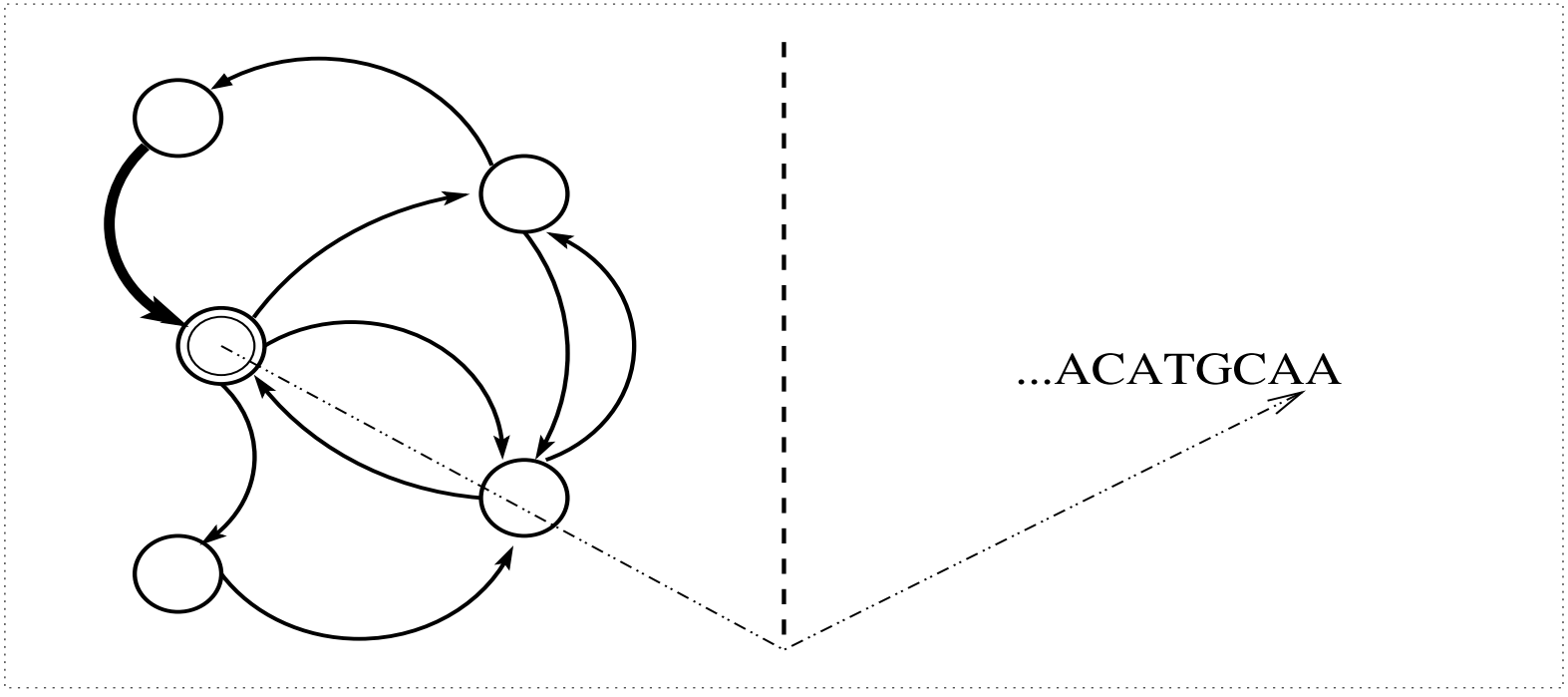
- chaque état caché a une loi d'émission differente.
- on passe d'un état caché à l'autre par un processus markovien.
- à chaque étape, on émet un caractère.



$\Longrightarrow$ on veut estimer l'ordre = le nb d'états cachés

Chaînes de Markov cachées

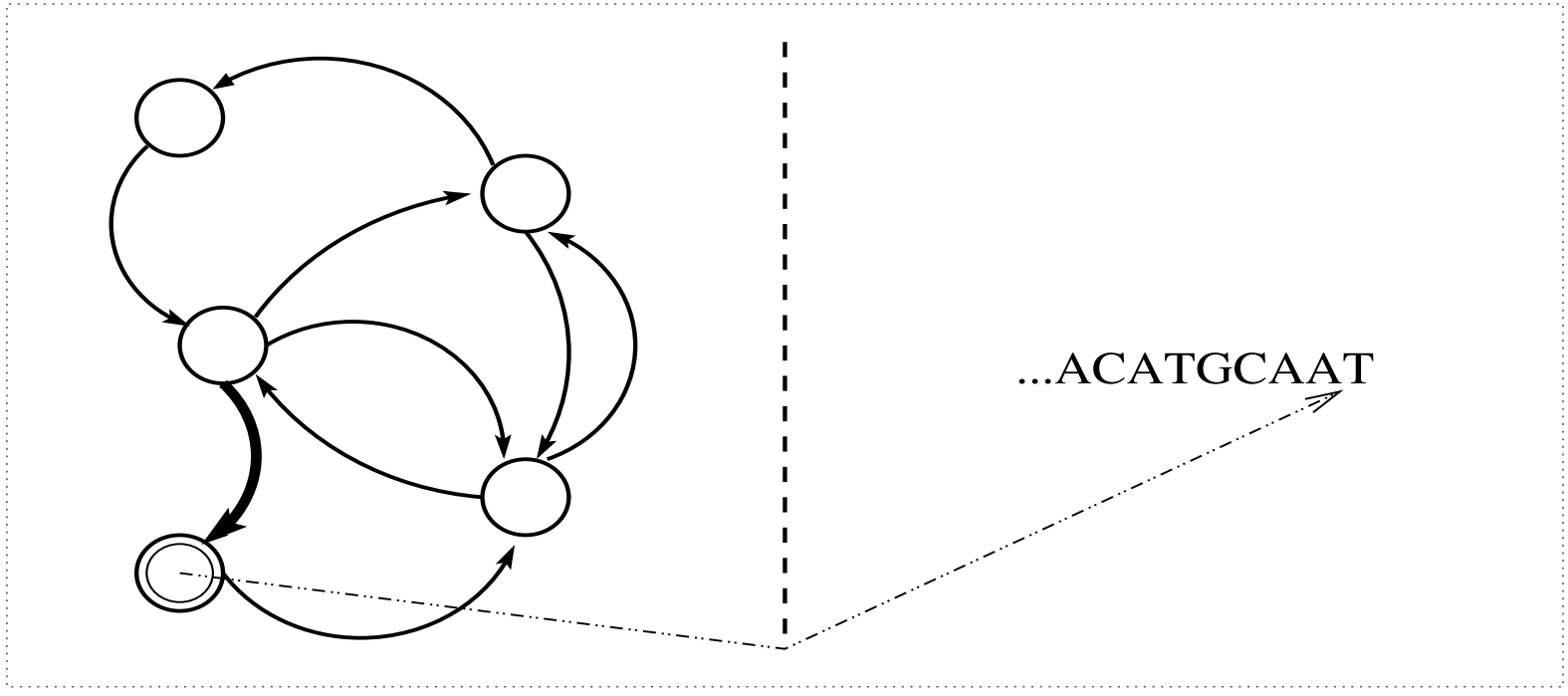
- chaque état caché a une loi d'émission differente.
- on passe d'un état caché à l'autre par un processus markovien.
- à chaque étape, on émet un caractère.



$\Longrightarrow$ on veut estimer l'ordre = le nb d'états cachés

Chaînes de Markov cachées

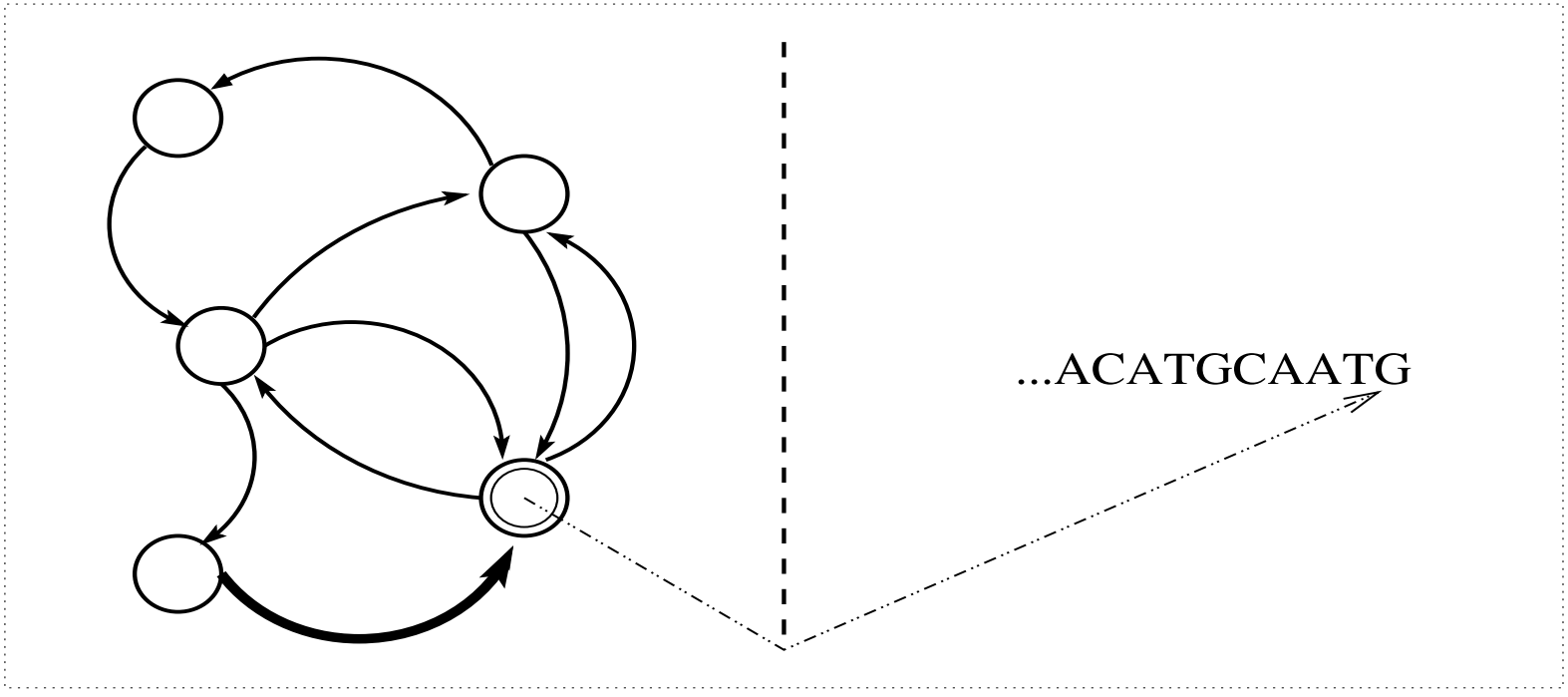
- chaque état caché a une loi d'émission differente.
- on passe d'un état caché à l'autre par un processus markovien.
- à chaque étape, on émet un caractère.



$\Longrightarrow$ on veut estimer l'ordre = le nb d'états cachés

Chaînes de Markov cachées

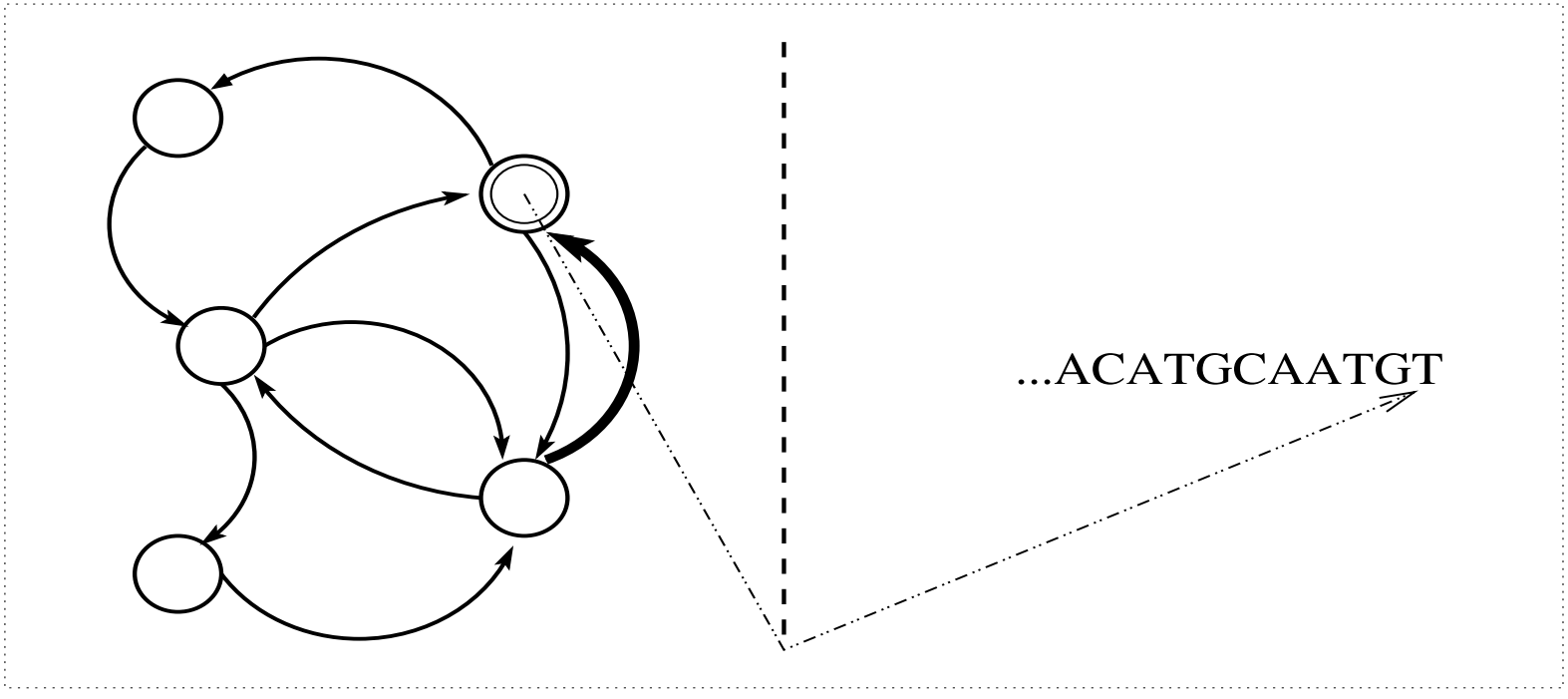
- chaque état caché a une loi d'émission differente.
- on passe d'un état caché à l'autre par un processus markovien.
- à chaque étape, on émet un caractère.



$\Longrightarrow$ on veut estimer l'ordre = le nb d'états cachés

Chaînes de Markov cachées

- chaque état caché a une loi d'émission differente.
- on passe d'un état caché à l'autre par un processus markovien.
- à chaque étape, on émet un caractère.

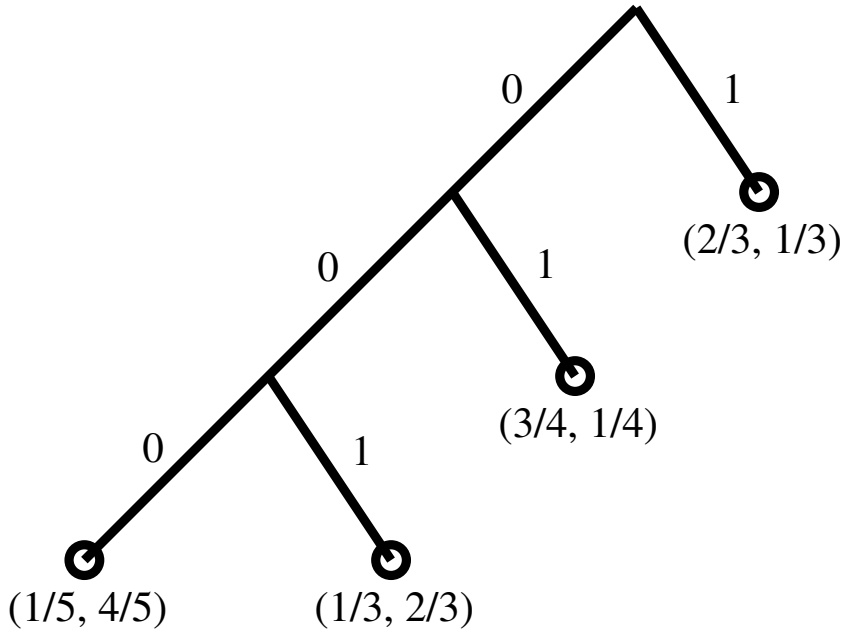


$\Longrightarrow$ on veut estimer l'ordre = le nb d'états cachés

Sources à arbre de contexte

Définition informelle : Une source à arbre de contexte (ou VLMC : Variable Length Markov Chain) est une chaîne de Markov dont l'ordre peut dépendre du passé.

$$T = \{1, 10, 100, 000\}$$



$$\begin{aligned} & P(X_1^4 = 00110 | X_{-1}^0 = 10) \\ = \quad & P(X_1 = 0 | X_{-1}^0 = 10) \\ \times \quad & P(X_2 = 0 | X_{-1}^1 = 100) \\ \times \quad & P(X_3 = 1 | X_{-1}^2 = 1000) \\ \times \quad & P(X_4 = 1 | X_{-1}^3 = 10001) \\ \times \quad & P(X_5 = 0 | X_{-1}^4 = 100011) \end{aligned}$$

Une source à arbre de contexte est paramétrée par :

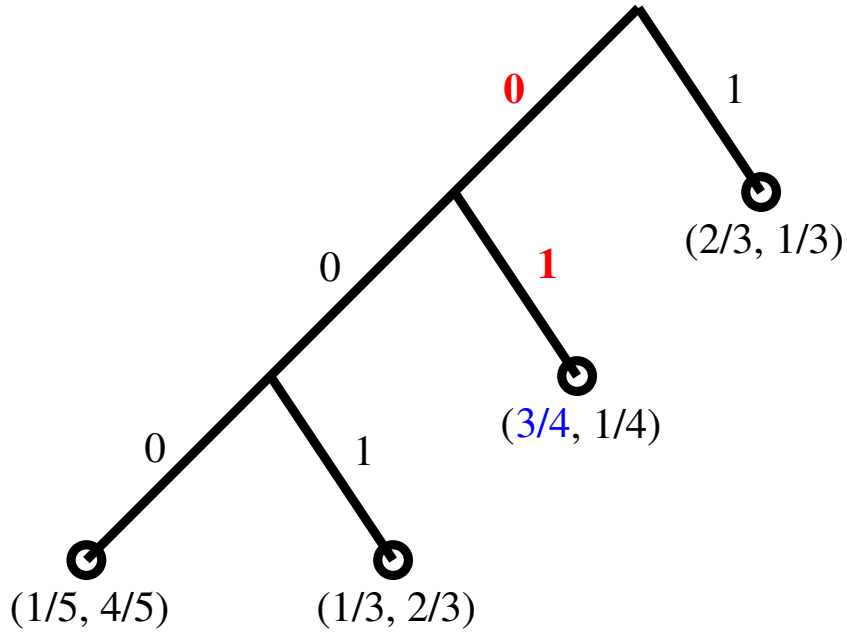
$$\Theta_T = \left\{ \left(\theta_1^s, \ldots, \theta_{|A|}^s \right) : s \in T, \sum_{i=1}^{|A|} \theta_i^s = 1 \right\}$$

Sources à arbre de contexte

Définition informelle : Une source à arbre de contexte (ou VLMC : Variable Length Markov Chain) est une chaîne de Markov dont l'ordre peut dépendre du passé.

$$T = \{1, \mathbf{10}, 100, 000\}$$

$$\begin{aligned} & P(X_1^4 = 00110 | X_{-1}^0 = 10) \\ = \quad & P(X_1 = 0 | X_{-1}^0 = \mathbf{10}) \qquad 3/4 \\ \times \quad & P(X_2 = 0 | X_{-1}^1 = 100) \\ \times \quad & P(X_3 = 1 | X_{-1}^2 = 1000) \\ \times \quad & P(X_4 = 1 | X_{-1}^3 = 10001) \\ \times \quad & P(X_5 = 0 | X_{-1}^4 = 100011) \end{aligned}$$



Une source à arbre de contexte est paramétrée par :

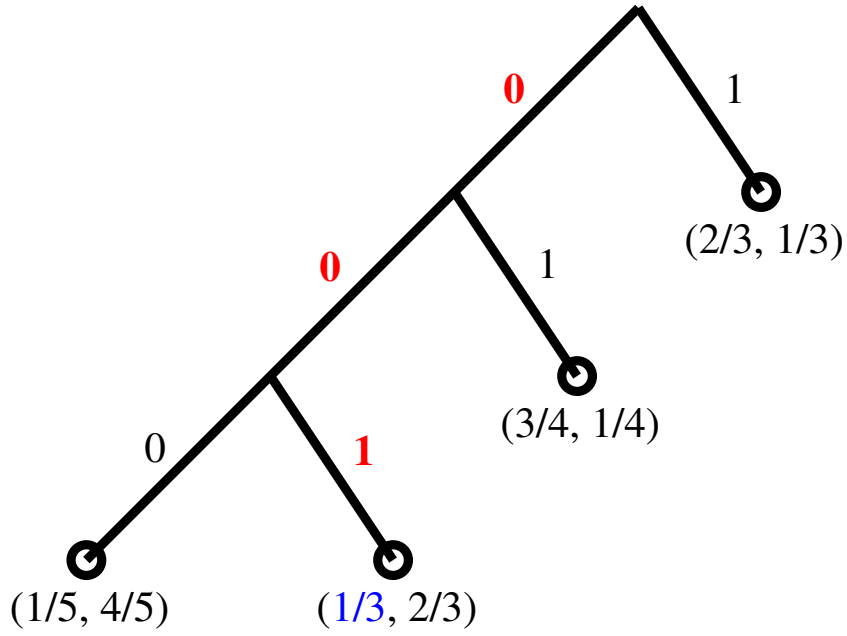
$$\Theta_T = \left\{ \left(\theta_1^s, \ldots, \theta_{|A|}^s \right) : s \in T, \sum_{i=1}^{|A|} \theta_i^s = 1 \right\}$$

Sources à arbre de contexte

Définition informelle : Une source à arbre de contexte (ou VLMC : Variable Length Markov Chain) est une chaîne de Markov dont l'ordre peut dépendre du passé.

$$T = \{1, 10, \mathbf{100}, 000\}$$

$$\begin{aligned} & P(X_1^4 = 00110 | X_{-1}^0 = 10) \\ = \quad & P(X_1 = 0 | X_{-1}^0 = 10) && 3/4 \\ \times \quad & P(X_2 = 0 | X_{-1}^1 = \mathbf{100}) && 1/3 \\ \times \quad & P(X_3 = 1 | X_{-1}^2 = 1000) \\ \times \quad & P(X_4 = 1 | X_{-1}^3 = 10001) \\ \times \quad & P(X_5 = 0 | X_{-1}^4 = 100011) \end{aligned}$$



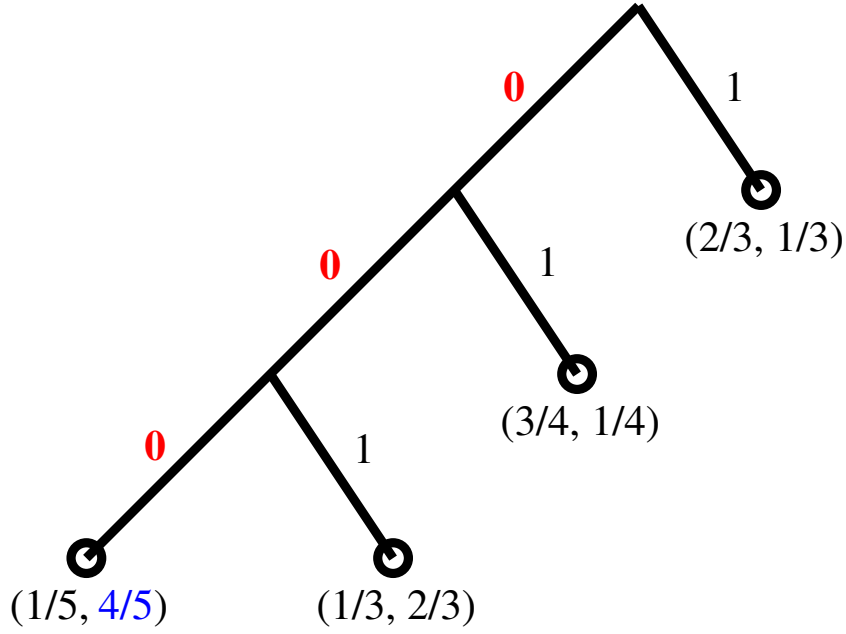
Une source à arbre de contexte est paramétrée par :

$$\Theta_T = \left\{ \left(\theta_1^s, \ldots, \theta_{|A|}^s \right) : s \in T, \sum_{i=1}^{|A|} \theta_i^s = 1 \right\}$$

Sources à arbre de contexte

Définition informelle : Une source à arbre de contexte (ou VLMC : Variable Length Markov Chain) est une chaîne de Markov dont l'ordre peut dépendre du passé.

$$T = \{1, 10, 100, \mathbf{000}\}$$



$$\begin{aligned}
& P(X_1^4 = 00110 | X_{-1}^0 = 10) \\
= \quad & P(X_1 = 0 | X_{-1}^0 = 10) && 3/4 \\
\times \quad & P(X_2 = 0 | X_{-1}^1 = 100) && 1/3 \\
\times \quad & P(X_3 = 1 | X_{-1}^2 = 1\mathbf{000}) && 4/5 \\
\times \quad & P(X_4 = 1 | X_{-1}^3 = 10001) \\
\times \quad & P(X_5 = 0 | X_{-1}^4 = 100011)
\end{aligned}$$

Une source à arbre de contexte est paramétrée par :

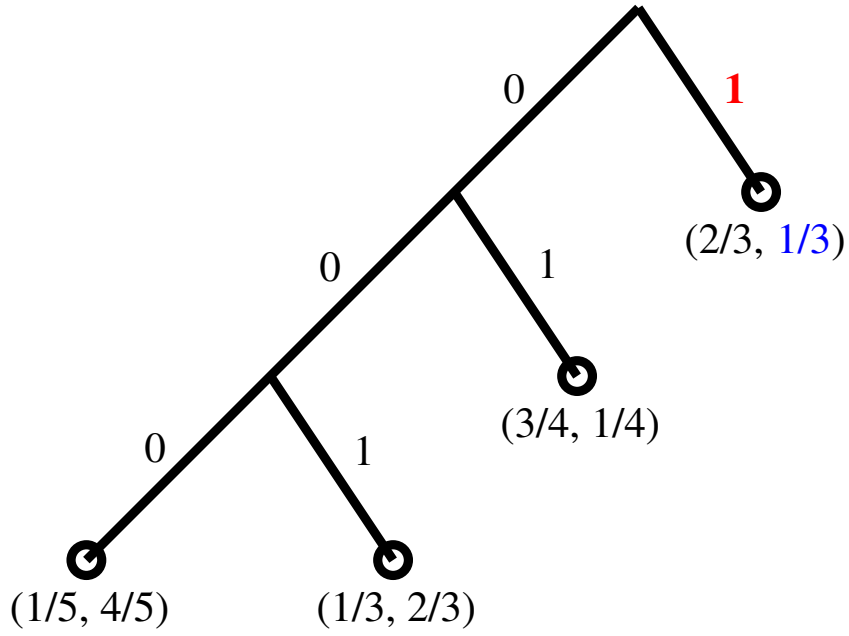
$$\Theta_T = \left\{ \left(\theta_1^s, \ldots, \theta_{|A|}^s\right) : s \in T, \sum_{i=1}^{|A|} \theta_i^s = 1 \right\}$$

Sources à arbre de contexte

Définition informelle : Une source à arbre de contexte (ou VLMC : Variable Length Markov Chain) est une chaîne de Markov dont l'ordre peut dépendre du passé.

$$T = \{1, 10, 100, 000\}$$

$$\begin{aligned} & P(X_1^4 = 00110 | X_{-1}^0 = 10) \\ = \quad & P(X_1 = 0 | X_{-1}^0 = 10) && 3/4 \\ \times \quad & P(X_2 = 0 | X_{-1}^1 = 100) && 1/3 \\ \times \quad & P(X_3 = 1 | X_{-1}^2 = 1000) && 4/5 \\ \times \quad & P(X_4 = 1 | X_{-1}^3 = 10001) && 1/3 \\ \times \quad & P(X_5 = 0 | X_{-1}^4 = 100011) \end{aligned}$$



Une source à arbre de contexte est paramétrée par :

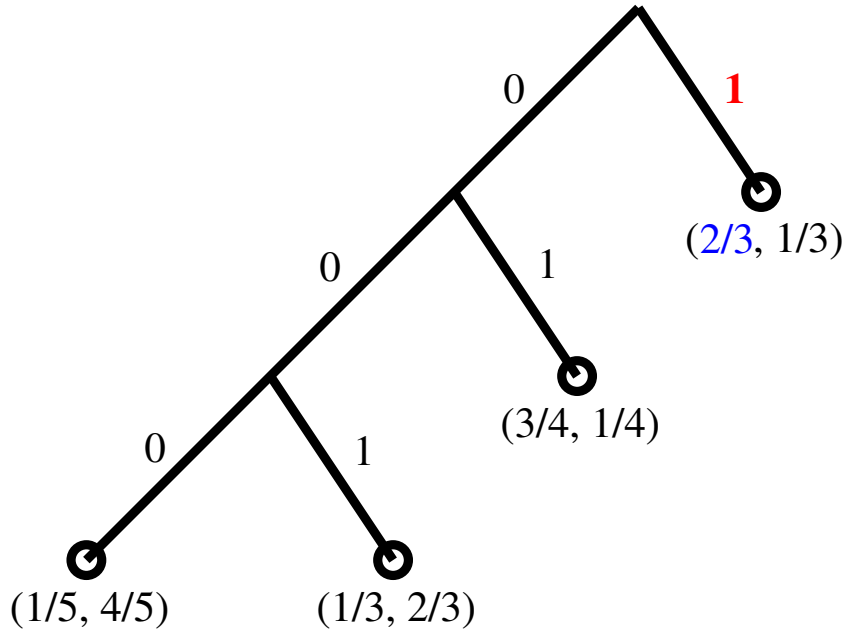
$$\Theta_T = \left\{ \left(\theta_1^s, \ldots, \theta_{|A|}^s \right) : s \in T, \sum_{i=1}^{|A|} \theta_i^s = 1 \right\}$$

Sources à arbre de contexte

Définition informelle : Une source à arbre de contexte (ou VLMC : Variable Length Markov Chain) est une chaîne de Markov dont l'ordre peut dépendre du passé.

$$T = \{1, 10, 100, 000\}$$

$$\begin{aligned} & P(X_1^4 = 00110 | X_{-1}^0 = 10) \\ = \quad & P(X_1 = 0 | X_{-1}^0 = 10) && 3/4 \\ \times \quad & P(X_2 = 0 | X_{-1}^1 = 100) && 1/3 \\ \times \quad & P(X_3 = 1 | X_{-1}^2 = 1000) && 4/5 \\ \times \quad & P(X_4 = 1 | X_{-1}^3 = 10001) && 1/3 \\ \times \quad & P(X_5 = 0 | X_{-1}^4 = 10001\mathbf{1}) && 2/3 \end{aligned}$$

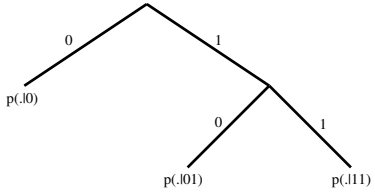


Une source à arbre de contexte est paramétrée par :

$$\Theta_T = \left\{ \left(\theta_1^s, \ldots, \theta_{|A|}^s \right) : s \in T, \sum_{i=1}^{|A|} \theta_i^s = 1 \right\}$$

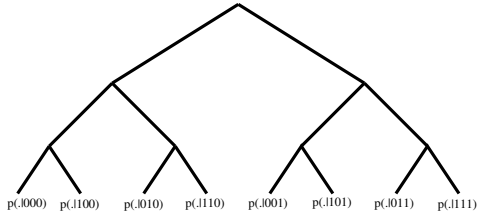
VLMC versus Chaînes de Markov

- Une source à arbre de contexte fini de profondeur d est une chaîne de Markov d'ordre d.
- Une chaîne de Markov d'ordre r est une source à arbre de contexte correspondant à un arbre complet de profondeur r.



$$\Rightarrow \quad M = \begin{pmatrix} p(.|0) \\ p(.|0) \\ p(.|01) \\ p(.|11) \end{pmatrix}$$

$$M = \begin{pmatrix} p(.|000) \\ p(.|100) \\ \vdots \\ p(.|111) \end{pmatrix} \quad \Rightarrow$$



$\Longrightarrow$ Beaucoup plus flexible : grand nombre de modèles par dimension

$\dim \Theta$	1	2	3	4	5	6	7	8	...	16
Markov(ordre)	(0) 1	(1) 1	0	(2) 1	0	0	0	(3) 1	...	(4) 1
VLMC	1	1	2	5	14	42	132	429	...	9.694.845

Choix de modèle

A partir des observations. . .

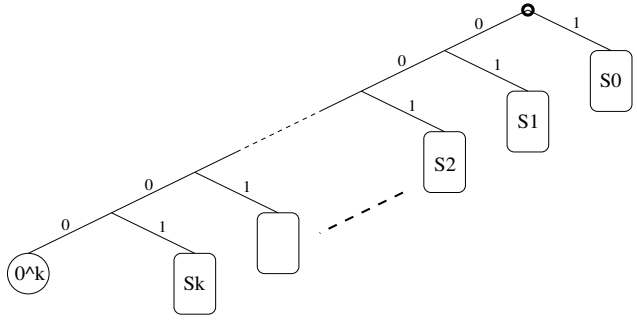
. . . on doit retrouver le modèle

1 5 3 8 6 9 4 1 2 . . . $\Rightarrow$ HMM à k états cachés.

1 1 0 0 0 1 1 0 1 0 0 0 0 1

1 4 1 2 5

$\Rightarrow$



- Nos estimateurs dérivent du principe MDL.
- Pour montrer leur consistance, on utilise des inégalités de mélange.

Plan de l'exposé

- Présentation des problèmes et modèles
- Théorie de l'information et principe MDL
- Estimateur d'arbre de contexte
- Estimateurs d'ordre de HMM

Codage Source

ATCAGAATC → 0011011000110011010

compression sans perte

Winzip, compress, etc.

But : minimiser la longueur de code

Codage Source : modèle de Shannon

Source P

= processus stationnaire sur l'**alphabet A**

ici, A={A,C,T,G}

code ϕ_n :A ->{0,1}*

ATCAGAATC → 00110110001100011010

message X_1^n (n=9)

compression sans perte

Winzip, compress, etc.

mot de code

$\phi_n(X_1^n)$

But : minimiser la longueur de code moyenne

$$E_P[|\phi_n(X_1^n)|]$$

Entropie

- **Théorème de Shannon ('48) :**

$$\mathbb{E}_{\mathbb{P}}\left[|\phi_n(x)|\right] \geqslant H_n(\mathbb{P}) \stackrel{\Delta}{=} \mathbb{E}_{\mathbb{P}}\left[-\log \mathbb{P}(X_1^n)\right],$$

  et il existe un code qui atteint la borne (à 1 près).

- Quand n augmente, $\frac{1}{n} H_n(\mathbb{P}) \longrightarrow H(\mathbb{P})$ = taux entropique de la source $\mathbb{P}$ = nb de bits nécessaires au codage de chaque symbole émis.

- **Inégalité de Kraft :** A tout code ϕ_n on peut associer la (sous-) probabilité q_n sur A^n :

$$q_n(\cdot) = 2^{-|\phi_n(\cdot)|}.$$

  Réciproquement, le codage arithmétique associe à $q_n \in \mathfrak{M}_1\left(1^n\right)$ le code ϕ_n tq :

$$|\phi_n(\cdot)| = -\log q_n(\cdot) \ \ (+Cte).$$

  Retenir que $\quad -\log q_n(x) =$ longueur de code

- Rq: théorème de Shannon = "la meilleur loi de codage est $q_n(\cdot) = \mathbb{P}(X_1^n = \cdot)$".

Codage universel

- Codeur et décodeur connaissent juste le **modèle** de la source :
  $\mathbb{P} \in \mathcal{M} = \{\mathbb{P}_\theta : \theta \in \Theta\}$
  Exemples de modèles : processus i.i.d, chaînes de Markov d'ordre k, VLMC, HMM à k états cachés...

- Il faut une seule loi de codage q_n pour toutes les sources $\mathbb{P}_\theta(\theta \in \Theta)$ redondance :

$$R_n(q_n, \theta) \triangleq \mathbb{E}_{\mathbb{P}_\theta}\left[|\phi_n(X)|\right] - H_n(\mathbb{P}_\theta) = KL(\mathbb{P}_\theta^n | q_n)$$

information de Küllback-Leibler entre la source P et la loi de codage q_n.

1. **Codage deux temps** : transmettre $\hat{\theta}\left(X_1^n\right)$, puis $\phi_{\hat{\theta}}\left(X_1^n\right)$.
   Ex: modèle sans mémoire, message $x_1^7 = AATCGTA \implies$ on envoie $(3, 1, 2, 1)$ puis le code de x avec la proba de codage $\left(\frac{3}{7}, \frac{1}{7}, \frac{2}{7}, \frac{1}{7}\right)$.
   $\implies$ Longueur de code : $\frac{|A|-1}{2} \log n - \log P_{\hat{\theta}}(x_1^n)$.

2. **Mélange** : on choisit $q_n(x) = \int P_\theta(X_1^n = x)\nu(d\theta)$.

Redondance minimax

- L'universalité du codeur q_n est mesurée par la redondance dans le pire des cas, et la meilleure possible est la *redondance minimax*

$$R_n(\mathcal{M}) = \inf_{q_n} \sup_{\theta \in \Theta} R_n(q_n, \theta)$$

- **Th. minimax [Sion -Haussler]** :
  la redondance minimax est atteinte par un *mélange* $q_n(\cdot) = \int \mathbb{P}_\theta(X_1^n = \cdot)\nu(\mathrm{d}\theta)$.
- **Th. [Rissanen '84, etc.]** : pour les modèles $\mathcal{M}$ paramétriques de dimension d,

$$R_n(\mathcal{M}) \sim \frac{d}{2} \log n$$

- $\Longrightarrow$ Deux notions de longueur de code optimale pour le message x_1^n dans $\mathcal{M}$:
  - $-\log P_{\hat{\theta}}(x_1^n) + \frac{d}{2} \log n$ bits, et
  - $-\log q_n^{\mathcal{M}}(x_1^n)$ bits, où $q_n^{\mathcal{M}}$ = mélange minimax dans $\mathcal{M}$.

Principe MDL

"Choisis le modèle qui donne *la plus courte description* des données''

- Pour utiliser ce principe, il faut une notion objective de "plus courte description des données" dans un modèle : l'approche minimax la fournit.

  longueur de description objective = celle donnée par un codeur minimax.

- Estimateur "deux temps" : $\arg\min_i \inf_{P\in M_i} -\log \hat{P}\left(x_1^n\right) + \frac{\dim M_i}{2}\log n$ coïncide avec un estimateur du maximum de vraisemblance pénalisé avec une pénalité BIC.
- Estimateur "mélange" : $\arg\min_i -\log \int_{\theta\in\Theta_i} P_\theta\left(x_1^n\right)\nu_i\left(d\theta\right)$.
- Les estimateurs BIC sont souvent consistants, alors les estimateurs de mélanges ont parfois besoin d'être pénalisés un peu.
- C'est une heuristique : il faut dans chaque cas prouver qu'elle donne un estimateur consistant, notamment grâce à des inégalités de mélange.

Plan de l'exposé

- Présentation des problèmes et modèles
- Théorie de l'information et principe MDL
- Estimateur d'arbre de contexte
- Estimateurs d'ordre de HMM

Estimateurs BIC et KT

$$\Theta_T = \left\{ \left(\theta_1^s, \ldots, \theta_{|A|}^s\right) : s \in T, \sum_{i=1}^{|A|} \theta_i^s = 1 \right\} \Longrightarrow \text{dimension du modèle} = |T|\,(|A|-1)\,.$$

- Estimateur BIC : maximum de vraisemblance pénalisé

$$\hat{T}_{BIC} = \arg\min_T - \log \widehat{P_T}(x_1^n) + \frac{|T|\,(|A|-1)}{2} \log n.$$

- Estimateur de mélange ν_T = produit de Dirichlet (1/2) (un par feuille)

$$\hat{T}_{\text{Mix}} = \arg\min_{T \in \mathbb{N}} - \log \int_{\Theta_T} P_T(x_1^n) d\nu_T(\theta).$$

- Il y a un nombre exponentiel de modèles par dimension.

Consistance

- **Théorème [Csiszár& Shields '00]:** $\hat{T}_{\mathrm{Mix}}$ n'est pas consistant : il n'arrive pas à reconnaître $\mathcal{B}\left(\frac{1}{2}\right)$.
- **Théorème [Csiszár& Talata '04]:** Si l'on restreint la minimisation aux arbres de profondeur plus petite que $D(n) = o(\log n)$, alors presque sûrement à partir d'un certain rang :

$$\hat{T}_{\mathrm{BIC}\leqslant D} = \hat{T}_{\mathrm{Mix}\leqslant D} = T_0.$$

- **Théorème [G. '05]:** $\hat{T}_{\mathrm{BIC}}$ est consistant car presque sûrement, à partir d'un certain rang, il est de taille majorée par

$$\left|\hat{T}_{\mathrm{BIC}}\right| = o\left(\frac{\log n}{\log\log\log n}\right).$$

+ Algorithme séquentiel linéaire pour calculer les estimateurs illimités $\hat{T}_{\mathrm{BIC}}$ and $\hat{T}_{\mathrm{Mix}}$, grâce aux **arbres compacts de suffixes**.

- La preuve s'appuie sur des inégalités de mélanges pour VLMC.

Plan de l'exposé

- Présentation des problèmes et modèles
- Théorie de l'information et principe MDL
- Estimateur d'arbre de contexte
- Estimateurs d'ordre de HMM

Paramétrisation du modèle

Modèle $\mathcal{M}_k (k \in \mathbb{N})$ = ensemble des HMM à k états cachés paramétré par

$$\Theta_k = \left\{ (p_{jj'})_{1 \leqslant j,j' \leqslant k} : \sum_{j'=1}^{k} p_{jj'} = 1 \right\} \times \left\{ m = (m_1, \ldots, m_k) \in \mathbb{R}^k \right\}$$

- p est la matrice de transition de la CdM cachée
- m_j la moyenne des émissions de l'état j

$$\dim \Theta_k = k(k-1) + k = k^2$$

Cas poissonnien : conditionnellement à $Z_n = j$, $X_n \sim \mathcal{P}(m_j)$.

Cas gaussien : conditionnellement à $Z_n = j$, $X_n \sim \mathcal{N}(m_j, \sigma^2)$, σ^2 fixé mais inconnu.

Réfs pour les alphabets finis : MV pénalisé [Finesso '91, Kieffer '93], procédures bayésiennes [Liu-Narayan '94], étude conjointe [Gassiat-Boucheron '03].

Deux estimateurs d'ordre

- **Maximum de vraisemblance pénalisé** :

$$\hat{k}_{ML} = \arg\min_{k\in\mathbb{N}} -\log \widehat{p_k}(x_1^n) + \mathrm{pen}(n,k).$$

- **Mélange** q_n^k :

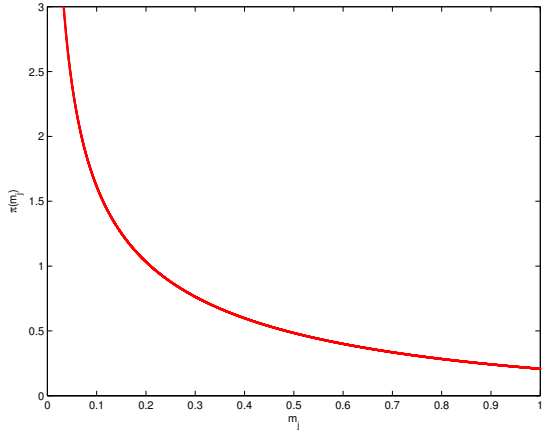
$$\hat{k}_{MIX} = \arg\min_{k\in\mathbb{N}} -\log q_n^k(x_1^n) + \mathrm{pen}(n,k).$$

- BIC: $\mathrm{pen}(n,k) = \frac{k^2}{2}\log n$, très difficile à étudier.
- Cas gaussien : on pert l'interprétation en termes de codage, mais on n'a besoin que d'inégalités de mélange !
- On pénalise aussi le mélange - c'est souvent nécessaire, cf. $B(1/2)$ pour les VLMC.

Mélange dans $\mathcal{M}_k$

Le mélange q_n^k est obtenu avec le prior ν_k sur Θ_k tel que, pour un certain $\tau > 0$, on ait sous ν_k :

- p et m sont indépendantes,
- la distribution initiale $p^o_{j'} = 1/k$ pour tout $j' \leqslant k$ est déterministe,
- les vecteurs $(p_{jj'} : j' \leqslant k)$ $(j \leqslant k)$ sont indépendants et suivent une loi de Dirichlet de paramètre $(1/2, \ldots, 1/2)$,
- les moyennes $m_1, \ldots, m_k$ sont indépendantes, identiquement distribuées selon la loi normale $\mathcal{N}_{0,\tau}$ dans le cas gaussien, et avec la loi Gamma$(\tau, 1/2)$ dans le cas poissonnien.



$\Longrightarrow$ on utilise des priors conjugués, avec des paramètres inspirés des mélanges de Krichevsky-Trofimov qui sont optimaux pour les alphabets finis.

Inégalités de mélange et consistance

- Inégalités de mélange :

  Poisson : $0 \leqslant \sup_{\theta\in\Theta_k} \log \mathbb{P}_\theta(X_1^n) - \log q_n^k(X_1^n) \leqslant \frac{k^2}{2}\log n + k\tau X_{(n)} + c_{kn}$.

  Gauss : $0 \leqslant \sup_{\theta\in\Theta_k} \log f_\theta(X_1^n) - \log q_n^k(X_1^n) \leqslant \frac{k^2}{2}\log n + \frac{k}{2\tau^2}|X|^2_{(n)} + d_{kn}$.

- Posons $S_{kn} = D_{kn} + k(k+1)\varphi_n \log n$ dans le cas gaussien, et
  $S_{kn} = E_{kn} + k(k+1)\frac{\log n}{\sqrt{\log\log n}}$ dans le cas poissonien.

  Si

$$\mathrm{pen}(n,k) = \sum_{\ell=1}^{k} \frac{\ell^2+\alpha}{2}\log n + C_{kn} + S_{kn},$$

  alors presque sûrement à partir d'un certain rang on a $\hat{k}_{ML} = k_0$.

  Si

$$\mathrm{pen}(n,k) = \sum_{\ell=1}^{k-1} \frac{\ell^2+\alpha}{2}\log n + S_{kn},$$

  alors presque sûrement à partir d'un certain rang on a $\hat{k}_{MIX} = k_0$.

Remarques sur la preuve

- On a besoin de pénaliser plus que BIC, à cause des maxima et de la technique de preuve.
- Comportements différents des maxima : dans le cas Poissonnien $X_{(n)} = o(\log n)$, dans le cas Gaussien $|X|^2_{(n)} = \Theta(\log n)$.
- Preuves "imbriquées" : même pour $\hat{k}_{ML}$ on utilise les inégalités de mélange.
- La sous-estimation est facile à éviter, pas la sur-estimation !
- Pour des mélanges gaussiens et poissonniens (états cachés iid), on peut faire la même chose en remplaçant le nombre de degrés de liberté k^2 par $2k - 1$.
- Avantage : pas besoin de borne a priori sur l'ordre ni sur les paramètres des lois d'émission.
- Inconvénient : en pratique la vraisemblance n'est pas facile à maximiser $\Longrightarrow$ algorithme EM.