

# Mondrian Forests

**HMMA 308 : Apprentissage Statistique**

**KANDOUCI Walid**

[https:](https://github.com/WalidKandouci/HMMA308--Mondrian-Forests)

[//github.com/WalidKandouci/HMMA308--Mondrian-Forests](https://github.com/WalidKandouci/HMMA308--Mondrian-Forests)

Université de Montpellier



# Sommaire

Introduction

Arbres de Mondiran

Application

Conclusion

# Introduction

## Rappel forêts aléatoires:

- ▶ Algorithme bagging
- ▶ Bon résultat sur de vrai données
- ▶ simples à mettre en œuvre

# Introduction

## Forêts Mondrian:

- ▶ Nouvelle classe des forêts aléatoires
- ▶ Efficaces
- ▶ Offre meilleure estimation d'incertitude que les forêts aléatoires



# Arbres de Mondrian

## L'approche:

- ▶ chaque noeud  $j$  a exactement un noeud parent, sauf pour un noeud racine distingué  $\epsilon$  qui n'a pas de parents
- ▶ chaque noeud  $j$  est le parent d'exactlyement zéro ou deux noeuds enfants : (le noeud de gauche " $left(j)$ " et le noeud de droite " $right(j)$ ")

# Arbres de Mondrian

## L'algorithm:

---

**Algorithm 1** SampleMondrianTree( $\lambda, \mathcal{D}_{1:n}$ )

---

**Initialisation:**  $T = \emptyset$ ,  $\text{leaves}(T) = \emptyset$ ,  $\delta = \emptyset$ ,  $\xi = \emptyset$ ,  $\tau = \emptyset$ ,  $N(\epsilon) = \{1, 2, \dots, n\}$

SampleMondrianBlock( $\epsilon, \mathcal{D}_{N(\epsilon)}, \lambda$ )

---

**Algorithm 2** SampleMondrianBlock( $j, \mathcal{D}_{N(j)}, \lambda$ )

---

Add  $j$  to  $T$

For all  $d$ , set  $\ell_{jd}^x = \min(\mathbf{X}_{N(j),d})$ ,  $u_{jd}^x = \max(\mathbf{X}_{N(j),d})$

Sample  $E$  from exponential distribution with rate  $\sum_d (u_{jd}^x - \ell_{jd}^x)$

**if**  $\tau_{\text{parent}(j)} + E < \lambda$  **then**

    Set  $\tau_j = \tau_{\text{parent}(j)} + E$

    Sample split dimension  $\delta_j$ , choosing  $d$  with probability proportional to  $u_{jd}^x - \ell_{jd}^x$

    Sample split location  $\xi_j$  uniformly from interval  $[\ell_{j\delta_j}^x, u_{j\delta_j}^x]$

    Set  $N(\text{left}(j)) = \{n \in N(j) : \mathbf{X}_{n,\delta_j} \leq \xi_j\}$  and  $N(\text{right}(j)) = \{n \in N(j) : \mathbf{X}_{n,\delta_j} > \xi_j\}$

    SampleMondrianBlock ( $\text{left}(j), \mathcal{D}_{N(\text{left}(j))}, \lambda$ )

    SampleMondrianBlock ( $\text{right}(j), \mathcal{D}_{N(\text{right}(j))}, \lambda$ )

**else** Set  $\tau_j = \lambda$  and add  $j$  to leaves ( $T$ )

# Arbres de Mondrian

Les arbres de Mondrian diffèrent des arbres de décision comme suit:

- ▶ Les divisions sont échantillonnées indépendamment des  $Y_{N(j)}$
- ▶ Chaque noeud  $j$  est associé avec un temps de division  $\tau_j$
- ▶  $\lambda$  contrôle le nombre totale des divisions
- ▶ la division représenté par un noeud interne  $j$  ne tient que dans  $B_j^x$  et non pas  $B_j$

# Application

Fonction "random\_mondrian"

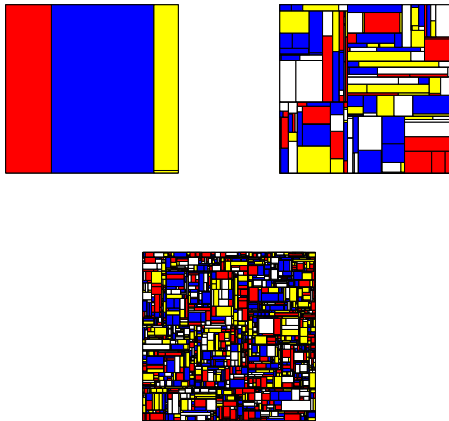


Figure: Exemple générateur aléatoire Mondrian (budget=2,10,50)



# Conclusion

- ▶ Forêts de Mondrian = Processus de Mondrian + Forêts aléatoires
- ▶ Peut fonctionner en mode batch ou en mode en ligne
- ▶ Meilleure estimation d'incertitude que les forêts aléatoires