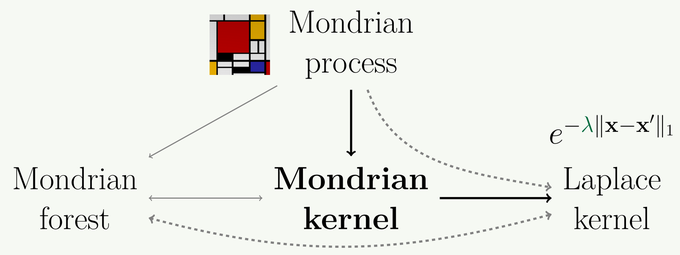
The Mondrian Kernel

Matej Balog [1,3,5] Balaji Lakshminarayanan [2] Zoubin Ghahramani [3] Daniel M. Roy [4] Yee Whye Teh [5]

[1]MPI for Intelligent Systems, Tübingen [2]Gatsby Unit, UCL [3]University of Cambridge [4]University of Toronto [5]University of Oxford

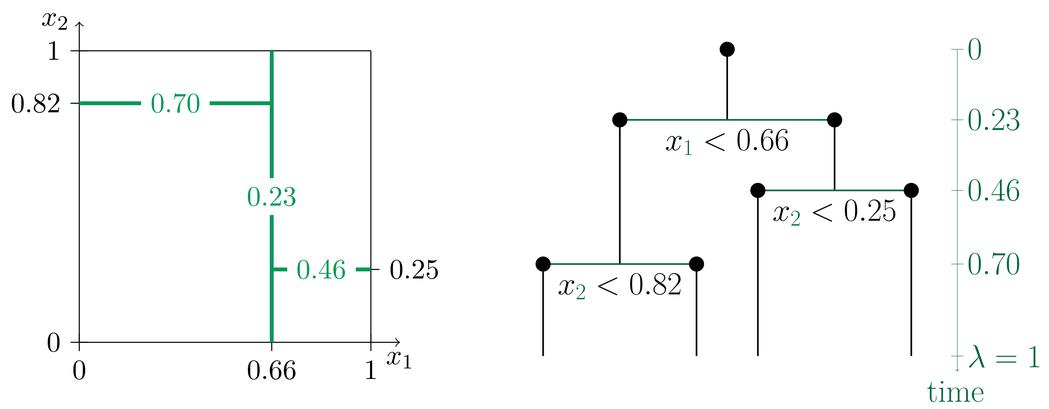
Contributions

1. Mondrian process ↔ Laplace kernel connection
2. fast learning of Laplace kernel width from data
3. a kernel ↔ random forest connection



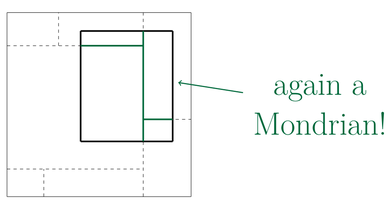
Mondrian process

is a guillotine-partition, randomly refined as time progresses, until a **lifetime** λ. [1, 2]



Main properties of the Mondrian process:

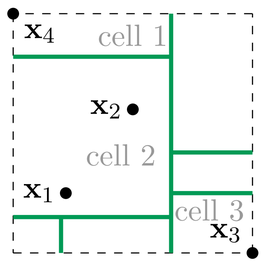
- A box with sum of lengths L splits after $\sim \text{Exp}(L)$ random time.
- **Projectivity**: restriction to a sub-box is again a Mondrian process.



Mondrian kernel

Randomized feature map ϕ:

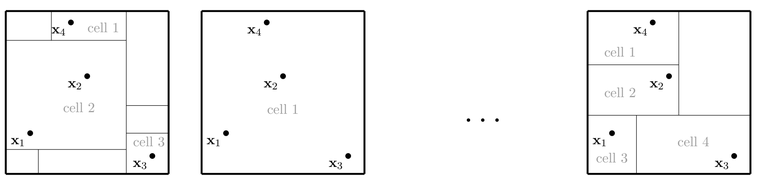
1. sample a Mondrian partition with lifetime λ,
2. label non-empty partition cells $1, 2, \ldots,$
3. encode a data point $\mathbf{x}$ in cell c as an indicator vector $\phi(\mathbf{x})$ with a 1 in position c.



$\mathbf{x}$	$\phi(\mathbf{x})$
$\mathbf{x}_1$	[0 1 0]
$\mathbf{x}_2$	[0 1 0]
$\mathbf{x}_3$	[0 0 1]
$\mathbf{x}_4$	[1 0 0]

$$k_1(\mathbf{x}, \mathbf{x}') := \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle = \begin{cases} 1 & \text{if } \mathbf{x}, \mathbf{x}' \text{ in the same cell} \\ 0 & \text{otherwise} \end{cases}$$

Repeat M times, concatenate, and normalize feature vectors:



$$\begin{aligned}
\phi(\mathbf{x}_1) &= \tfrac{1}{\sqrt{M}}\big[\quad 0\,1\,0 \quad 1 \quad \cdots \quad 0\,0\,1\,0 \quad\big] \\
\phi(\mathbf{x}_2) &= \tfrac{1}{\sqrt{M}}\big[\quad 0\,1\,0 \quad 1 \quad \cdots \quad 0\,1\,0\,0 \quad\big] \\
\phi(\mathbf{x}_3) &= \tfrac{1}{\sqrt{M}}\big[\quad 0\,0\,1 \quad 1 \quad \cdots \quad 0\,0\,0\,1 \quad\big] \\
\phi(\mathbf{x}_4) &= \tfrac{1}{\sqrt{M}}\big[\quad 1\,0\,0 \quad 1 \quad \cdots \quad 1\,0\,0\,0 \quad\big]
\end{aligned}$$

Mondrian kernel of order M:

$$k_M(\mathbf{x}, \mathbf{x}') := \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle = \frac{1}{M} \sum_{m=1}^{M} \mathbb{1}_{\{\mathbf{x},\mathbf{x}' \text{ in same cell of } m\text{-th partition}\}}$$

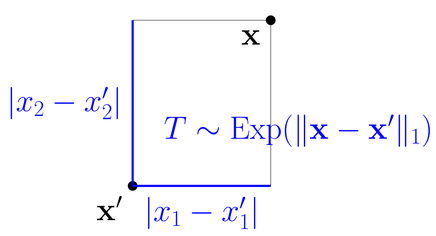
Mondrian-Laplace link

Theorem The Mondrian kernel of order ∞ is a.s. the Laplace kernel.

Proof. By the strong law of large numbers, a.s.,

$$\begin{aligned}
k_\infty(\mathbf{x}, \mathbf{x}') &:= \lim_{M \to \infty} k_M(\mathbf{x}, \mathbf{x}') \\
&= \mathbb{P}\left[\mathbf{x}, \mathbf{x}' \text{ in the same cell}\right] \\
&= \mathbb{P}\left[\text{Exp}\left(\|\mathbf{x} - \mathbf{x}'\|_1\right) > \lambda\right] \\
&= \exp\left(-\lambda \|\mathbf{x} - \mathbf{x}'\|_1\right)
\end{aligned}$$

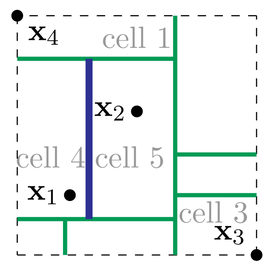
□



Note: *Mondrian lifetime λ ↔ inverse width (length-scale) of the Laplace kernel.*

Fast kernel width learning

Feature space reusal
Running the Mondrian process up to a terminal lifetime Λ, the generated features ϕ approximate all Laplace kernels with inverse kernel widths $\lambda \in [0, \Lambda]$.



$\mathbf{x}$	$\phi(\mathbf{x})$
$\mathbf{x}_1$	[0 *1* 0 **1 0**]
$\mathbf{x}_2$	[0 *1* 0 **0 1**]
$\mathbf{x}_3$	[0 *0* 1 **0 0**]
$\mathbf{x}_4$	[1 *0* 0 **0 0**]

Example: Linear regression
MAP weights:

$$\mathbf{w} = \mathbf{A}^{-1}\mathbf{\Phi}^T\mathbf{y}$$

where $\mathbf{A} := (\mathbf{\Phi}^T\mathbf{\Phi} + \delta^2\mathbf{I}_C)$

When a cut is added as λ increases:

1. $\mathbf{A}$, $\text{chol}(\mathbf{A})$ can be updated in $\mathcal{O}(C^2)$
2. $\mathbf{w}$ can be updated in $\mathcal{O}(C^2 + N)$

Example: (S)GD trainable models
Maintain weights $\mathbf{w}$ directly. When r-th feature splits into two, reinitialize:

$$[w_1 \;\cdots\; w_{r-1} \;\; w_r \;\; w_{r+1} \;\cdots\; w_C]$$
$$\downarrow$$
$$[w_1 \;\cdots\; w_{r-1} \;\; \cancel{w_r} \;\; w_{r+1} \;\cdots\; w_C \;\; w_r \;\; w_r]$$

One SGD iteration is $\mathcal{O}(M)$.

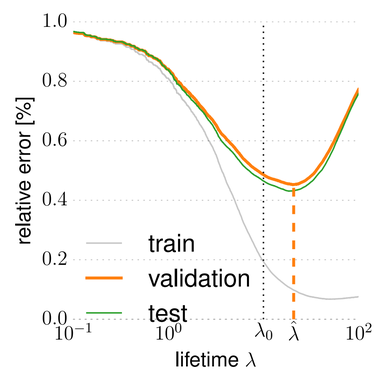


Figure : Recovering ground truth lifetime $\lambda_0 = 10$.

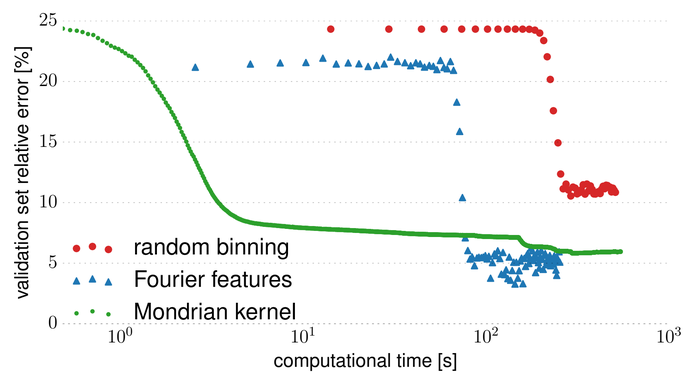


Figure : Performance vs computation time for kernel width cross validation. SGD on random features, dataset CPU [3].

Link to Mondrian forest

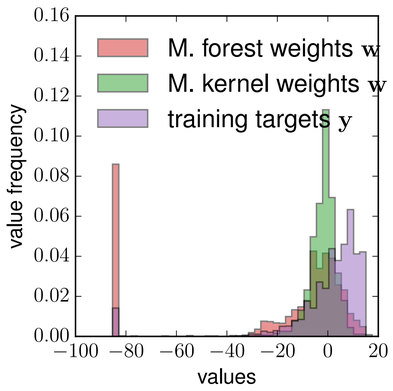
Mondrian forest [4, 5] = random forest with Mondrian process as randomization.

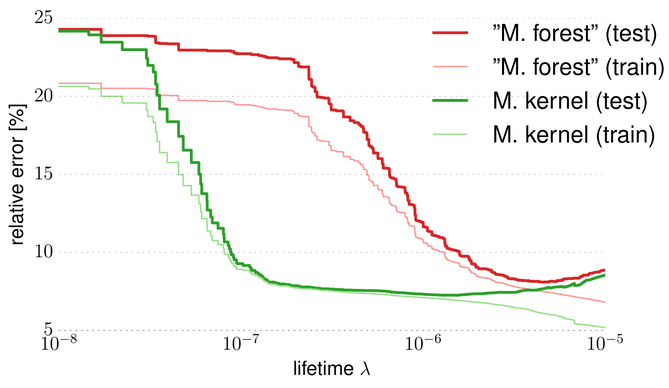
Mondrian forest	**Mondrian kernel**
sample M Mondrians	sample M Mondrians
$\frac{1}{M}\sum_{m=1}^{M} \text{loss}(y_n, \hat{y}_n^{(m)})$	$\text{loss}\left(y_n, \frac{1}{M}\sum_{m=1}^{M} \hat{y}_n^{(m)}\right)$
fit parameters independently	fit parameters jointly

Relationship between random forest and kernel methods:

Mondrian forest → (joint fitting) → **Mondrian kernel** → ($M \to \infty$) → Laplace kernel

Comparison of Mondrian kernel and Mondrian forest using the same set of Mondrian samples on the CPU dataset [3]:





References

[1] Daniel M. Roy and Yee Whye Teh. The Mondrian process. In *Adv. Neural Information Proc. Systems (NIPS)*, 2009.

[2] Daniel M. Roy. *Computability, inference and modeling in probabilistic programming*. PhD thesis, Massachusetts Institute of Technology, 2011.

[3] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Adv. Neural Information Proc. Systems (NIPS)*, 2007.

[4] Balaji Lakshminarayanan, Daniel M Roy, and Yee Whye Teh. Mondrian forests: Efficient online random forests. In *Adv. Neural Information Proc. Systems (NIPS)*, 2014.

[5] Balaji Lakshminarayanan, Daniel M Roy, and Yee Whye Teh. Mondrian forests for large scale regression when uncertainty matters. In *Int. Conf. Artificial Intelligence Stat. (AISTATS)*, 2016.

Contact: matej.balog@gmail.com