Arbres de Contextes Probabilisés

Aurélien Garivier

TELECOM ParisTech

Plan

Le modèle

- Motivations
- Noyaux et Arbres de Contextes Probabilisés

Procédures de choix de modèle

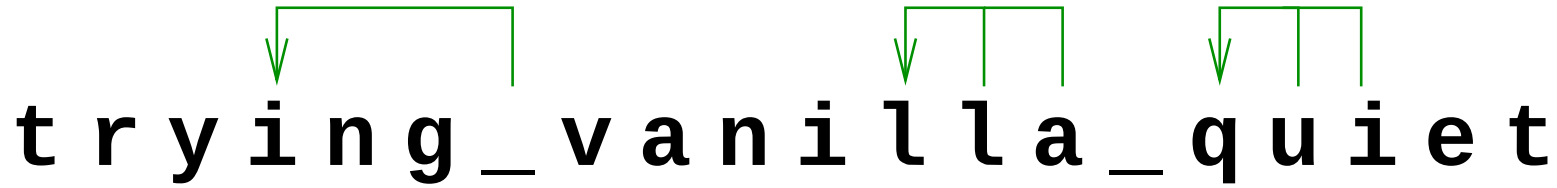
- Deux algorithmes
- Maximum de vraisemblance pénalisée
- Consistance et déviations de processus auto-normalisés
- Estimation jointe de deux sources partageant de l'information

Agrégation de modèles en théorie de l'information

- Codage universel
- Mélange et double mélange

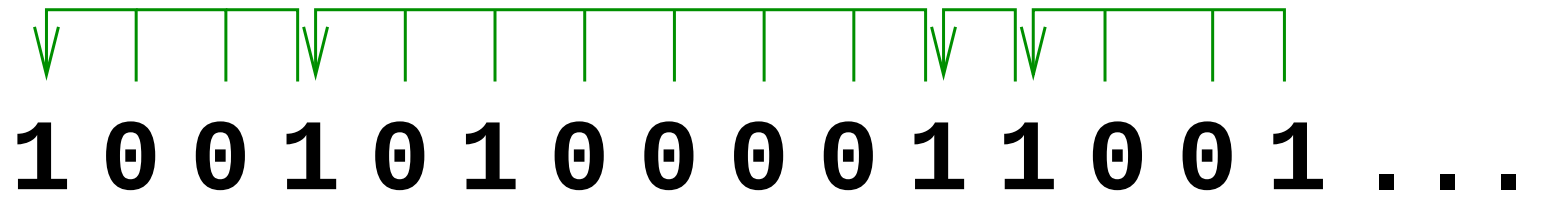
Mémoire adaptative

- Compression de données:



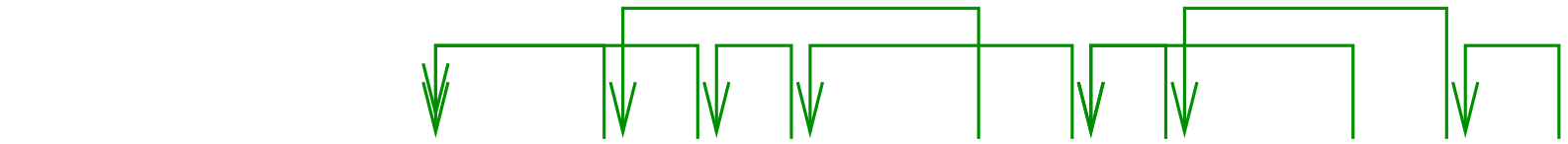
- Linguistique:

- Processus de renouvellement:



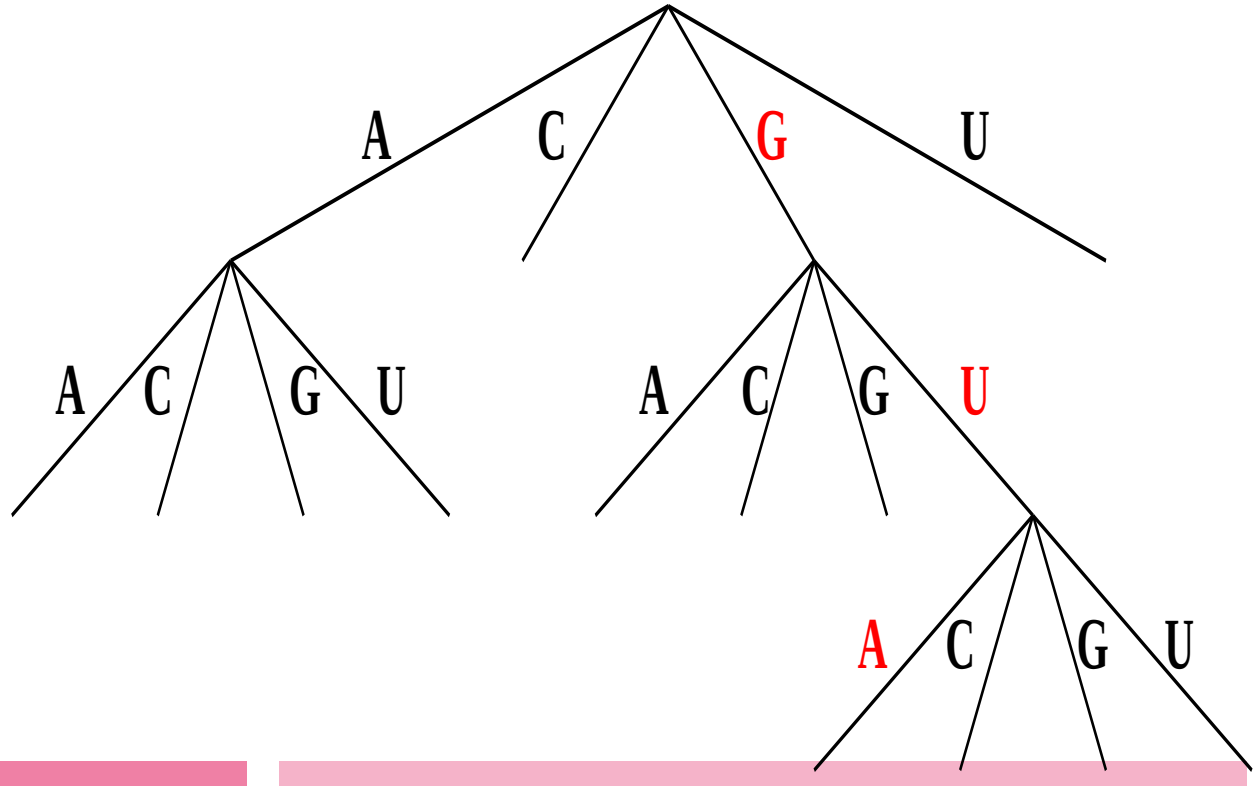
- Musique, biologie, optimisation, ...

Séquences biologiques



start

Modèle de mémoire donné par:



Limites des modèles markoviens

- Compression: $|A| = 2$, $k = 8 \implies \dim \Theta = 256$
- Séquences biologiques: $|A| = 4$, $k = 6 \implies \dim \Theta \approx 12000$
- Linguistique: $|A| = 3000$, $k = 10 \implies \dim \Theta = \ldots$
- Processus de renouvellement : mémoire infinie

Besoin d'une plus grande flexibilité: il faut pouvoir mettre beaucoup de mémoire uniquement quand c'est nécessaire!

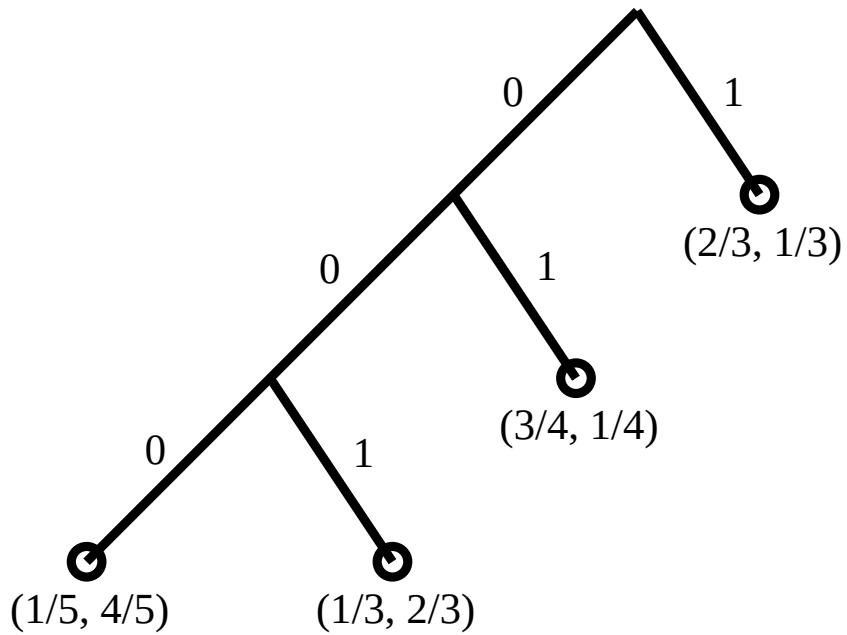
Arbres de contextes probabilisés

Un arbre de contexte probabilisé (CTS) ou Chaîne de Markov d'ordre variable (VLMC) est une chaine de Markov dont l'ordre est autorisé à dépende des valeurs prises dans le passé.

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

$$P(X_1^4 = 00110 | X_{-1}^0 = 10)$$

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



Arbres de contextes probabilisés

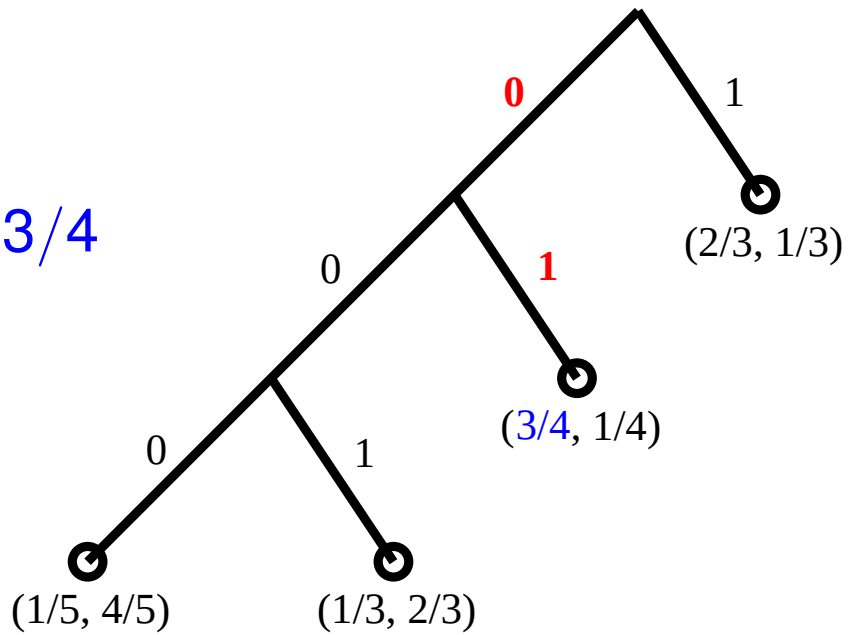
Un arbre de contexte probabilisé (CTS) ou Chaîne de Markov d'ordre variable (VLMC) est une chaine de Markov dont l'ordre est autorisé à dépende des valeurs prises dans le passé.

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

$$P(X_1^4 = 00110 | X_{-1}^0 = 10)$$

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

$3/4$



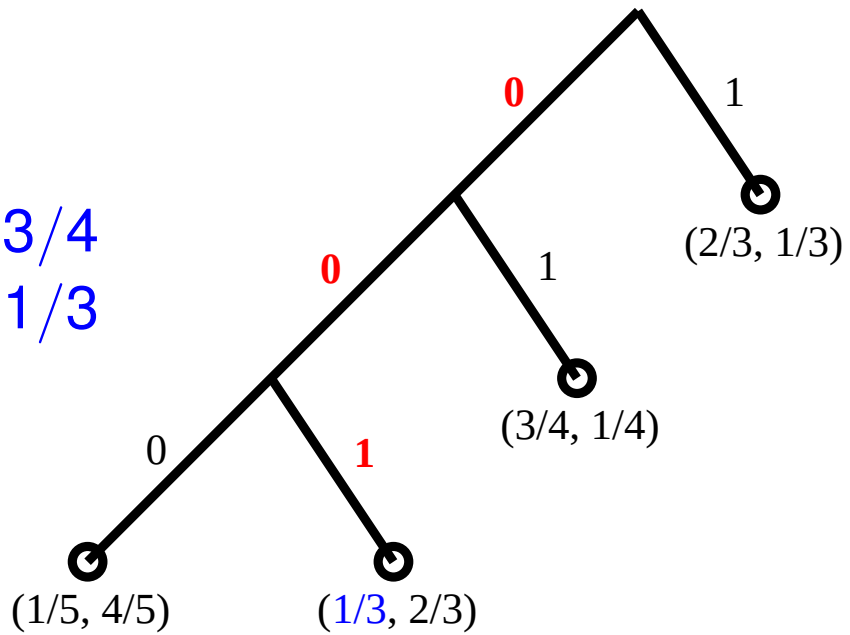
Arbres de contextes probabilisés

Un arbre de contexte probabilisé (CTS) ou Chaîne de Markov d'ordre variable (VLMC) est une chaine de Markov dont l'ordre est autorisé à dépende des valeurs prises dans le passé.

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

$$P(X_1^4 = 00110 | X_{-1}^0 = 10)$$

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



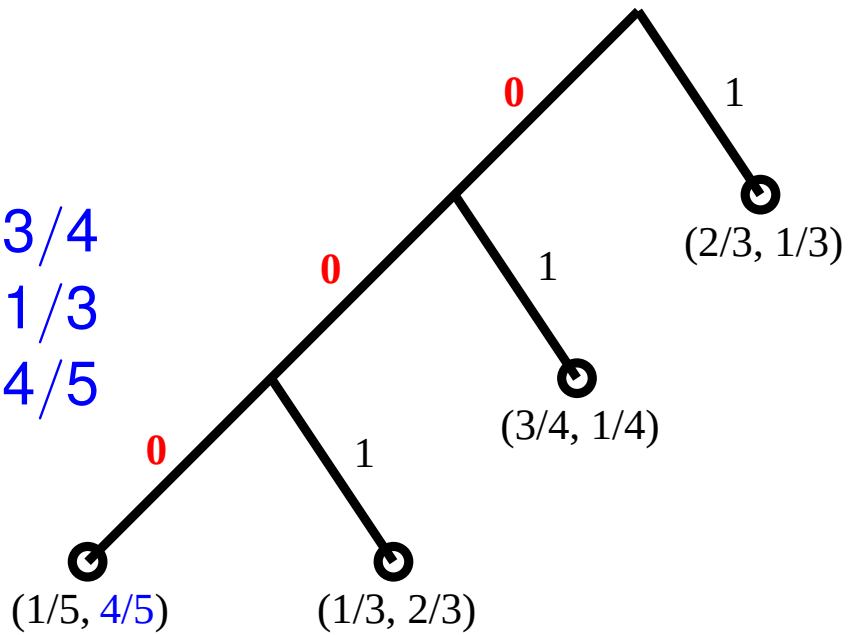
Arbres de contextes probabilisés

Un arbre de contexte probabilisé (CTS) ou Chaîne de Markov d'ordre variable (VLMC) est une chaine de Markov dont l'ordre est autorisé à dépende des valeurs prises dans le passé.

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

$$P(X_1^4 = 00110 | X_{-1}^0 = 10)$$

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



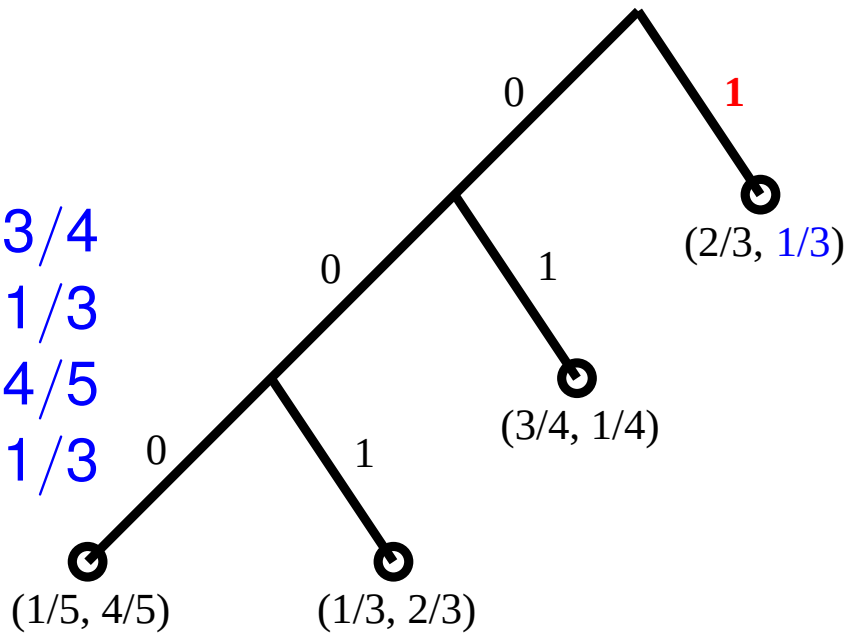
Arbres de contextes probabilisés

Un arbre de contexte probabilisé (CTS) ou Chaîne de Markov d'ordre variable (VLMC) est une chaine de Markov dont l'ordre est autorisé à dépende des valeurs prises dans le passé.

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

$$P(X_1^4 = 00110 | X_{-1}^0 = 10)$$

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



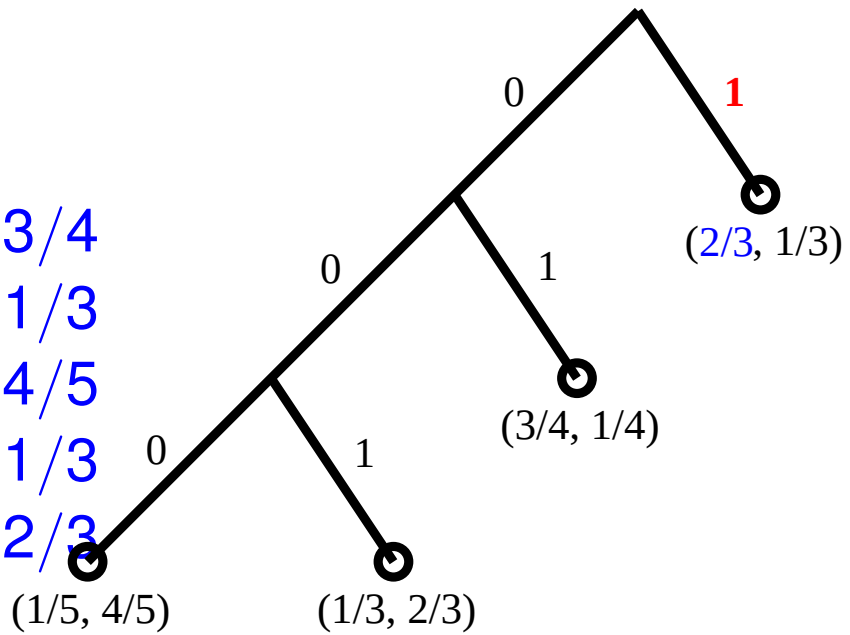
Arbres de contextes probabilisés

Un arbre de contexte probabilisé (CTS) ou Chaîne de Markov d'ordre variable (VLMC) est une chaine de Markov dont l'ordre est autorisé à dépende des valeurs prises dans le passé.

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

$$P(X_1^4 = 00110 | X_{-1}^0 = 10)$$

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



Plan

Le modèle

- Motivations
- Noyaux et Arbres de Contextes Probabilisés

Procédures de choix de modèle

- Deux algorithmes
- Maximum de vraisemblance pénalisée
- Consistance et déviations de processus auto-normalisés
- Estimation jointe de deux sources partageant de l'information

Agrégation de modèles en théorie de l'information

- Codage universel
- Mélange et double mélange

Les Noyaux

- On munit l'espace des suites semi-infinies $A^{-\mathbb{N}^*}$ de la distance ultra-métrique suivante :

$$\delta(\underline{w}, \underline{z}) = 2^{\sup\{k \leq 0 : x_k \neq y_k\}}$$

- Les boules (ouvertes et fermées) de cet espace sont les

$$\mathcal{T}(s) = \{\underline{w} \in A : x_{-|s|+1:0} = s\}$$

pour $s \in A^*$.

- Un *noyau* est une application $P : A^{-\mathbb{N}} \to \mathcal{M}_1(A)$, et l'image de $\underline{w} \in A^{-\mathbb{N}}$ est notée $P(\cdot|\underline{w})$.
- Un noyau est dit *continu* s'il l'est comme application entre les espaces métriques $\left(A^{-\mathbb{N}}, \delta\right)$ et $(\mathcal{M}_1(G), |\cdot|_{TV})$.

Processus associé à un noyau

- Vieux problème: pour un noyau P donné, existe-t-il un (unique) processus stationnaire ergodique dont P est une version des probabilités conditionnelles?
- **Théorème [Fernandez-Galves '02]**: si $\sum_{a \in A} \inf_{s \in T} P(a|s) > 0$ et si les

$$\beta_k = \max_{a \in A} \sup \left\{ |P(a|s) - P(a|t)| : (s,t) \in T^2 \text{ and } s^0_{-k+1} = t^0_{-k+1} \right\}$$

  sont sommables, alors la réponse est oui.
- Idée : simulation exacte par *couplage par le passé*.

Arbres de contextes probabilisés

- Un *arbre complet de suffixes* (CSD) est un ensemble $T \subset A^*$ qui définit une partition de $A^{-\mathbb{N}}$:

$$A^{-\mathbb{N}} = \bigcup_{s \in T} \mathcal{T}(s)$$

- Ainsi, T est une CSD ssi

$$\forall x_{-\infty}^0 \in A^{-\mathbb{N}}, \exists! L \in \mathbb{N} : x_{-L}^0 \in T;$$

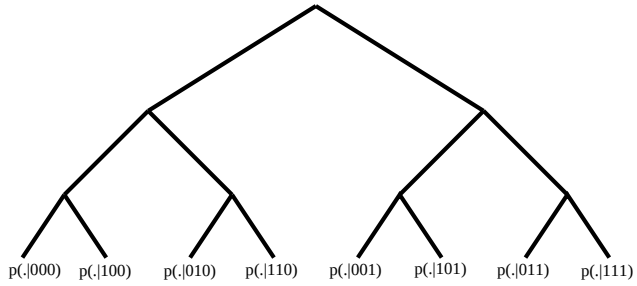
- Le noyau d'une CTS est constant sur chaque composante d'un CSD : il est donc caractérisé par la donnée d'une famille de $|T|$ distributions conditionnelles $\{P_T(\cdot|s) : s \in T\}$:

$$\forall x_{-\infty}^0 \in A^{-\mathbb{N}} P_T(\cdot|x_{-\infty}^0) = P_T(\cdot|x_{-L}^0).$$

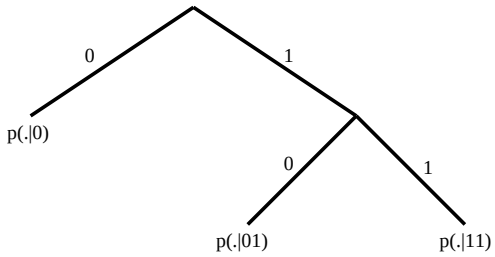
CTS vs Chaines de Markov

- Les Chaînes de Markov d'ordre r sont des CTS dont l'arbre est complet de profondeur r.

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



- Les CTS bornés de profondeur d sont des chaînes de Markov d'ordre d.



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

Vraisemblance et estimation

- La vraisemblance admet une écriture aussi simple que pour les chaînes de Markov:

$$P_T(x_1^n|x_{-\infty}^0) = \prod_{i=1}^{n} P_T(x_i|x_{i-L_i}^{i-1}) = \prod_{s\in T}\prod_{i\in I_s} P_T(x_i|s),$$

où $I_s = \{i \in \{1,\dots,n\} : x_{i-|s|}^{i-1} = s\}$.

- L'estimateur du maximum de vraisemblance dans le modèle T est donc: pour $s \in T$,

$$\hat{P}_T(\cdot|s) = \frac{N(sa)}{N(s)},$$

où $N(s) = \sum_{i=1}^{n} \mathbb{1}_{x_{i-|s|}^{i-1}=s} = |I_s|$.

Plan

Le modèle
- Motivations
- Noyaux et Arbres de Contextes Probabilisés

Procédures de choix de modèle
- Deux algorithmes
- Maximum de vraisemblance pénalisée
- Consistance et déviations de processus auto-normalisés
- Estimation jointe de deux sources partageant de l'information

Agrégation de modèles en théorie de l'information
- Codage universel
- Mélange et double mélange

Algorithme Context

- Introduit par Rissanen en 1981, ressemble à CART.
- Pour tout $s \in A^*$, on calcule la mesure de distortion

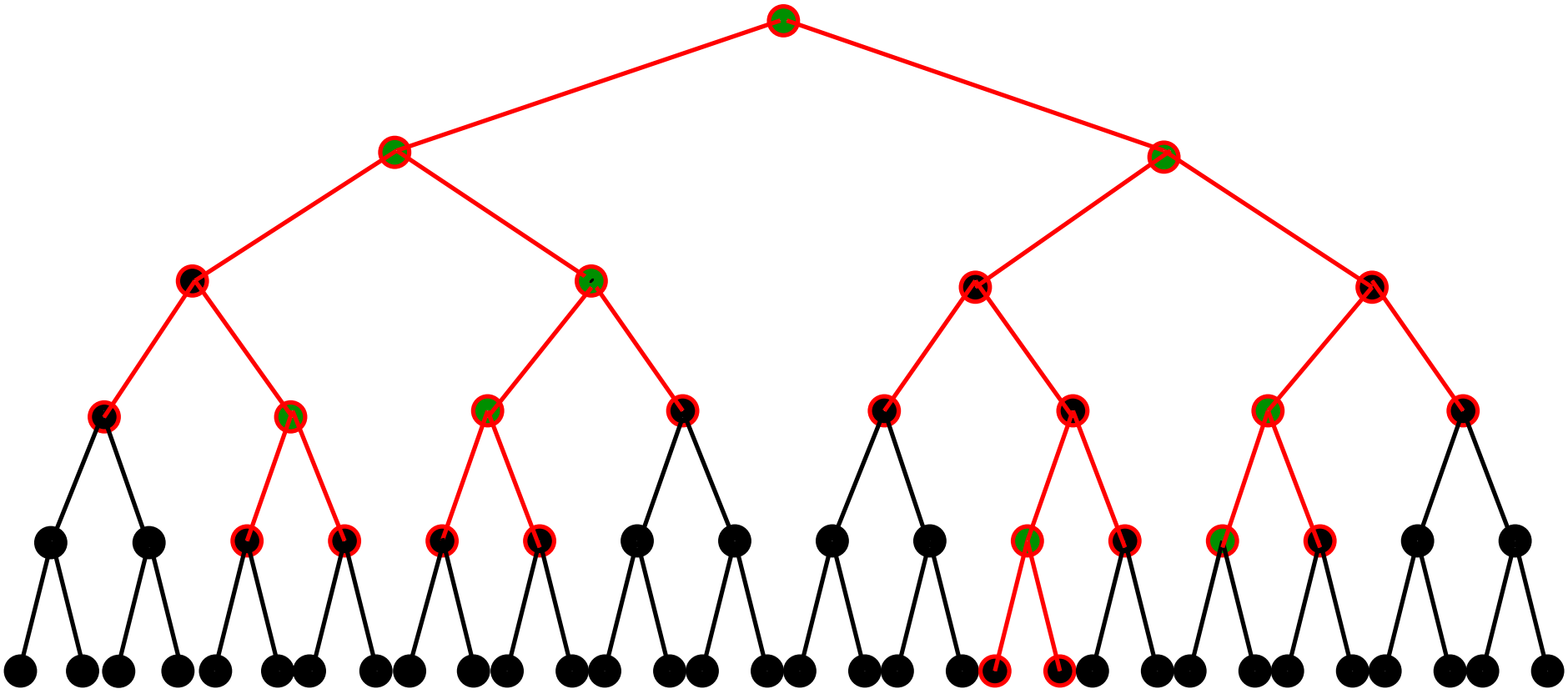
$$\delta(s) = \max_{a \in A} \left\| \hat{P}(\cdot|s) - \hat{P}(\cdot|as) \right\| .$$

- On garde tous les $s \in A^*$ tels que

$$\exists u \in A^* : \delta(us) \geq \varepsilon(n)$$

comme noeuds internes de $\hat{T}_C$. Ainsi $\hat{T}_C$ contient tous les noeuds actifs, leurs ancêtres et leur enfants.

Algorithm Context: Illustration



Noeuds actifs - Arbre estimé $\hat{T}_C$

Plan

Le modèle
- Motivations
- Noyaux et Arbres de Contextes Probabilisés

Procédures de choix de modèle
- Deux algorithmes
- Maximum de vraisemblance pénalisée
- Consistance et déviations de processus auto-normalisés
- Estimation jointe de deux sources partageant de l'information

Agrégation de modèles en théorie de l'information
- Codage universel
- Mélange et double mélange

Maximum de vraisemblance pénalisée

- On choisit

$$\hat{T}_{pml} = \underset{T}{\operatorname{argmax}} \log \hat{P}_T(x_1^n | x_{-\infty}^0) + pen(n, T),$$

où $pen(n, T)$ est une fonction de pénalité croissante en n et $|T|$.

- Pénalité BIC

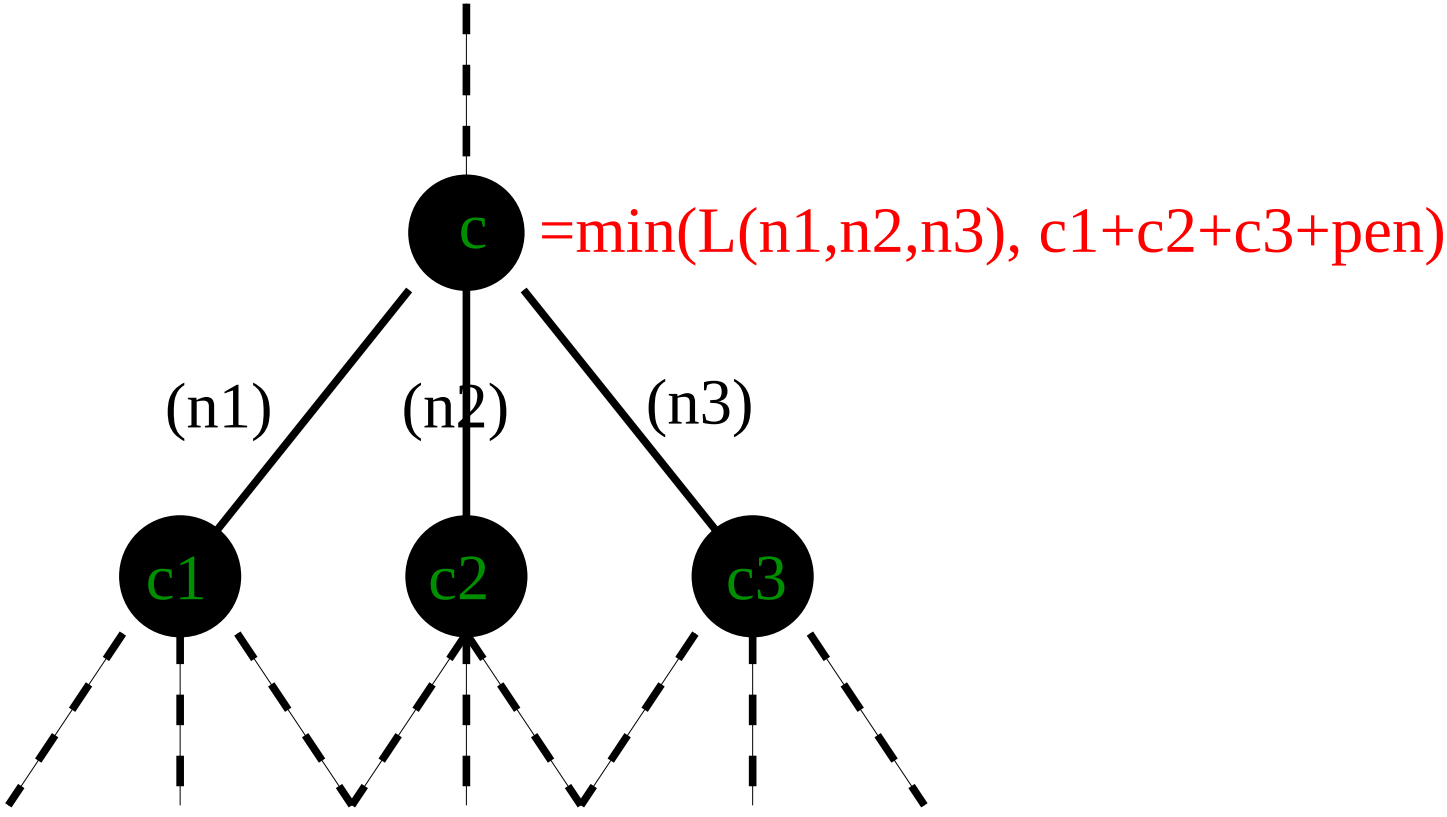
$$pen(n, T) = \frac{|T|(|A| - 1)}{2} \log n.$$

- Principe Minimum Description Length: choisir le modèle qui donne la plus courte description des données (= qui permet de mieux les compresser)
- Variante MDL: estimé de Krichevski-Trofimov

$$\hat{T}_{KT} = \underset{T}{\operatorname{argmax}} \log KT_T(x_1^n | x_{-\infty}^0).$$

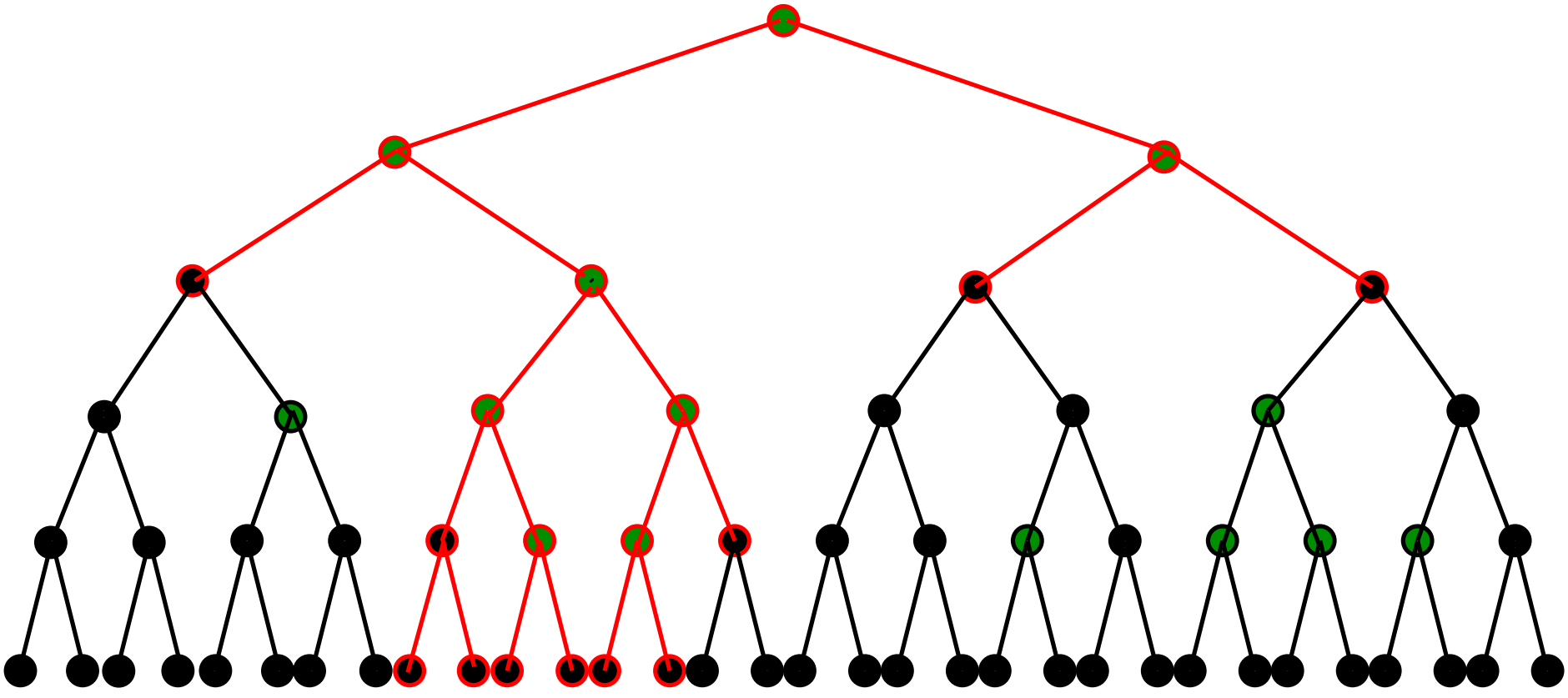
Calcul de $\hat{T}_{pml}$

Procédure récursive “Context Tree Maximization” : un noeud s est dit actif si $x_{l(s)}$ se code mieux avec de la mémoire.



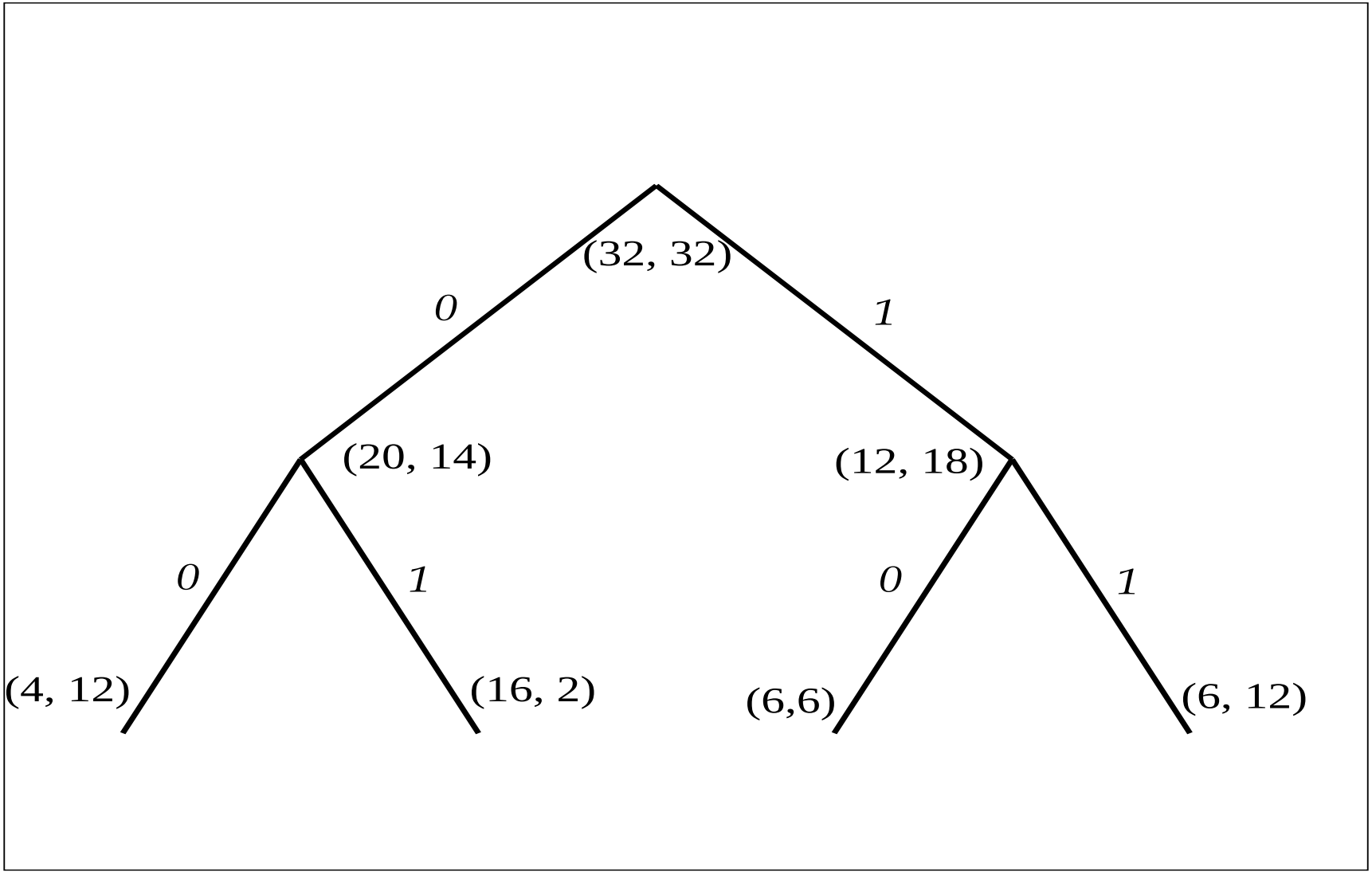
En partant du sommet, on ne garde que les noeuds actifs.

PML: Illustration

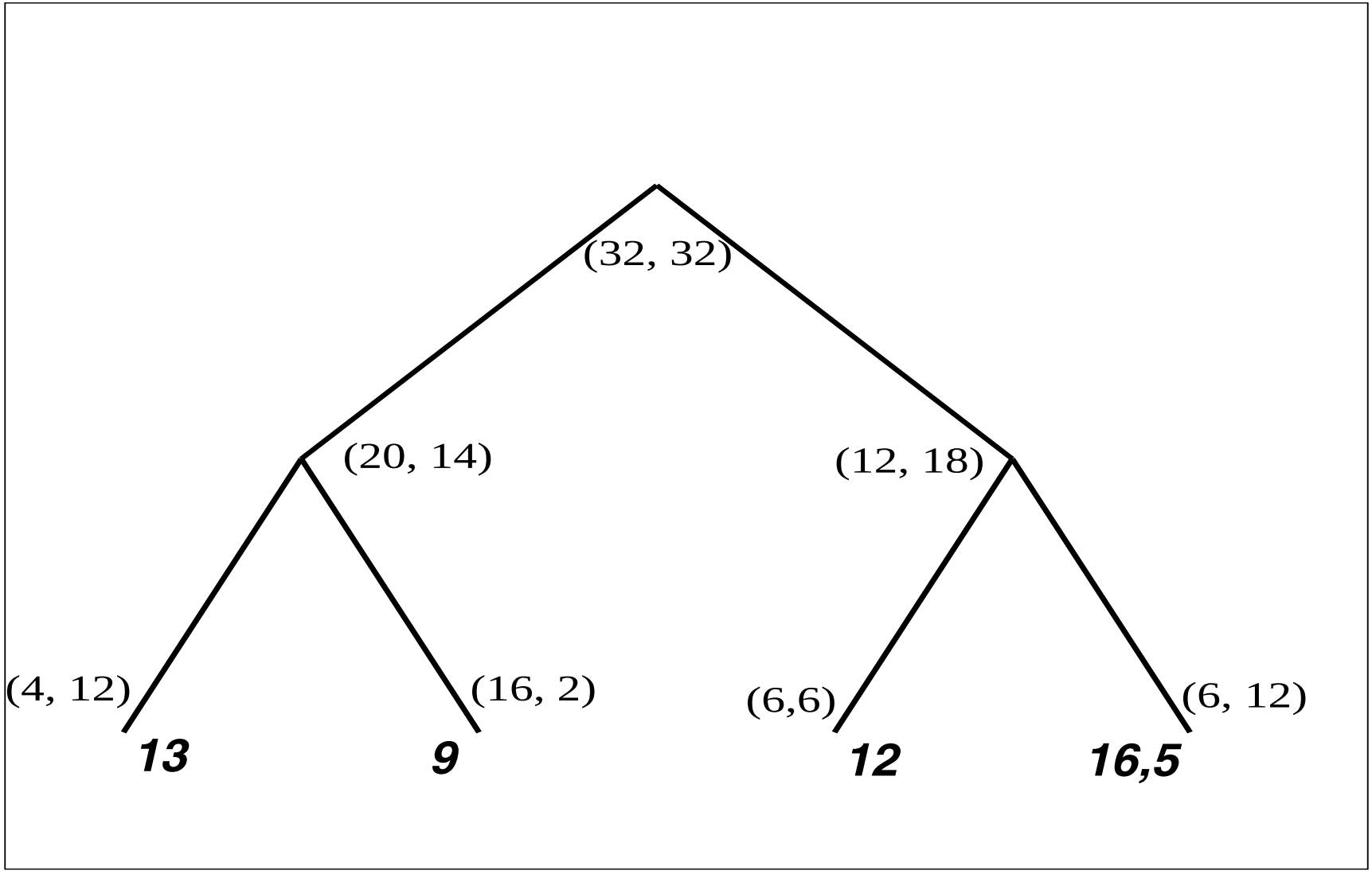


Noeuds actifs - Arbre estimé $\hat{T}_{pml}$

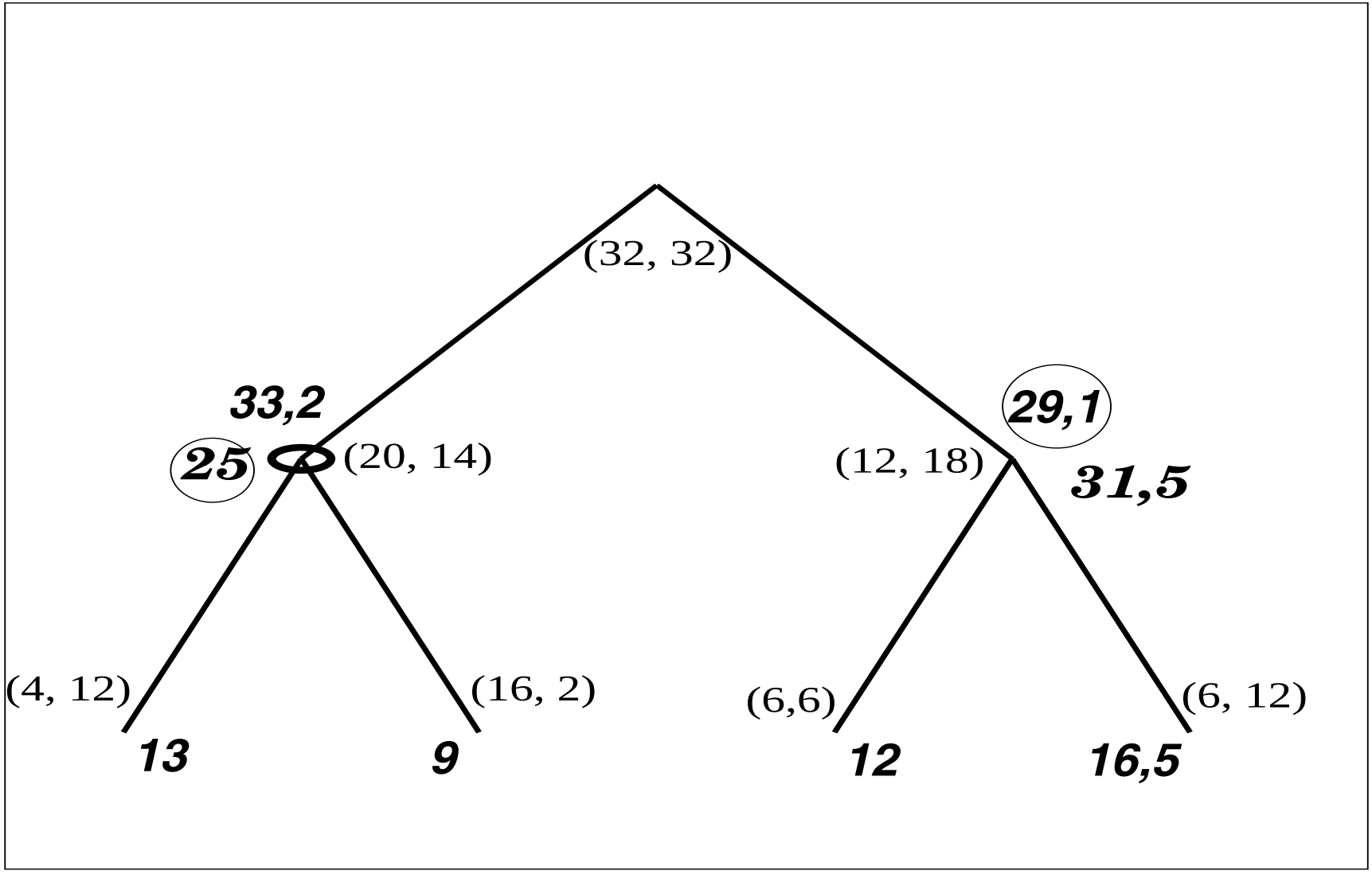
PML: Exemple



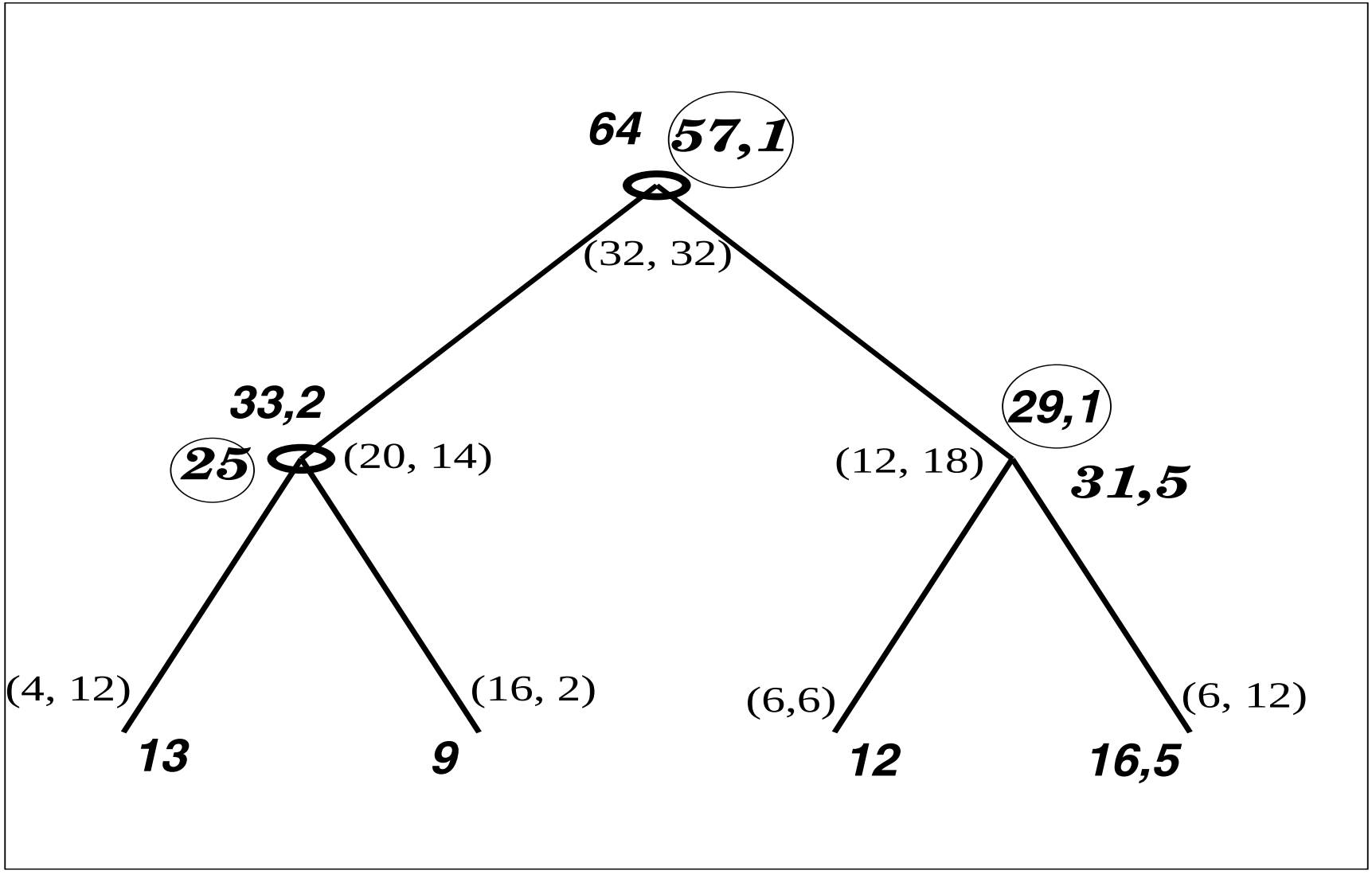
PML: Exemple



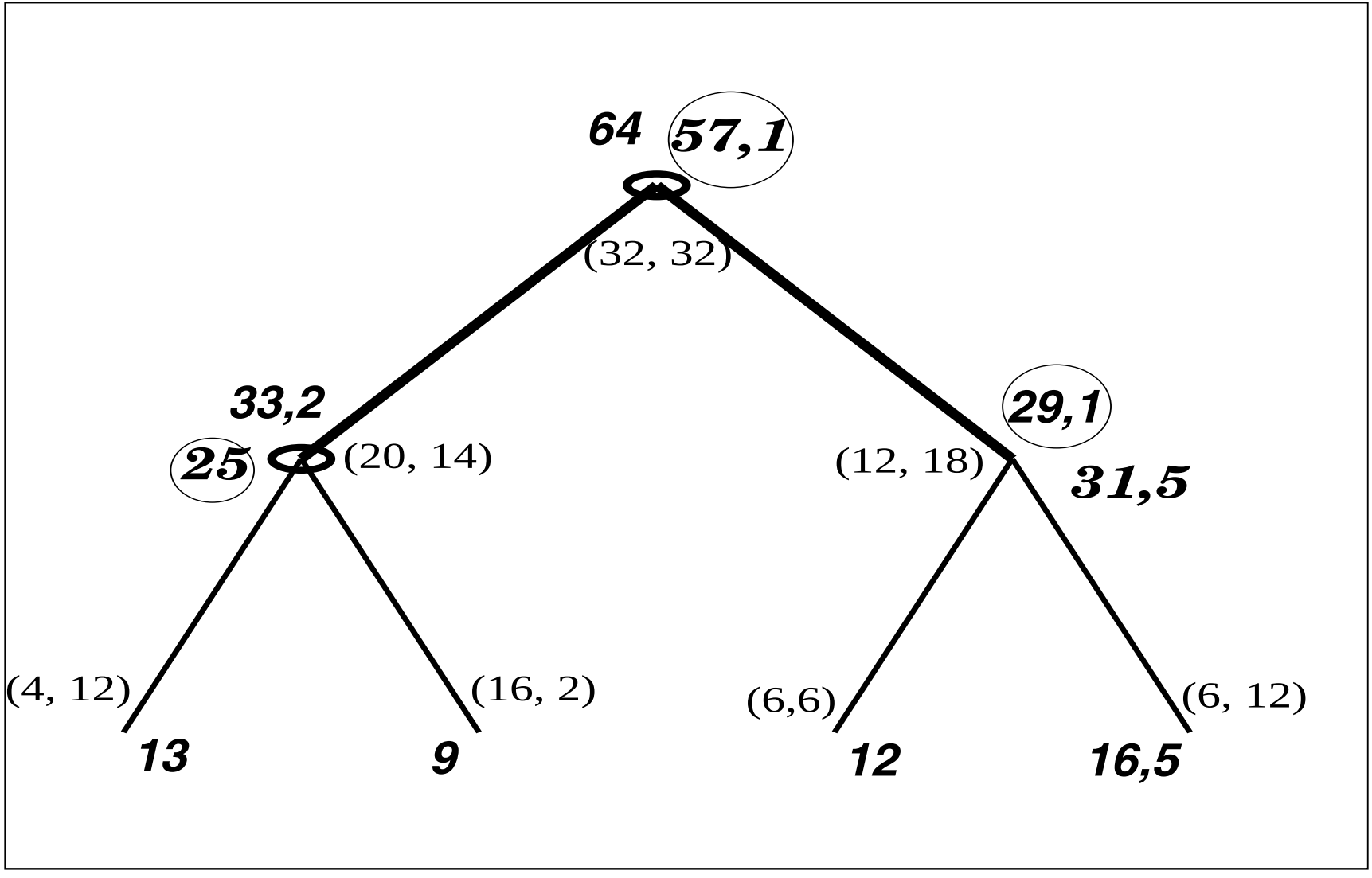
PML: Exemple



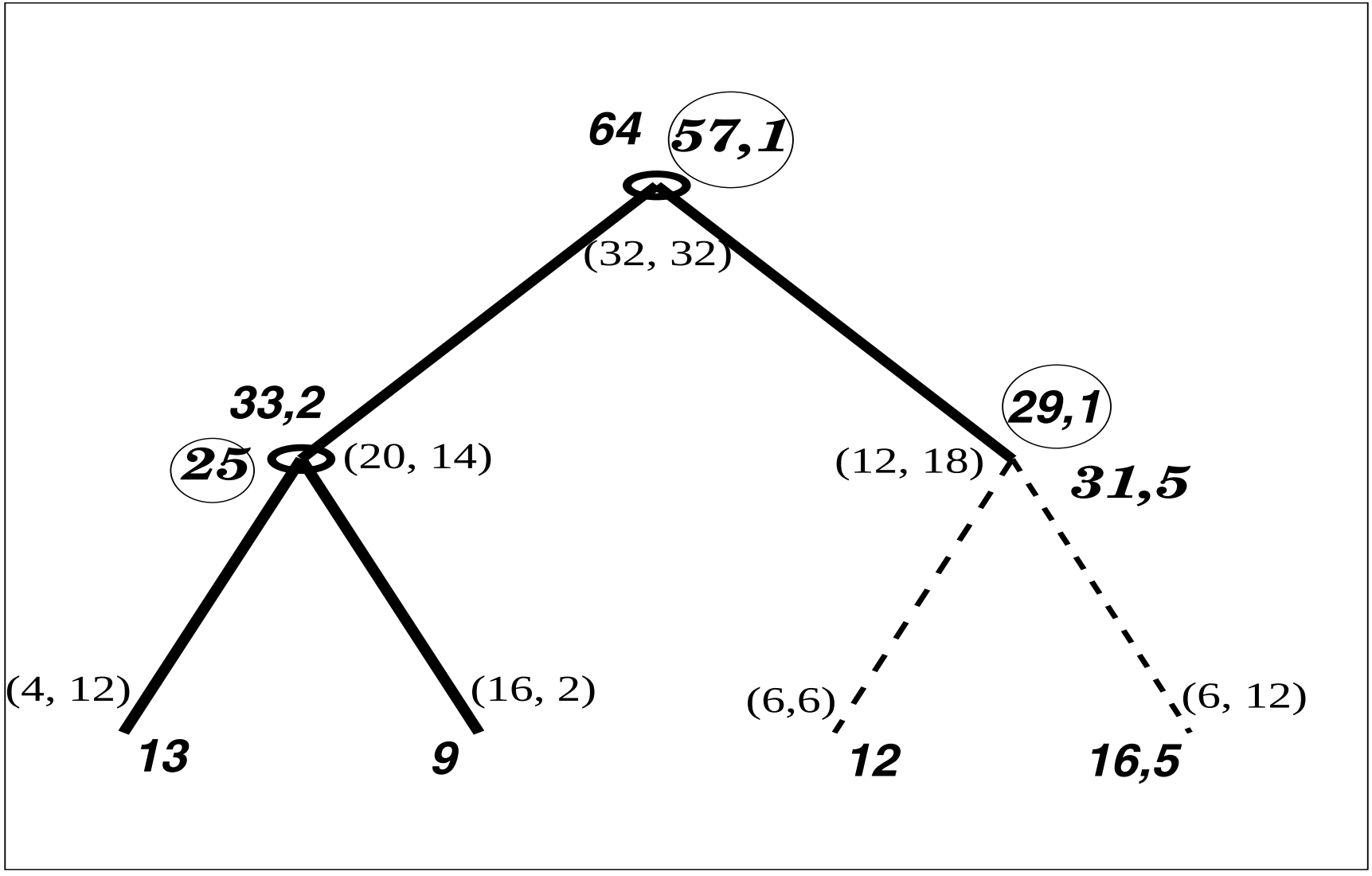
PML: Exemple



PML: Exemple



PML: Exemple



Comparaison des deux estimateurs

- - Pour l'algorithme Context, l'activité d'un noeud se mesure uniquement dans ce noeud.
  - Pour PML, l'activité d'un noeud prend en compte tout ce qui est sous ce noeud.
- - L'algorithme Context garde une branche dès que son noeud le plus profond est actif.
  - PLM ne garde que des noeuds actifs.
- $\Longrightarrow$ pour des choix de paramètres comparables, on montre que l'agorithme Context sélectionne systématiquement des arbres plus grands que PLM.

Plan

Le modèle
- Motivations
- Noyaux et Arbres de Contextes Probabilisés

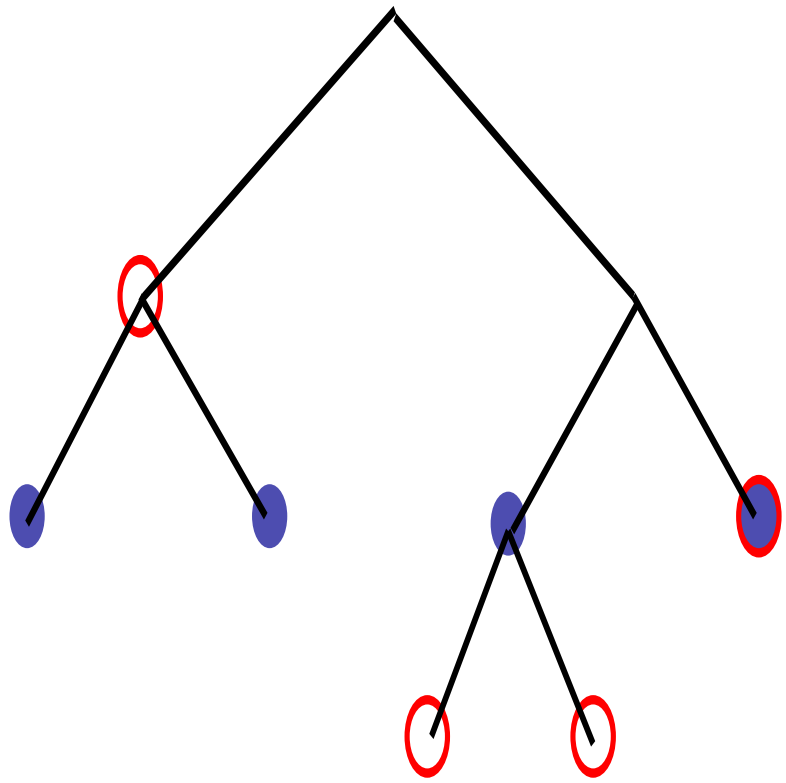
Procédures de choix de modèle
- Deux algorithmes
- Maximum de vraisemblance pénalisée
- Consistance et déviations de processus auto-normalisés
- Estimation jointe de deux sources partageant de l'information

Agrégation de modèles en théorie de l'information
- Codage universel
- Mélange et double mélange

Sous-estimation et Sur-estimation

Deux erreurs d'estimation sont possibles:

1. sous-estimation:
   $\exists s \in T_0 : s \notin \hat{T}$
   $\Longrightarrow$ "facilement" évitée (régime de grandes déviations) à vitesse exponentielle
2. sur-estimation:
   $\exists s \in \hat{T} : s \notin T_0$
   $\Longrightarrow$ plus délicat, pas de taux exponentiels [Finesso '92]



Résultats asymptotiques

- **Théorème [Rissanen '81, ...]**: Pour un arbre fini T_0, si $\varepsilon(n) = C\log(n)/n$, alors quand $n \to \infty$:

$$P(\hat{T}_C \neq T_0) \to 0.$$

- **Théorème [Csiszár and Talata '06, Garivier '06]**: Si $K \in \mathbb{N}^*$ et si $\hat{T}_{pml}$ maximise la vraisemblance pénalisée parmi les arbres de hauteur $D(n) = o(\log n)$, alors

$$\hat{T}_{pml}^{|K} = T_0^{|K}$$

presque sûrement pour n assez grand. Pour un arbre fini T_0, pas besoin de restreindre la maximisation.

- Résultats non asymptotiques : cf [Galves-Maume-Deschamps Schmitt '05, Leonardi '08, Garivier-Leonardi] avec hypothèses.

Comment étudier la convergence ?

- Pour l'algorithme Context on doit contrôler

$$\|\hat{P}(\cdot|s) - P(\cdot|s)\|.$$

- Pour le maximum de vraisemblance pénalisée, il faut majorer

$$KL\left(\hat{P}(\cdot|s), P(\cdot|s)\right).$$

- Dans les deux cas, on se ramène à l'étude des maxima d'une martingale "moyenne normalisée" du type:

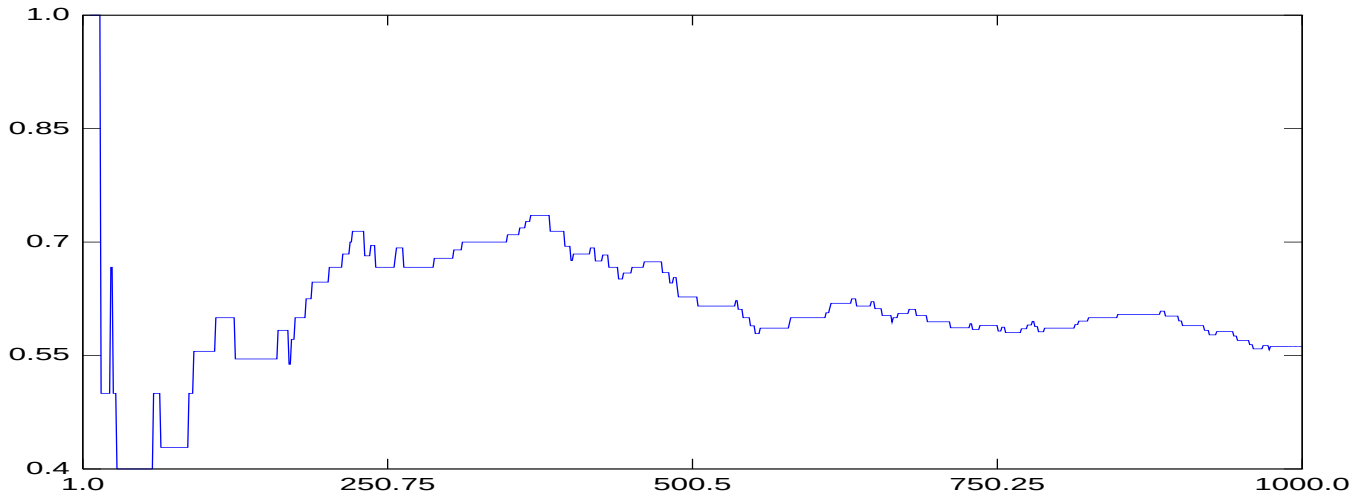
$$Z_t = \frac{1}{\sqrt{N_t(s)}} \sum_{u=1}^{t} (\mathbb{1}_{\{X_u=a} - P(a|s)\}) \mathbb{1}_{\{X_{u-|s|}^{u-1}=s\}}.$$

- Bornes non-asymptotiques adapter les preuves de LLI.

Quelle est l'activité normale d'un noeud ?

Pour tout contexte possible s, l'estimateur du maximum de vraisemblance de la loi conditionnelle est :

$$\forall a \in A, \hat{P}(a|s) = \frac{1}{n} \sum_{k=1}^{n} \mathbb{1}_{\{X_k = a\}} \mathbb{1}_{\{X_{k-|s|}^{k-1} = s\}}$$



Plan

Le modèle
- Motivations
- Noyaux et Arbres de Contextes Probabilisés

Procédures de choix de modèle
- Deux algorithmes
- Maximum de vraisemblance pénalisée
- Consistance et déviations de processus auto-normalisés
- Estimation jointe de deux sources partageant de l'information

Agrégation de modèles en théorie de l'information
- Codage universel
- Mélange et double mélange

Formulation du problème

- $X = (X_n)_{n\in\mathbb{Z}}$ et $Y = (Y_n)_{n\in\mathbb{Z}}$ sont des CTS indépendantes de lois P_X et P_Y
- P_X et P_Y partagent certains contextes et certaines lois conditionnelles, mais pas toutes
- On note $\tau_X = \tau_0 \cup \tau_1$ le CSD de P_X, et $\tau_Y = \tau_0 \cup \tau_2$ le CSD de P_Y
- On veut estimer τ_X et τ_Y: est-ce possible de faire mieux qu'en traitant les deux problèmes séparément?

Estimation jointe par maximum de vraisemblance

- La vraisemblance jointe s'écrit:

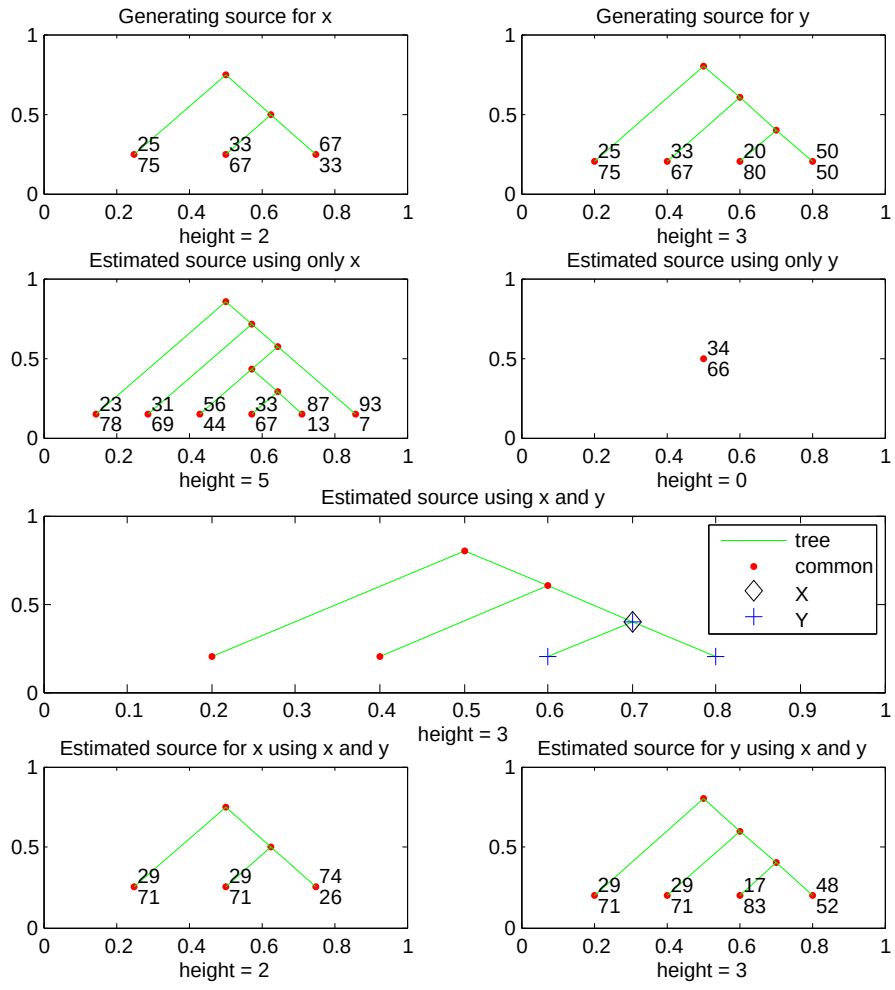
$$\sum_{s\in\tau_1}\sum_{a\in A} N_{n,X}(s,a)\log\left(\frac{N_{n,X}(s,a)}{N_{n,X}(s)}\right)$$
$$+\sum_{s\in\tau_2}\sum_{a\in A} N_{m,Y}(s,a)\log\left(\frac{N_{m,Y}(s,a)}{N_{m,Y}(s)}\right)$$
$$+\sum_{s\in\tau_0}\sum_{a\in A}\left[N_{n,X}(s,a)+N_{m,Y}(s,a)\right]\log\left(\frac{N_{n,X}(s,a)+N_{m,Y}(s,a)}{N_{n,X}(s)+N_{m,Y}(s)}\right)$$

- Procédure récursive "à la Context Tree Maximization" consistante [Galves-Garivier-Gassiat]

Exemple de résultats

Quand on répète l'expérience 100 fois avec des échantillons de taille $n_X = n_Y = 400$, voici ce qui se passe :

- τ_X est bien estimé avec X seul 96 fois, et avec notre algo 80 fois : la performance est donc dégradée.
- Par contre, τ_Y est bien estimé avec Y seul 47 fois, et avec notre algo 85 fois
- Surtout, τ_X et τ_Y sont séparément bien estimés 45 fois, et conjointement 67 fois.



Plan

Le modèle
- Motivations
- Noyaux et Arbres de Contextes Probabilisés

Procédures de choix de modèle
- Deux algorithmes
- Maximum de vraisemblance pénalisée
- Consistance et déviations de processus auto-normalisés
- Estimation jointe de deux sources partageant de l'information

Agrégation de modèles en théorie de l'information
- Codage universel
- Mélange et double mélange

Codage source

ATCAGAATC → 0011011000110011010

compression sans perte

Winzip, compress, etc.

But : minimiser la longueur de code

Codage source : Shannon

Source P

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

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

code ϕ_n :A ->{0,1}*

ATCAGAATC → 001101100011001101O

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 (Shannon '48)** :

$$\mathbb{E}_P\left[|\phi_n(x)|\right] \geq H_n(P) \stackrel{\text{def}}{=} \mathbb{E}_P\left[-\log P^n(X_1^n)\right]$$

. . . et il existe un code qui atteint ce taux (à 1 bit près).

- En outre,

$$\frac{1}{n} H_n(P) \to H(P)$$

taux entropique de la source P
=nombre minimal de bit par symbole.

Loi de codage

- A chaque code $\phi_n(x)$ on peut associer une (sous-)probabilité q_n sur A^n telle que
$$q_n(\cdot) = 2^{-|\phi_n(\cdot)|}$$
- Inversement, grâce au codage arithmétique, on peut associer à toute (sous-)probabilité q_n sur A^n un code ϕ_n tel que $|\phi_n(\cdot)| = -\log q_n(\cdot)$ $(+Cte)$.

Conclusion: code $\phi_n \iff$ loi de codage q_n

En particulier, $-\log q_n(x) =$ longueur de code.

- Le théorème de Shannon '48 dit que la meilleure loi de codage est la loi du processus !
- La perte obtenue en utilisant une autre loi q_n est appelée regret
$$-\log q_n\left(X_1^n\right) - \left(-\log P^n\left(X_1^n\right)\right) = \log \frac{P^n\left(X_1^n\right)}{q_n\left(X_1^n\right)}.$$

Codage universel

- Que faire si la loi de la source est inconnue ??
- . . . et si on veut un seul code pour plusieurs sources à la fois ?

$\Longrightarrow$ on a besoin d' une seule loi de codage q_n pour tout une classe de sources

$$\Lambda = \{P_\theta, \theta \in \Theta\}$$

c'est donc un problème d'estimation de densité. . .

$\Longrightarrow$ Redondance inévitable :

$$\begin{aligned} \mathbb{E}_{P_\theta}\left[\left|\phi\left(X_1^n\right)\right|\right] - H\left(X_1^n\right) &= \mathbb{E}_{P_\theta}\left[\log q_n\left(X_1^n\right) + \log P_\theta\left(X_1^n\right)\right] \\ &= KL\left(P_\theta, q_n\right) \end{aligned}$$

. . . avec perte logarithmique = information de Küllback-Leibler entre P_θ et q_n.

Exemples de codeurs universels

1. **Codage deux-temps**
   - Code d'abord $\hat{\theta}\left(X_1^n\right) = \text{argmin}_{\theta\in\Theta} - \log P_\theta\left(X_1^n\right)\ldots$
   - $\ldots$ puis X_1^n avec la loi de codage $P_{\hat{\theta}}^n$.

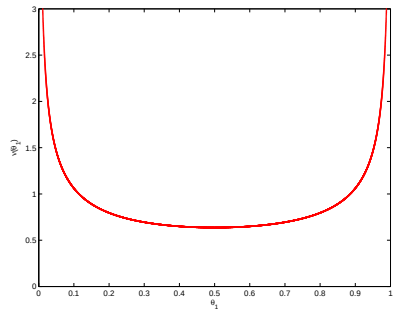
   Ex: (i.i.d.)

   $$\Longrightarrow \quad \text{regret } \frac{|A|-1}{2}\log n.$$

   $X_1^9 = AAATACAGT : \hat{\theta} = (5,1,2,1)/9$
2. **Codage par mélange** si ν est un prior sur Θ, on prend

$$q_n^\nu(x_1^n) = \int_\Theta P_\theta\left(x_1^n\right)\ d\nu\left(\theta\right)$$

Ex: (i.i.d.) $\rightarrow \nu = \text{Dirichlet}\left(\frac{1}{2},\ldots,\frac{1}{2}\right)$



Mesures d'universalité

1. **Redondance dans le pire des cas :**

$$R^+(q_n, \Theta) = \sup_{\theta \in \Theta} \mathbb{E}_{P_\theta}\left[\log \frac{P_\theta^n(X_1^n)}{q_n(X_1^n)}\right] = \sup_{\theta \in \Theta} KL(P_\theta, q_n)$$

2. **Redondance bayésienne** avec le prior π:

$$R_\pi^-(q_n, \Theta) = \mathbb{E}_\pi\left[\mathbb{E}_{P_\theta}\left[\log \frac{P_\theta^n(X_1^n)}{q_n(X_1^n)}\right]\right] = \mathbb{E}_\pi\left[KL(P_\theta, q_n)\right]$$

$$\Longrightarrow \quad R^-(q_n, \Theta) \leq R^+(q_n, \Theta)$$

Mesures de complexité

1. **Redondance minimax**:

$$R_n^+(\Theta) = \inf_{q_n} R^+(q_n, \Theta) = \min_{q_n} \max_{\theta} KL\left(P_\theta^n, q_n\right)$$

2. **Redondance bayésienne**:

$$R_{\pi,n}^-(\Theta) = \min_{q_n} \mathbb{E}_\pi \left[KL\left(P_\theta^n, q_n\right)\right]$$

$$R_n^-(\Theta) = \max_{\pi} \min_{q_n} \mathbb{E}_\pi \left[KL\left(P_\theta^n, q_n\right)\right]$$

Theorème maximin (Haussler '97, Sion)

$$R_n^-(\Theta) = R_n^+(\Theta),$$

et la redondance minimax est atteinte par un mélange π.

Plan

Le modèle
- Motivations
- Noyaux et Arbres de Contextes Probabilisés

Procédures de choix de modèle
- Deux algorithmes
- Maximum de vraisemblance pénalisée
- Consistance et déviations de processus auto-normalisés
- Estimation jointe de deux sources partageant de l'information

Agrégation de modèles en théorie de l'information
- Codage universel
- Mélange et double mélange

Mélange pour les VLMC

- Soit T un arbre de contexte à $|T|$ feuilles (= $|T|$ contextes).
- En considérant un produit de $|T|$ lois de Dirichlet$\left(\frac{1}{2},\ldots,\frac{1}{2}\right)$ comme a priori sur Θ_T, on définit le mélange de Krichevky-Trofimov q_T^ν qui vérifie:

$$q_T^\nu(x_1^n|x_{-\infty}^0) = \prod_{s\in T} q^\nu\left(T(x,s)\right).$$

- **Proposition (Shtarkov&al '93)**

$$-\log q_T^\nu(x_1^n|x_{-\infty}^0) \leq \inf_{\theta\in\Theta_T} p_\theta(x_1^n|x_{-\infty}^0) + \frac{|A|-1}{2}|T|\log^+\frac{n}{T} + |T|\log m + m - 1.$$

CTW : un double mélange

La relation

$$\pi(T) = 2^{-2|T|+1}$$

définit une loi de probabilité sur l'ensemble $\mathcal{T}$ de tous les arbres de contextes.
On définit donc la loi de probabilité **Context Tree Weighting** :

$$q_n^{\mathrm{CTW}}(x_1^n) = \sum_T \pi(T) q_T^{\nu}(x_1^n)$$

elle se calcule efficacement par CTM.

CTW : un double mélange

Proposition (Sharkov & al '93)

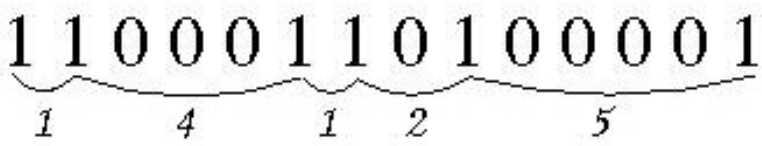
$$-\log q_n^{\mathrm{CTW}}(x_1^n|x_{-\infty}^0) \leq \inf_{T\in\mathcal{T}} \inf_{\theta\in\Theta_T} p_\theta(x_1^n|x_{-\infty}^0) + \frac{|A|-1}{2}|T|\log\frac{n}{T} + |T|(2+\log m) + m - 2.$$

$\Longrightarrow$ CTW est adaptatif sur les classes de CTS.

Quelle est sa performance sur des classes plus grandes ?

Processus de renouvellement

- Exemple de classe plus grande : processus de renouvellement.

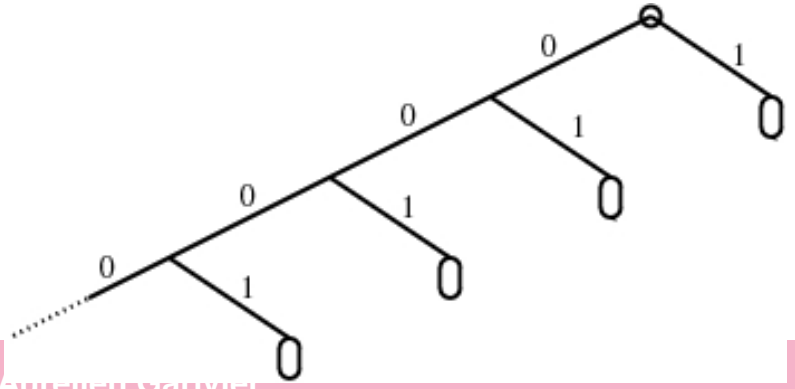


- Résultat de [Csiszár - Shields '96] : la redondance minimax y est:

$$R_n^- (\mathcal{R}) \sim R_n^* (\mathcal{R}) = \Theta\left(\sqrt{n}\right)$$

- Mais la preuve n'est pas constructive : existe-t-il un algorithme de compression qui se comporte bien sur cette classe ?

CTW est un bon candidat car les processus de renouvellement sont des "CTS infinis"



Résultats

- **Theorem (G. '06):** Il existe C_1 et C_2 telles que la redondance de sur la classe $\mathcal{R}$ des processus de renouvellement vérifie:

$$C_1 \sqrt{n} \log n \leq R_n^* \left(q_n^{\mathrm{CTW}}, \mathcal{R} \right) \leq C_2 \sqrt{n} \log n.$$

- **Theorem (G. '06):** I existe C_1 et C_2 telles que la redondance de sur la classe $\mathcal{MR}$ des processus de renouvellement markoviens vérifie:

$$C_3 n^{\frac{2}{3}} \log n \leq R_n^* \left(q_n^{\mathrm{CTW}}, \mathcal{MR} \right) \leq C_4 n^{\frac{2}{3}} \log n.$$

Commentaires

- Résultat d'adaptivité pour CTW sur une classe massive, équilibre biais/variance.
- Si la loi de renouvellement est bornée, CTW s'adapte aussi avec un regret en $O(\log n)$.
- Nécessite la prise en compte de contextes profonds dans le double mélange ($\Longrightarrow$ ne pas couper les arbres à profondeur $\log n$).