Redondance de l'algorithme Context Tree Weighting sur les processus de renouvellement

Aurélien Garivier

Université Paris Sud

Orsay

Plan de l'exposé

- Préliminaire
- Chaînes de Markov à ordre variable
- L'algorithme Context Tree Weighting
- Processus de Renouvellement
- Redondance de $\mathcal{CTW}$ sur les PR

Notations

- Soit A un alphabet fini de lettres. Les **mots** sur A sont les éléments de $A^* = \bigcup_{n=0}^{\infty} A^n$.
- Pour $x, s \in A^*$, le **nombre d'occurences** de s dans x est $N_x(s) = \sum_{i=1}^{|x|-s+1} \mathbb{1}_{x_i^{i+|s|-1}=s}$.
- Pour $P \in \mathbb{P}(A)$, l'**entropie** de P est ($\log = \log_2$) $H(P) = \sum_{a \in A} P(a) \log \frac{1}{P(a)}$.
- Pour $x \in A^n$, l'**entropie empirique** (coût de codage) de x est :

$$H(x) = nH(\hat{p}_x) = \sum_{a \in A} N_x(a) \log \frac{n}{N_x(a)}.$$

Redondances

Pour un code $C_n : A^n \to \{0,1\}$ et une source $\mathbb{P}$, on définit :

- la longueur de code “idéale” pour $x \in A^n$: $-\log \mathbb{P}(x)$
- la *redondance ponctuelle* : $R(C_n|P)(x) = L(C_n(x)) + \log \mathbb{P}(x)$
- la *redondance moyenne* : $\bar{R}(C_n|P) = \mathbb{E}_{\mathbb{P}}\left[L(C_n(X)) + \log \mathbb{P}(X)\right]$;
- la *redondance maximale* $R^*(C_n|P) = \max_x L(C_n(x)) + \log \mathbb{P}(x)$.

Codage universel

- Pour une classe de processus $\mathcal{S}$, on définit les *redondances minimax* :

$$\begin{aligned} \bar{R}_n(\mathcal{S}) &= \inf_C \sup_{\mathbb{P}\in\mathcal{S}} \bar{R}(C_n|P) \qquad \text{et} \\ R^*_n(\mathcal{S}) &= \inf_C \sup_{\mathbb{P}\in\mathcal{S}} R^*(C_n|P) \end{aligned}$$

- Pour les classes paramétriques à k degrés de liberté, $\bar{R}_n(\mathcal{S})$ et $R^*_n(\mathcal{S})$ sont au premier ordre égales à $\frac{k}{2}\log n$.
- Il n'existe pas de fonction $g(n) = o(n)$ telle que pour la classe des processus stationnaires ergodiques, on ait $\bar{R}_n(\mathcal{S}) = g(n)$.

Le mélange de Krichevsky-Trofimov

- Notons $\Theta_A = \left\{\theta \in [0,1]^A : \sum_{a\in A} \theta_a = 1\right\}$ et $\mathbb{P}_\theta$ la loi du processus iid sur $A^{\mathbb{Z}}$ défini par $\mathbb{P}_\theta(X_0 = a) = \theta_a$.
- Mélange de Krichevski-Trofimov sur A^n :

$$\mathcal{KT}(x_1^n) = \int_{\theta\in\Theta_A} P_\theta(x_1^n)\, \frac{\Gamma\left(\frac{|A|}{2}\right)}{\sqrt{|A|}\Gamma(\frac{1}{2})^{|A|}} \prod_{a\in A} \theta_a^{-1/2} \mathrm{d}\theta_a$$

- Calcul itératif : comme un MV avec |A| “demi-expériences” avant de commencer : $\mathcal{KT}(011) = \frac{1/2}{1} \times \frac{1/2}{2} \times \frac{3/2}{3} = \frac{1}{16}$.

$$-\log \mathcal{KT}(x) \leqslant \inf_{\theta\in\Theta} -\log P_\theta(x) + \frac{1}{2}\left(|A|-1\right)\log n + |A|/2.$$

Plan de l'exposé

- Préliminaire
- Chaînes de Markov à ordre variable
- L'algorithme Context Tree Weighting
- Processus de Renouvellement
- Redondance de $\mathcal{CTW}$ sur les PR

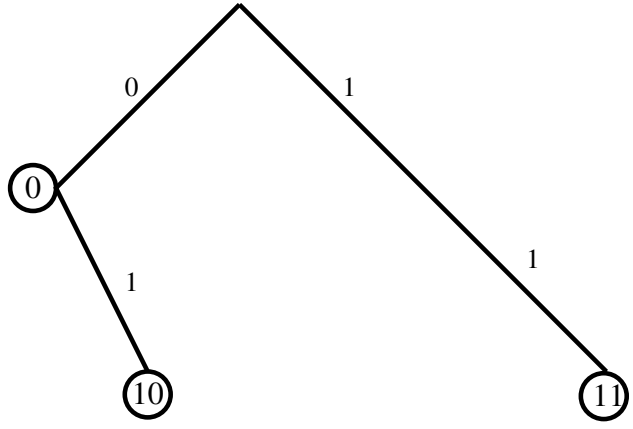
Dictionnaires complets de suffixes

- Un ensemble de mots $\mathcal{T}$ est un **dictionnaire complet de suffixes** (DCS) si $\forall x_{-\infty}^{0} \in A^{\mathbb{Z}-}, \exists! k \in \mathbb{N} : x_{-k}^{0} \in \mathcal{T}$.
- Exemples :
  - $\mathcal{T} = \{00, 10, 1\}$ en est un.
  - $\mathcal{T} = \{0, 10, 11\}$ n'en est pas un, car $\mathcal{T}(\ldots 010)$ n'est pas défini.
  - $\mathcal{T} = \{10^k : k \in \mathbb{N}\}$ en est un infini.
- On note $\mathcal{T}(y)$ l'unique suffixe d'un mot y dans $\mathcal{T}$.
- Pour $x = x_{-\infty}^{n}$ et $s \in \mathcal{T}$, le sous-mot de x qui apparaît dans le contexte s est $\mathcal{T}(x, s) = \bigodot_{i=1, \mathcal{T}\left(x_{-\infty}^{i-1}\right)=s}^{n} x_i$

Tries et arbres de suffixes

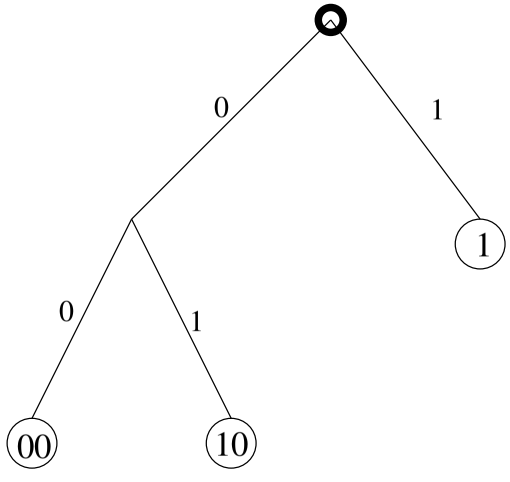
- Un **trie** est un arbre dont les arêtes sont étiquetées par des lettres.
- Aux noeuds d'un trie sont associés les mots obtenues en lisant les étiquettes *remontant* à la racine.
- Réciproquement, on peut associer un trie à tout ensemble de mots (comme ici $\{10, 0, 11\}$).

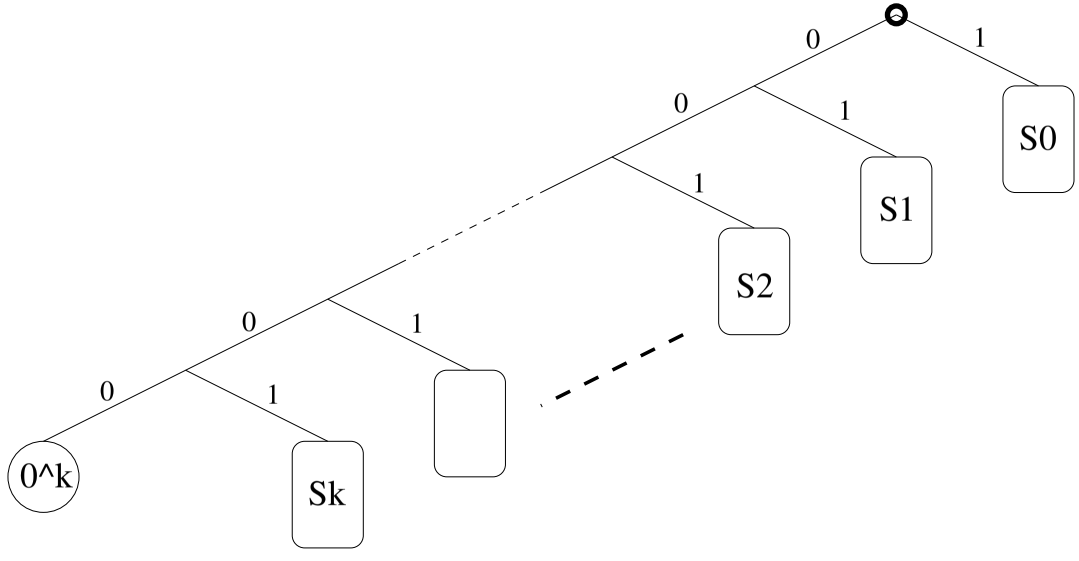
Mais il se peut que certains ne soient pas des feuilles !



Représentation des DCS comme trie

- A tout DCS $\mathcal{T}$ correspond un trie dont *les feuilles* sont associées les éléments de $\mathcal{T}$, et réciproquement.
- Exemples :





$$\mathcal{T} = \{00, 10, 1\} \qquad \mathcal{T} = \{0^k\} \cup \{10^j : 0 \leqslant j < k\}$$

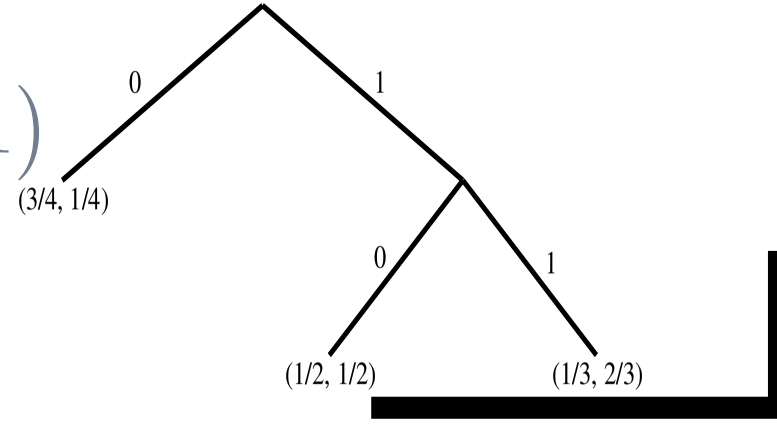
Sources à arbre de contexte

- Etant donné un DCS $\mathcal{T}$, appelé par la suite **arbre de contexte**, et $|\mathcal{T}|$ lois de probabilité sur A notées $p = (p(.|w))_{w\in\mathcal{T}}$, la **source à arbre de contexte** $\mathbb{P}_{\mathcal{T},p}$ est la loi stationnaire sur $A^{\mathbb{Z}}$ définie par

$$\mathbb{P}_{\mathcal{T},p}\left(X_1 = x_1 | X_{-\infty}^0 = x_{-\infty}^0\right) = p\left(x_1 | \mathcal{T}\left(x_{-\infty}^0\right)\right).$$

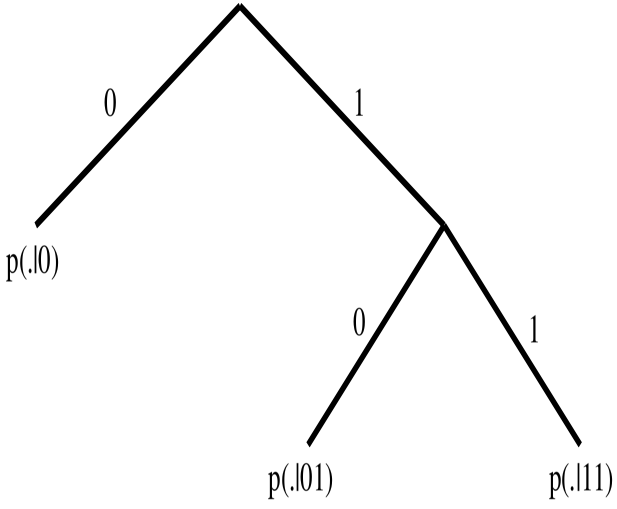
- Exemple :

$$\mathbb{P}\left(X_1^4 = 1001 | X_{-\infty}^0 = \ldots 01\right) = \frac{1}{2} \times \frac{1}{3} \times \frac{3}{4} \times \frac{1}{4}$$



Les CTS *finies* sont des chaînes de Markov

- La profondeur de l'arbre est alors l'ordre markovien.



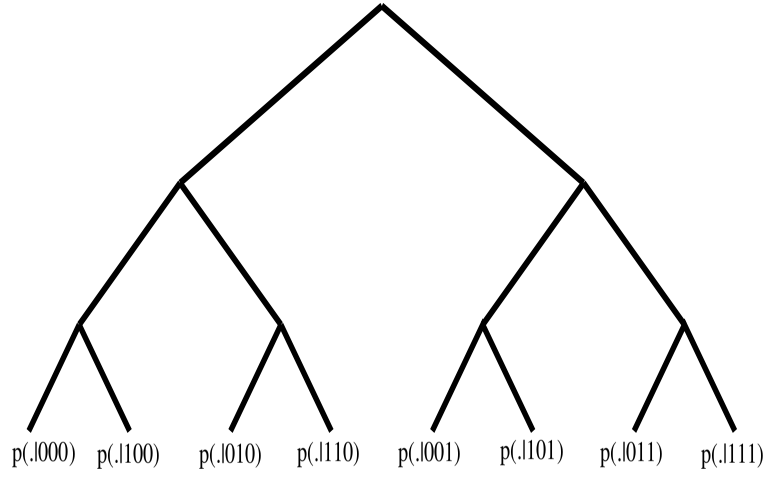
$$\rightarrow M = \begin{pmatrix} p(.|0) \\ p(.|0) \\ p(.|01) \\ p(.|11) \end{pmatrix}$$

- *Variable Length* Markov Chains : les CTS peuvent avoir moins de degrés de liberté pour une même taille de mémoire.

Les chaînes de Markov sont des CTS

- Le trie correspond est l'arbre plein de profondeur égal à l'ordre markovien :

$$M = \begin{pmatrix} p\left(.|000\right) \\ p\left(.|100\right) \\ \vdots \\ p\left(.|111\right) \end{pmatrix} \rightarrow$$



- $\Longrightarrow$ les CTS ont le pouvoir d'universalité des CDM et peuvent approcher toutes les sources stationnaires ergodiques.
- pas plus difficile à utiliser.

Expression de la vraisemblance

- $x = \bigodot_{s \in \mathcal{T}} \mathcal{T}(x, s) \rightarrow$ vraisemblance s'écrit :

$$\begin{aligned} P_{\mathcal{T},p}(x_1^n | x_{-\infty}^0) &= \prod_{i=1}^{n} p(x_i | \mathcal{T}(x_{-\infty}^{i-1})) \\ &= \prod_{s \in \mathcal{T}} p_s\left(\mathcal{T}(x, s)\right) \end{aligned}$$

- D'où l'expression du maximum de vraisemblance :

$$-\log \hat{P}_{\mathcal{T}}(x) = \sum_{s \in \mathcal{T}} H(\mathcal{T}(x, s))$$

Mélange de $\mathcal{KT}$ pour un modèle

- On définit

$$\mathcal{KT}_{\mathcal{T}}(x_1^n|x_{-\infty}^0) = \prod_{s\in\mathcal{T}} \mathcal{KT}\left(\mathcal{T}(x,s)\right)$$

- **Théorème** : il existe une constante C telle que :

$$-\log_2 \mathcal{KT}_{\mathcal{T}}(x_1^n|x_{-\infty}^{-1}) \leqslant \inf_{\theta\in\Theta^{\mathcal{T}}} -\log_2 \mathbb{P}_{\mathcal{T},\theta}(x_1^n|x_{-\infty}^{-1}) + |\mathcal{T}|\frac{|A|-1}{2}\log\left(\frac{n}{|T|}\right) + C\,|\mathcal{T}|$$

- Cette redondance (ponctuelle) est minimax dans ce modèle.

Plan de l'exposé

- Préliminaire
- Chaînes de Markov à ordre variable
- L'algorithme Context Tree Weighting
- Processus de Renouvellement
- Redondance de $\mathcal{CTW}$ sur les PR

Double mélange

- On présente le cas *binaire* pour simplifier
- Prior π sur les arbres : il y a $Catalan_s$ arbres à s feuilles d'où

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

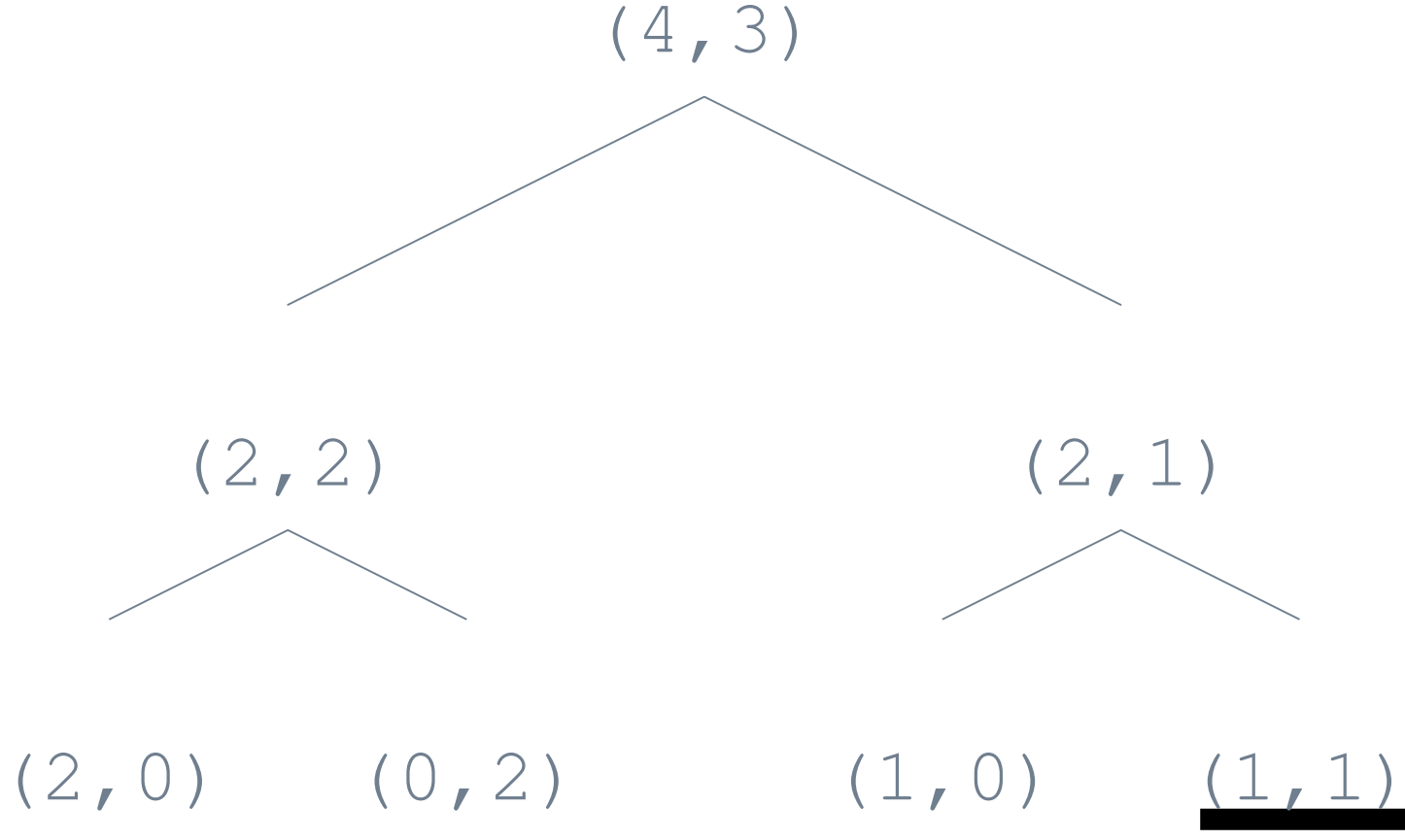
- On définit pour une collection $\mathbb{T}$ d'arbres

$$\mathcal{CTW}(x) = \sum_{\mathcal{T}\in\mathbb{T}} \mathcal{KT}_{\mathcal{T}}(x)\pi(\mathcal{T})$$

- C'est une loi de probabilité sur chaque A^n $\Longrightarrow$ codage arithmétique.

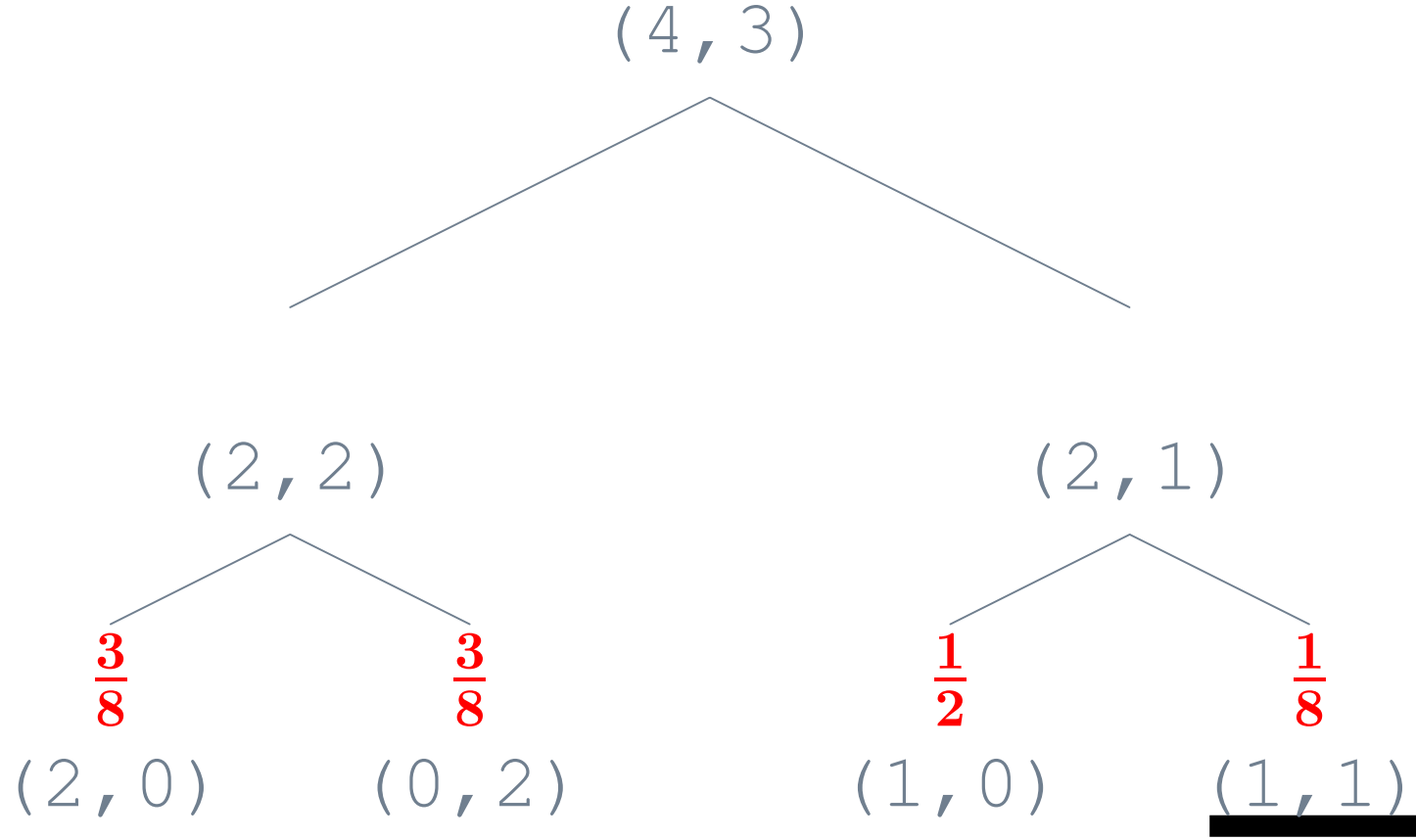
Cacul pratique de CTW

En chaque noeud, on fait la moyenne du coût propre et du sous-coût.



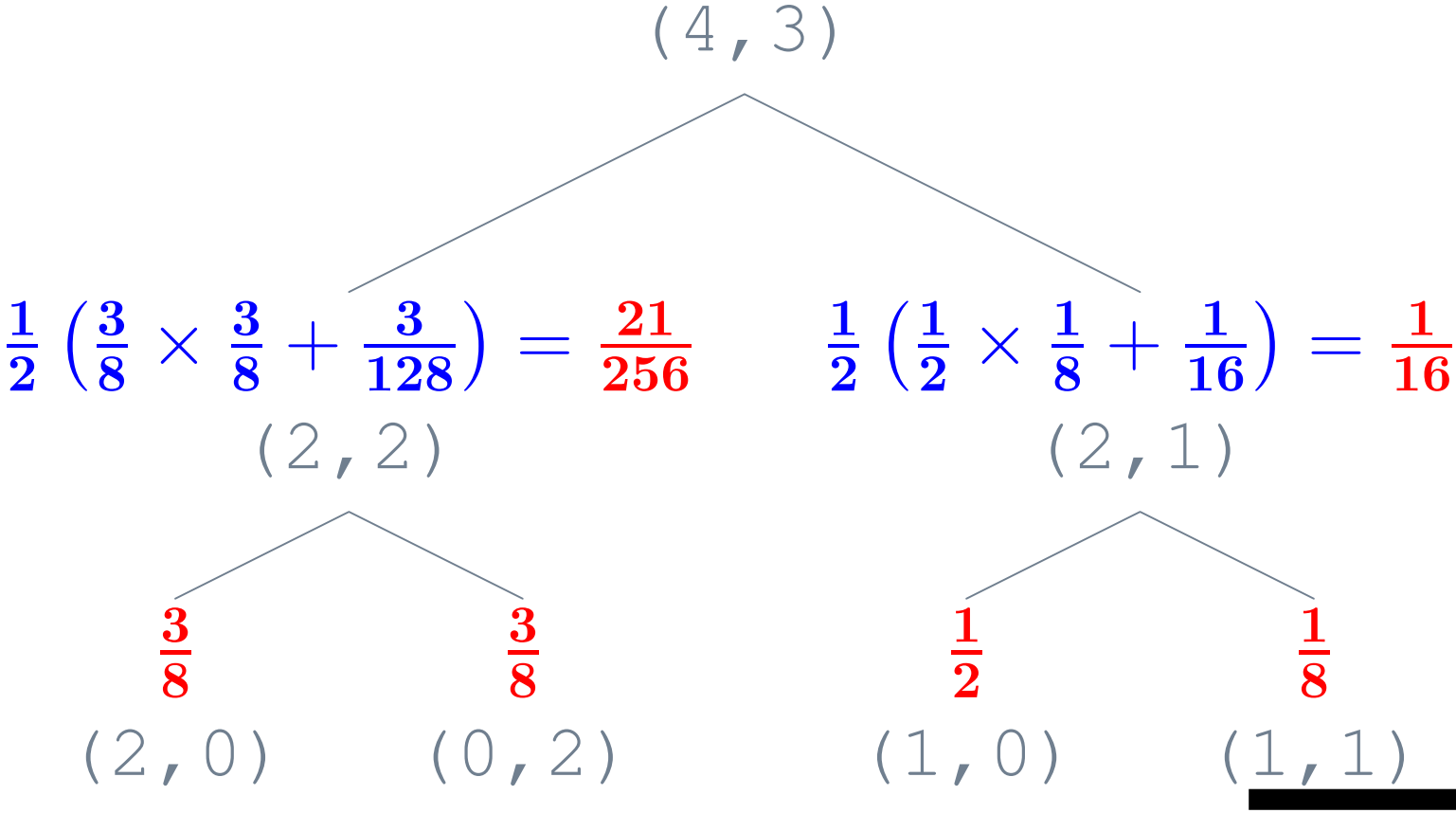
Cacul pratique de CTW

En chaque noeud, on fait la moyenne du coût propre et du sous-coût.



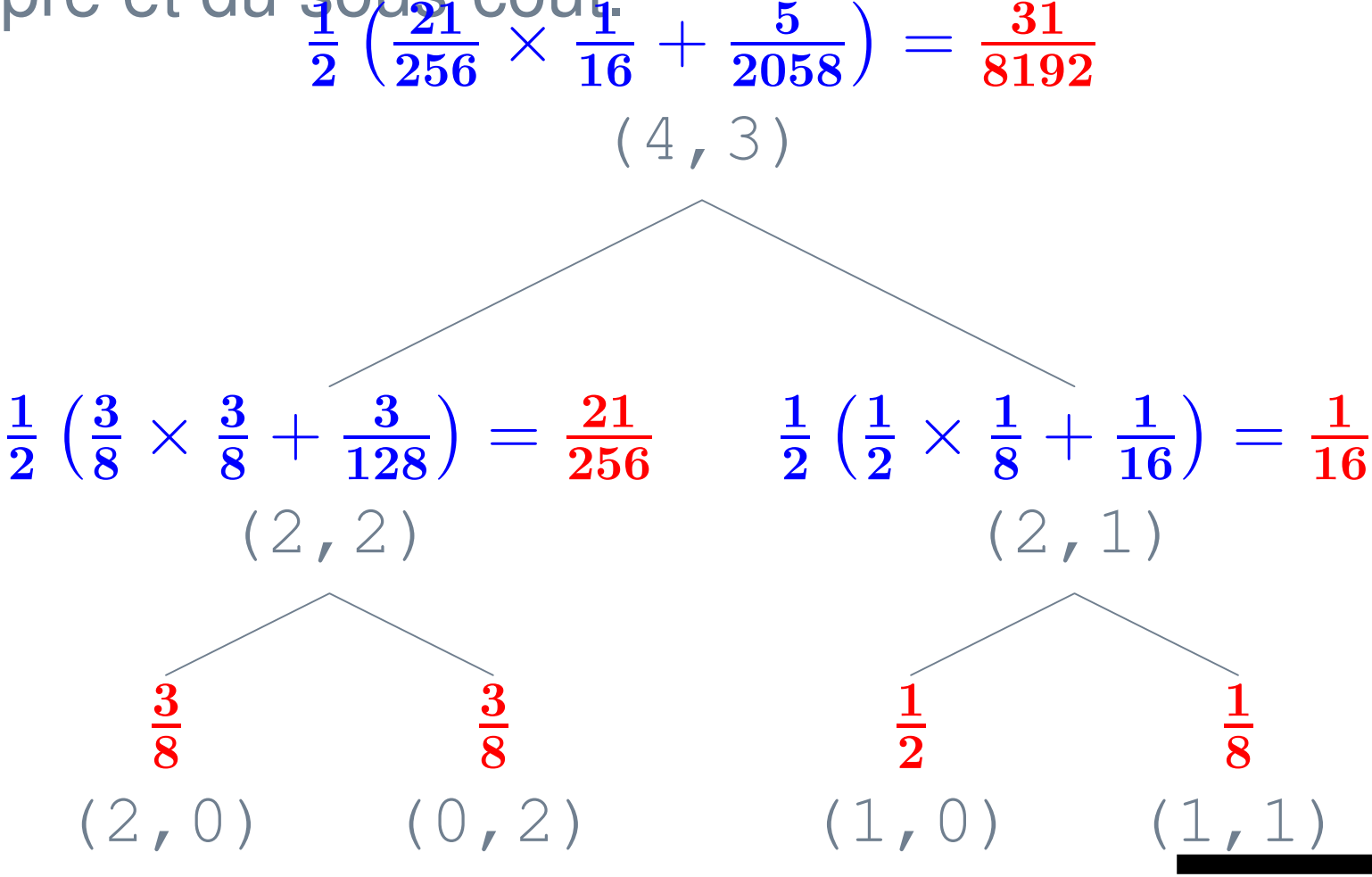
Cacul pratique de CTW

En chaque noeud, on fait la moyenne du coût propre et du sous-coût.



Cacul pratique de $\mathcal{CTW}$

En chaque noeud, on fait la moyenne du coût propre et du sous-coût.



Efficacité - optimalité

- **Théorème :** Pour tout $n \in \mathbb{N}$, pour tout $x \in A^n$:

$$-\log \mathcal{CTW}(x) \leqslant \inf_{\mathcal{T} \in \mathbb{T}} -\log \hat{P}_{\mathcal{T}}(x) + |\mathcal{T}| \log\left(\frac{n}{|\mathcal{T}|}\right) + 2|\mathcal{T}|$$

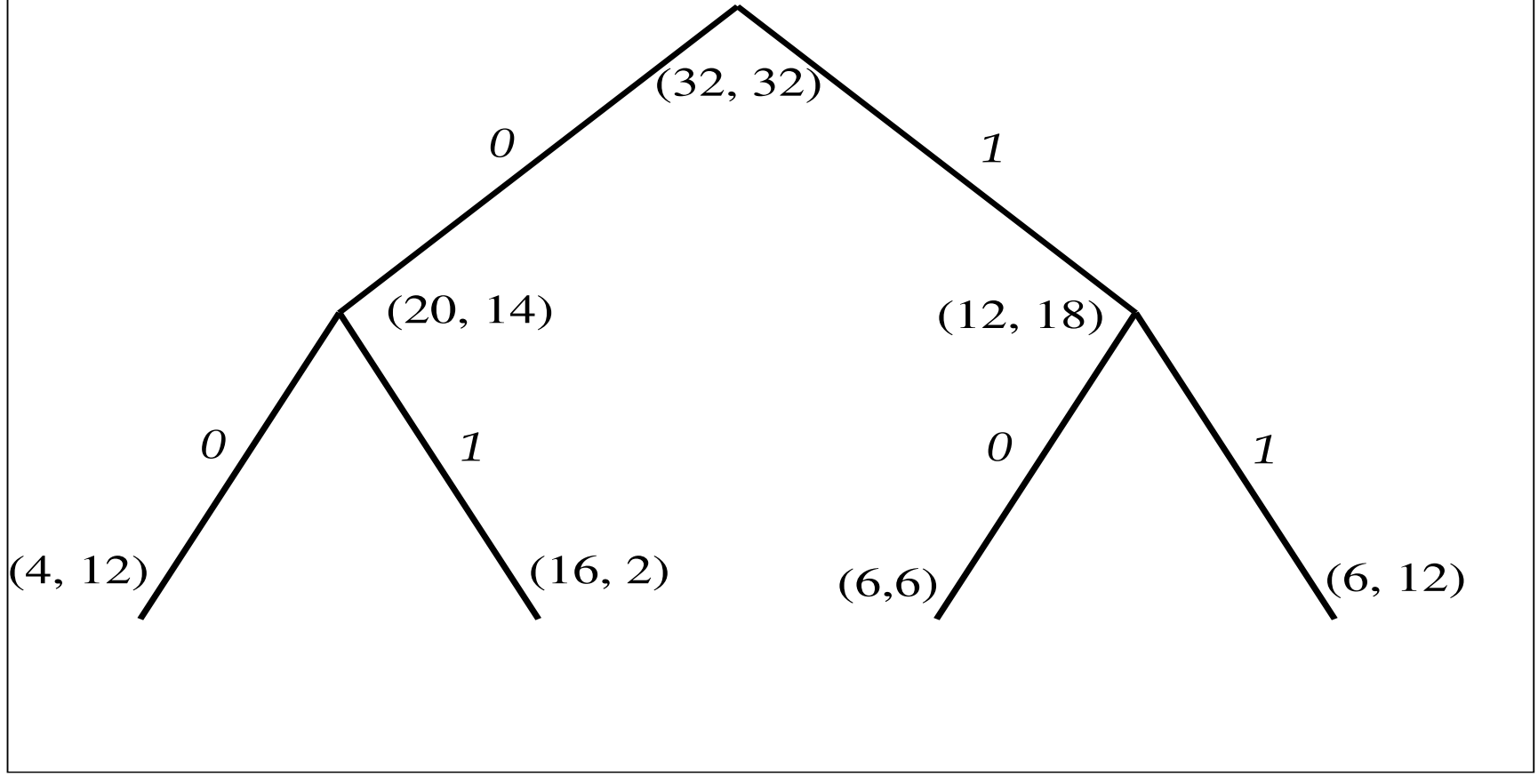
- Optimal au second ordre (minimax).
- Inégalité oracle non asymptotique.

La méthode CTM - Willems, Shtarkov, Tjalkens '95

- Idée : partir du *bas* de l'arbre
- Pour chaque noeud s calculer :
  - son **cout propre** $CP(s) = H\left(\mathcal{T}(x,s)\right)$
  - son **sous-coût** $SC(s) = \sum_{a\in A} MC(sa) + \frac{1}{2}\log n$
  - et son **meilleur cout** $MC(s) = \min CP(s), SC(s)$
- Trouver les **noeuds actifs** s tq $MC(s) = SC(s)$
- Choix final : partant de la racine, casser tous les noeuds actifs.

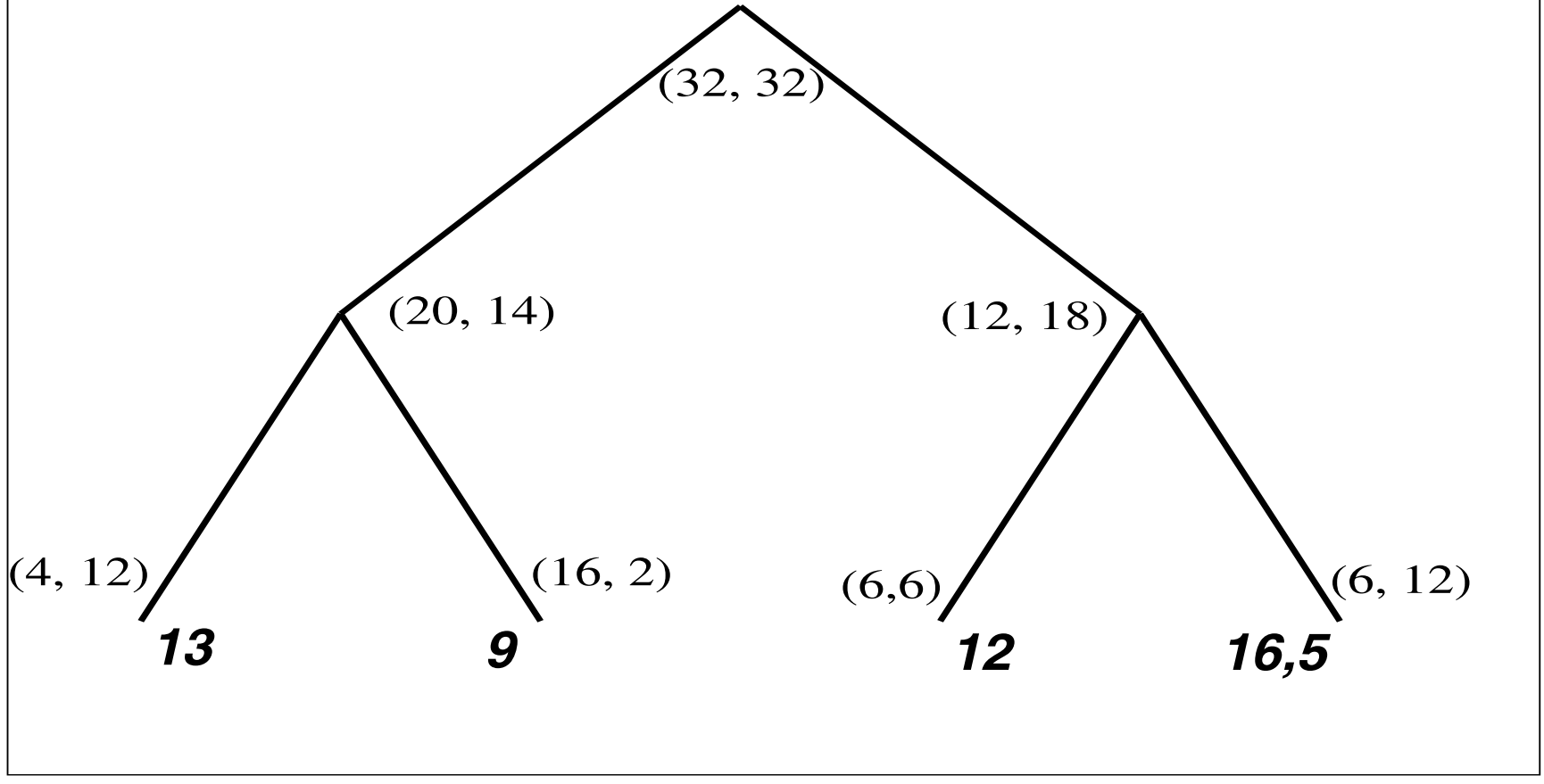
Exemple

$n = 64 \implies pen = \frac{1}{2} \log 64 = 3$



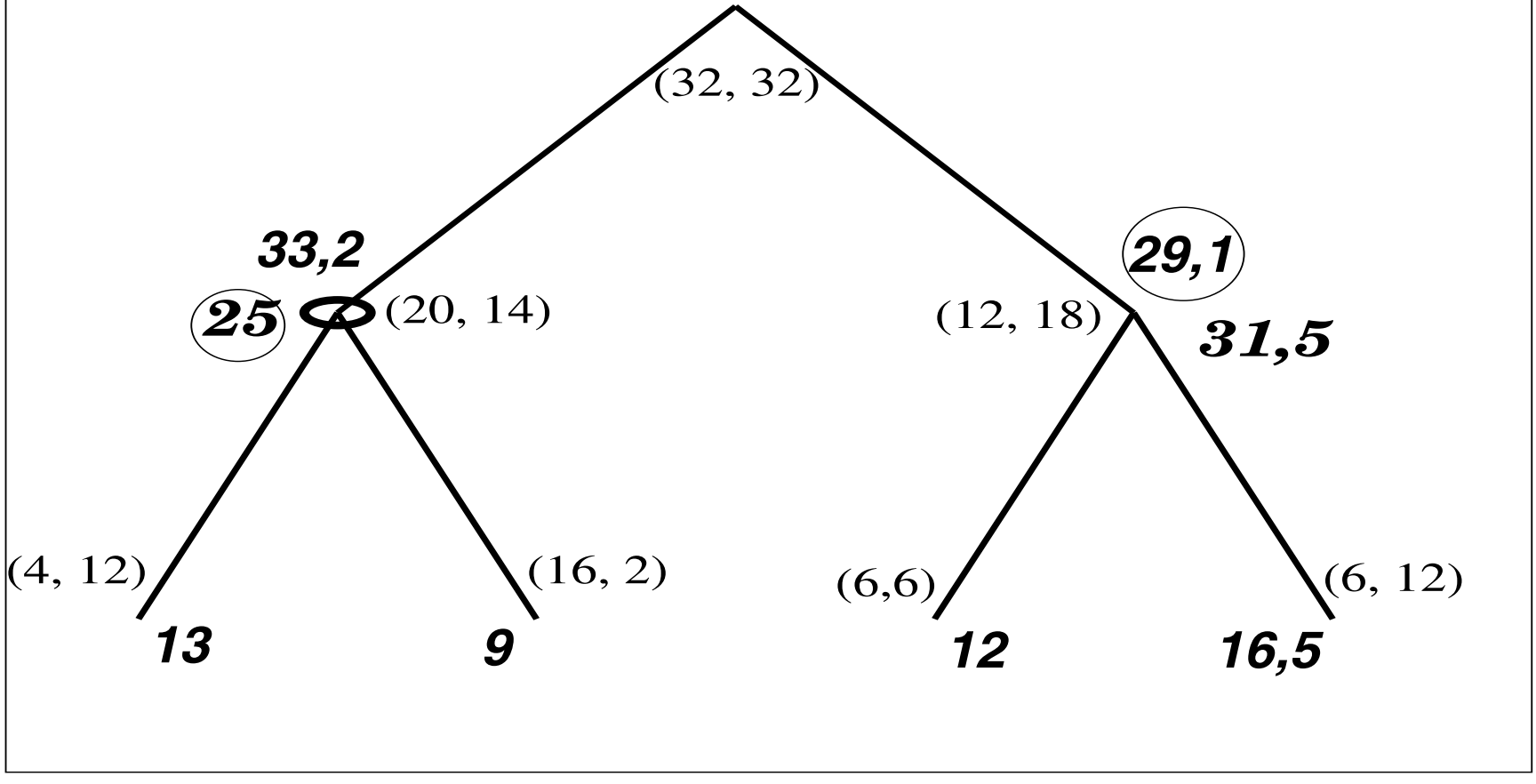
Exemple

$n = 64 \implies pen = \frac{1}{2} \log 64 = 3$



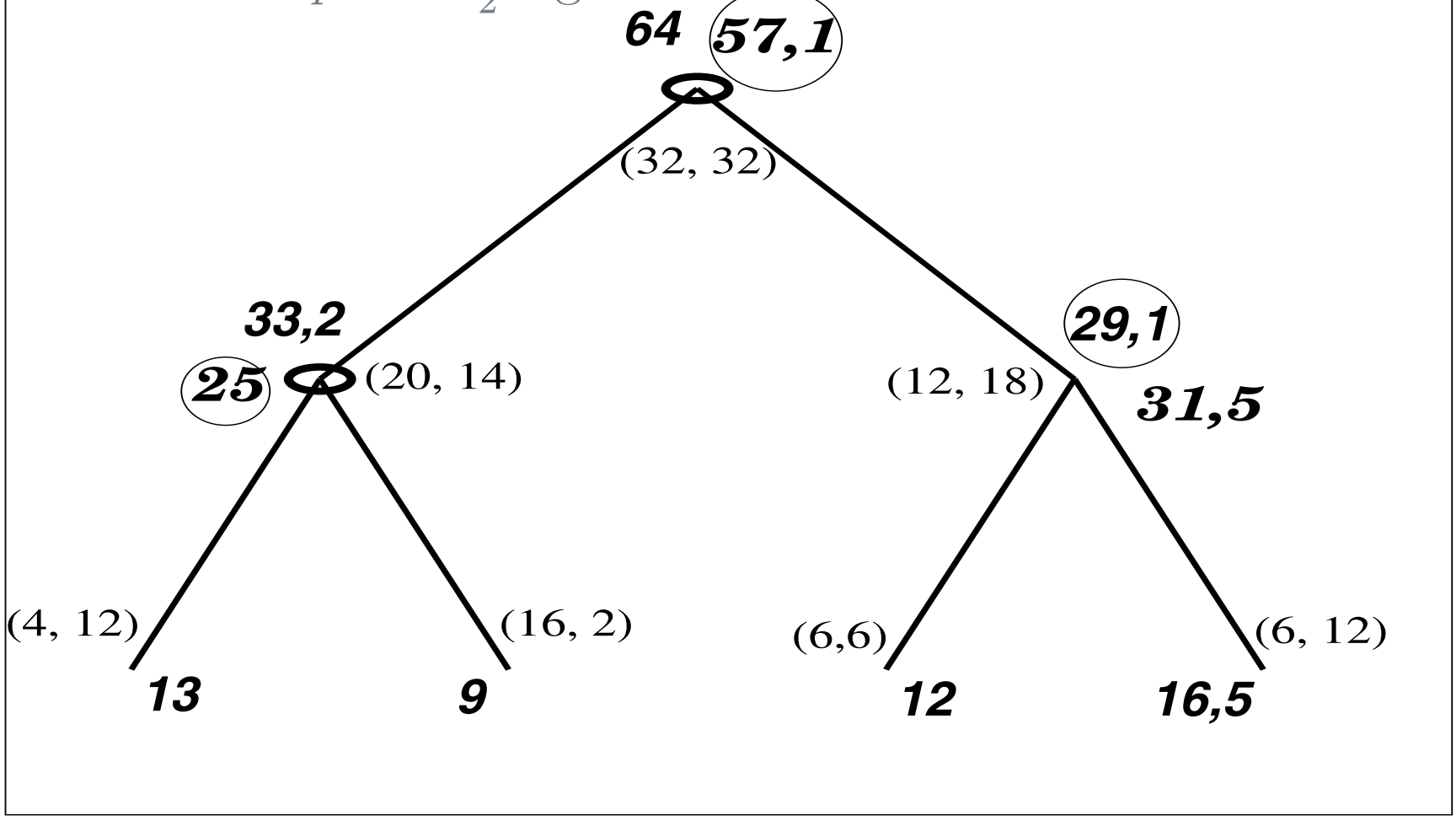
Exemple

$n = 64 \implies pen = \frac{1}{2} \log 64 = 3$



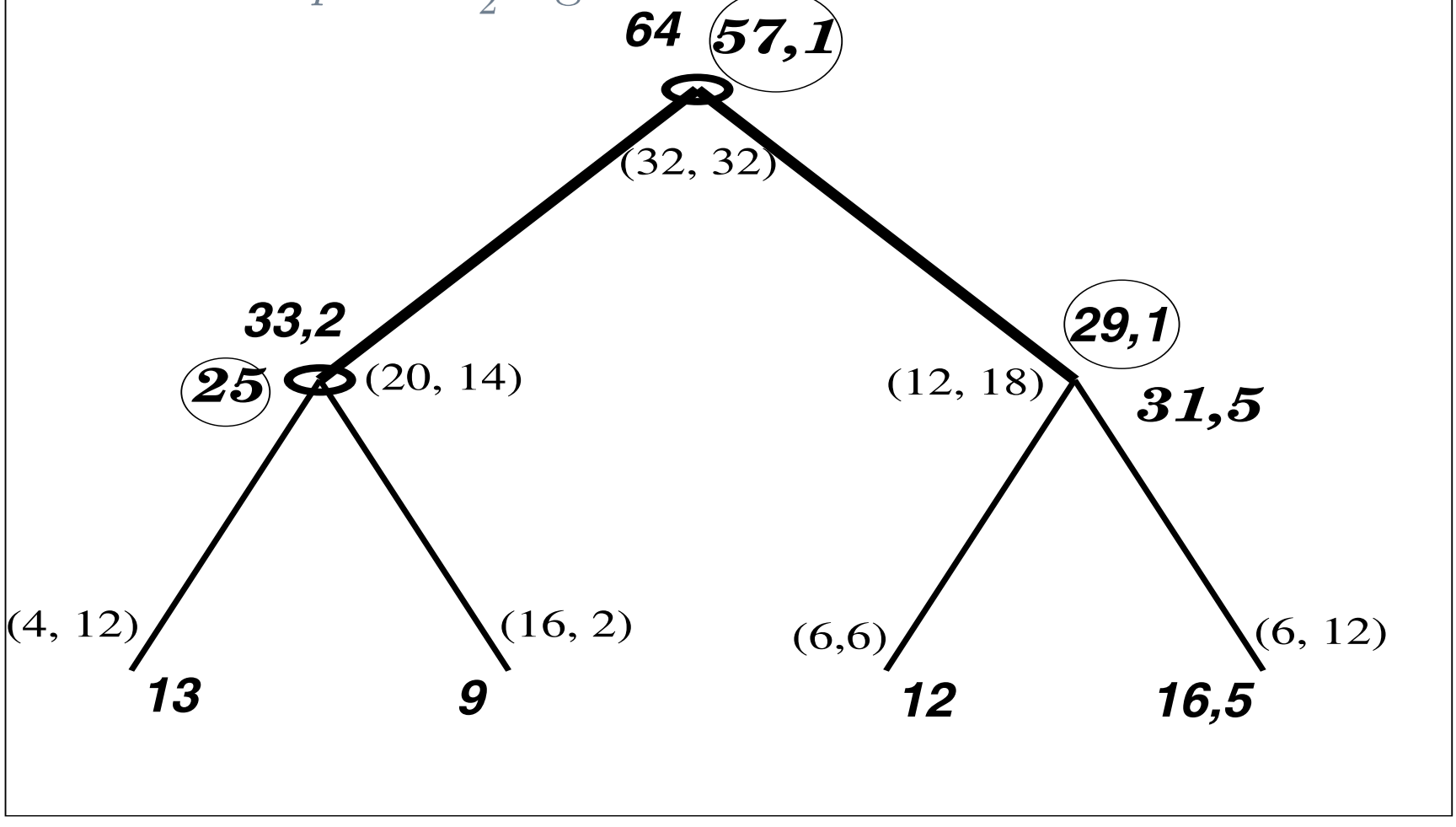
Exemple

$n = 64 \implies pen = \frac{1}{2} \log 64 = 3$



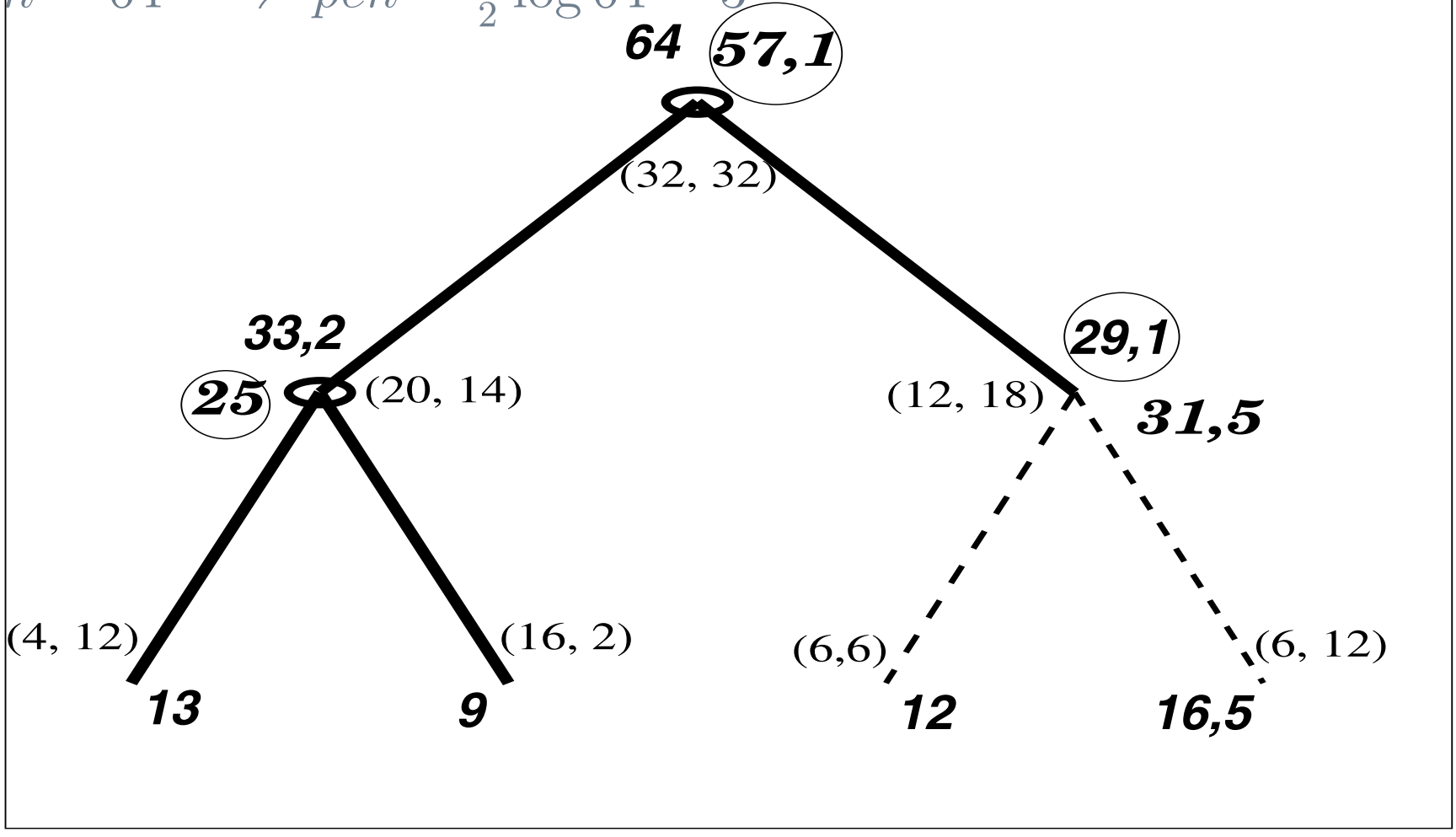
Exemple

$n = 64 \implies pen = \frac{1}{2} \log 64 = 3$



Exemple

$n = 64 \implies pen = \frac{1}{2}\log 64 = 3$

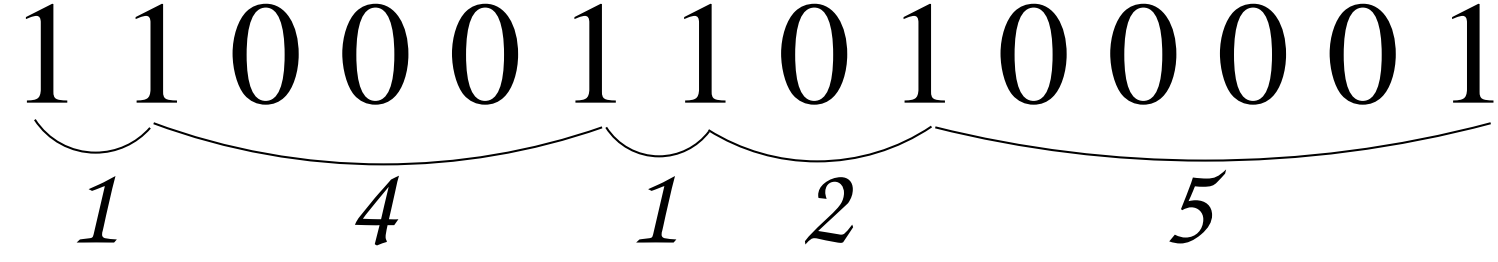


Plan de l'exposé

- Préliminaire
- Chaînes de Markov à ordre variable
- L'algorithme Context Tree Weighting
- Processus de Renouvellement
- Redondance de $\mathcal{CTW}$ sur les PR

Définition

- X est un **processus de renouvellement** s'il est à valeur dans $A = \{0, 1\}$ et si les distances entre deux '1' successifs de X sont des variables aléatoires indépendantes même loi sur $\mathbb{N}^*$.



- On définit de même les **processus de renouvellement markovien** à l'aide d'un noyau markovien sur $\mathbb{N}^*$.

Propriétés

- Un processus iid où chaque symbole suit une $\mathcal{B}(p)$ est un PR de loi géométrique $\mathcal{G}(p)$;
- Si sa loi de renouvellement est bornée, le PR est une chaînes de Markov.
- Si $x_1^n = 0^{t_0-1}1\ 0^{t_1-1}1\ 0^{t_2-1}1\ \ldots\ 0^{t_N-1}1\ 0^{t_{N+1}-1}$, et si la loi de renouvellement est Q, alors en posant $R_Q(t) = \sum_{u=t}^{\infty} Q(u)$ on a :

$$\mathbb{P}_Q^{\mathcal{R}}(x) = \left(\frac{1}{\mathbb{E}[Q]} R_Q(t_0)\right) \prod_{i=1}^{N} Q(t_i)\ R_Q\left(t_{N+1}\right).$$

Redondance minimax [Csiszár–Shields '96]

Theorème : Dans la classe $\mathcal{R}$ des processus de renouvellement, il existe deux constants positives c et C telles que $\forall n \in \mathbb{N}$:

$$c\sqrt{n} \leqslant \bar{R}_n(\mathcal{R}) \leqslant R_n^*(\mathcal{R}) \leqslant C\sqrt{n}.$$

Théorème : Dans les classe $\mathcal{MR}$ des processus de renouvellement markoviens, il existe des constantes c et C telles que $\forall n \in \mathbb{N}$:

$$cn^{2/3} \leqslant \bar{R}_n(\mathcal{MR}) \leqslant R_n^*(\mathcal{MR}) \leqslant Cn^{2/3}.$$

Remarques

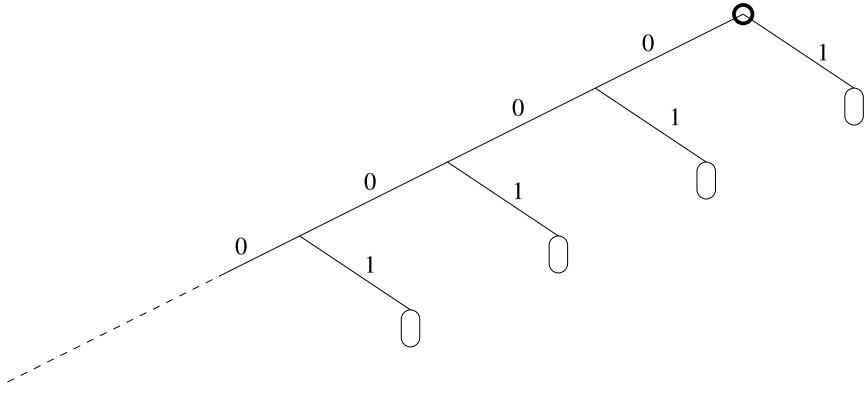
- On peut généraliser le résultat aux chaînes de Markov d'ordre quelconque.
- La preuve utilise le code de Shtarkov (NML) :
  - inutilisable : pas de calcul pratique ;
  - pas "adaptatif" : sur les PR bornés, la redondance reste $\Theta(\sqrt{n})$.
- PR = premier exemple de classe de "complexité intermédiaire", c'est-à-dire entre $\log n$ et n.

Plan de l'exposé

- Préliminaire
- Chaînes de Markov à ordre variable
- L'algorithme Context Tree Weighting
- Processus de Renouvellement
- Redondance de $\mathcal{CTW}$ sur les PR

Les P.R. commes Context Trees

- On peut visualiser les P.R. comme des sources à arbres de contexte infini



- Les probabilités conditionnelles sont données par :

$$\mathbb{P}_Q\left(1|10^{j-1}\right) = \frac{Q(j)}{R(j)}.$$

Redondance de $\mathcal{CTW}$ [Garivier '04]

- **Théorème**: Il existe des constantes positives c et C telles que pour tous les arbres de contexte $\mathcal{T}$ et tout $n \in \mathbb{N}$:

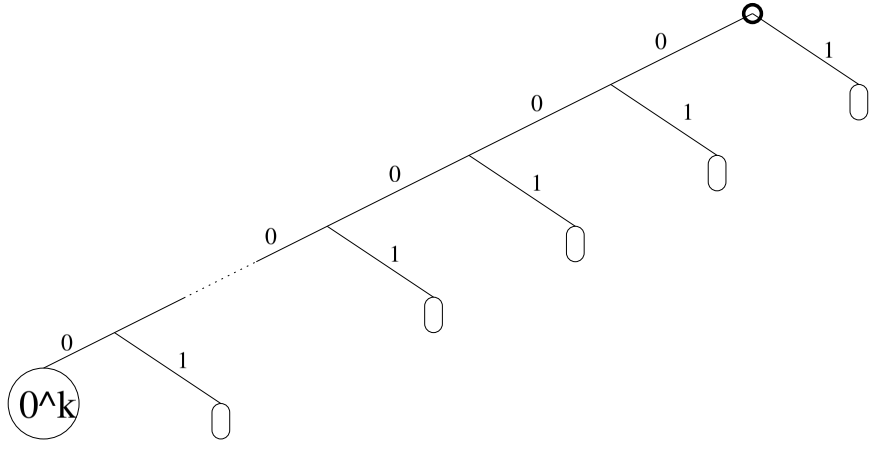
$$c\sqrt{n}\log n \leqslant R_n^*(\mathcal{CTW}, P.R.) \leqslant C\sqrt{n}\log n$$

- **Théorème**: Il existe des constantes positives C_1 et C_2 telles que pour tous les arbres de contexte $\mathcal{T}$ et tout $n \in \mathbb{N}$:

$$c\sqrt{n}\log n \leqslant R_n^*(\mathcal{CTW}, P.R.) \leqslant C\sqrt{n}\log n$$

Preuve de la BS

- On prend l'arbre $\mathcal{T}$ comme sur le graphe avec $k = \sqrt{n}$.



- $-\log \mathcal{CTW}(x) \geqslant -\log \mathcal{KT}_T(x) + 2k + 1$
- $-\log \mathcal{KT}(x) \leqslant -\log \hat{P}_{\mathcal{T}}(x) + \frac{2k+1}{2} \log n + 3k + 1.$
- $-\log \hat{P}_{\mathcal{T}}(x) \leqslant -\log \mathbb{P}_Q(x) - \log \mathcal{KT}(\mathcal{T}(0^k, x))$.
- $\mathcal{T}(0^k, x)$ contient au plus $n/k = \sqrt{n}$ symboles '1', donc se code avec moins de $\sqrt{n} \log n$ bits.

Implications

- La preuve montre aussi que si on se limite aux *petits modèles*, la redondance ponctuelle maximale est importante.
- $\mathcal{CTW}$ est donc presque *adaptatif* dans cette classe de processus de complexité intermédiaire, à très longue mémoire.
- Note : On utilise vraiment des arbres profonds et très déséquilibrés : c'est toute la *souplesse* des modèles à arbres de contexte.
- Ce résultat suggère que $\mathcal{CTW}$ est adapté pour le traitement des processus plus irréguliers que des cdm, à longue mémoire.

Micro-bibliographie

- **Universal modeling and coding** - Rissanen, Langdon, IEEE-IT 1981
- **Universal coding, Information, Prediction, and Estimation** -Rissanen, IEEE-IT 1984
- **The Context-Tree Weighting Method : basic Properties** - Willems, Shtarkov, Tjalkens IEEE-IT 1995
- **Redundancy Rates for Renewal and Other Processes** - Csiszár, Shields - IEEE-IT 1996
- **Variable length Markov chains** - Bühlmann, Wyner, Abraham, Annals of Statistics 1999
- **Context Tree Estimation for Not Necessarily Finite Memory Processes, via BIC and MDL.** - Csiszár, Talata (Budapest), IEEE-IT 2004