

1. Introduction

Nous considérons un treillis (ou pavage) dont les cellules sont colorées avec quatre couleurs, correspondant aux quatre combinaisons de booléens $(0, 0)$, $(1, 0)$, $(0, 1)$ et $(1, 1)$. La règle de pavage locale est la suivante : étant données deux cellules adjacentes sur une même colonne, représentées par (x_1, y_1) (pour la cellule du dessus) et (x_2, y_2) (pour celle du dessous), la cellule située à droite de la cellule du dessus prend la couleur (x_1, y_2) . Cette règle permet de visualiser les sommes de deux nombres premiers comme des intersections de booléens positionnés sur deux axes.

Le problème est que ce treillis a besoin en entrée de la suite des caractères de primalité (sur l'axe des abscisses et sur l'axe des ordonnées (ou sur la diagonale principale)) pour fonctionner. On nous a suggéré d'utiliser des chaînes de Markov. Nous allons expliquer comment cette modélisation peut se faire, et montrer les limites inhérentes à cette approche.

2. Rappels sur les chaînes de Markov

Une *chaîne de Markov* (à temps discret, à espace d'états fini ou dénombrable) est un processus stochastique $(X_n)_{n \geq 0}$ tel que la probabilité de transition d'un état à l'autre ne dépend que de l'état présent, et non du passé :

$$\mathbb{P}(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = \mathbb{P}(X_{n+1} = j \mid X_n = i) =: P_{i,j}.$$

La mémoire du processus est donc bornée : elle ne dépasse pas un pas.

On peut généraliser aux chaînes d'ordre k , où l'état futur dépend des k derniers états, mais la mémoire reste finie.

3. Deux types de modélisation

3.1. Modélisation du pavage lui-même

Considérons les colonnes du pavage comme des suites infinies de couleurs. La règle locale

$$(x_1, y_1)(x_2, y_2) \longmapsto (x_1, y_2)$$

permet de calculer la colonne suivant une certaine colonne. En effet, les éléments de la colonne C_{n+1} sont entièrement déterminés par la règle. On peut donc définir une *transition déterministe* sur l'espace des paires de colonnes :

$$T : C_n \mapsto C_{n+1}.$$

Ainsi, l'ensemble des pavages valides est un *sous-shift* (ou système dynamique symbolique) dont les transitions sont déterministes. Ce n'est pas une chaîne de Markov stochastique, mais un système

déterministe à mémoire 1.

Si l'on souhaite introduire du hasard, on peut considérer que l'état des booléens des axes de coordonnées est tiré selon une loi de probabilité, puis que la transition est déterministe. On obtient alors un processus aléatoire dont la distribution est une mesure de probabilité invariante par le shift, et on peut l'étudier comme une *chaîne de Markov déterministe* (où les probabilités de transition sont 0 ou 1).

3.2. Modélisation de l'entrée (la suite de primalité)

L'entrée du treillis est la suite caractéristique des nombres premiers, notée $(p_n)_{n \geq 0}$ avec $p_n = 0$ si n est premier et 1 sinon. Cette suite est déterministe, mais on pourrait chercher à la remplacer par un processus aléatoire markovien pour des simulations ou des approximations.

Il est connu que la suite des nombres premiers a une *complexité de sous-mots* qui croît comme $\frac{n}{\log n}$ (environ). Une chaîne de Markov à mémoire finie ne peut produire qu'une complexité de sous-mots au plus linéaire (ou exponentielle avec une base constante). Par conséquent, il est impossible de générer exactement la suite des primalités par une chaîne de Markov à mémoire finie. C'est le même obstacle que pour les automates finis : la mémoire nécessaire pour reconnaître ou produire la suite des nombres premiers est illimitée.

4. Conséquences pour le treillis

Le treillis hérite de la complexité de son entrée. Si l'on injecte la vraie suite de primalité, le pavage produit est unique et d'une complexité également élevée. En revanche, si l'on utilise une approximation markovienne (par exemple une suite de Bernoulli avec $p = \frac{1}{\log n}$ variant lentement), on obtient un pavage aléatoire dont on peut étudier les propriétés statistiques (espérance du nombre de cellules $(0, 0)$, corrélations, etc.), mais ce n'est plus le pavage exact.

5. Pistes pour une modélisation probabiliste

On peut envisager les approches suivantes :

1. **modèle de percolation** : remplacer la suite des primalités par des booléens indépendants et identiquement distribués de probabilité $p = \frac{1}{\log n}$ (densité locale des nombres premiers). Cela donne un pavage aléatoire dont les propriétés peuvent être étudiées par des méthodes de percolation.
2. **processus de renouvellement** : les écarts entre nombres premiers suivent des lois statistiques (par exemple la conjecture de Cramér). On peut simuler une suite de caractères de primalité en tirant les écarts entre nombres successifs selon une loi appropriée (exponentielle de paramètre variable). Ce processus a une mémoire finie (car l'état est le temps écoulé depuis le dernier nombre premier), mais il n'est pas stationnaire.

3. **chaînes de Markov d'ordre supérieur** : on peut tenter d'approcher la suite par une chaîne de Markov d'ordre k avec k grand, mais cela ne donne qu'une approximation et ne capture pas la vraie complexité.

6. Conclusion

En résumé :

- le pavage lui-même peut être vu comme une *chaîne de Markov déterministe* (transition 0 ou 1) sur l'espace des colonnes.
- l'entrée (suite de primalité) ne peut pas être exactement modélisée par une chaîne de Markov à mémoire finie, en raison de sa complexité.
- pour des simulations ou des études statistiques, on peut utiliser des modèles probabilistes approchés (percolation, renouvellement, etc.).

Cette analyse montre que la modélisation par chaînes de Markov est pertinente pour le pavage lui-même, mais rencontre les mêmes limites que les automates finis pour l'entrée.