

RaFIO : un algorithme de forêts aléatoires économe en E/S

Camélia Slimani¹, Stéphane Rubini¹, Jalil Boukhobza²

¹: Univ. Bretagne Occidentale, Lab-STICC (UMR6285), France

²: ENSTA Bretagne, Lab-STICC(UMR6285), France

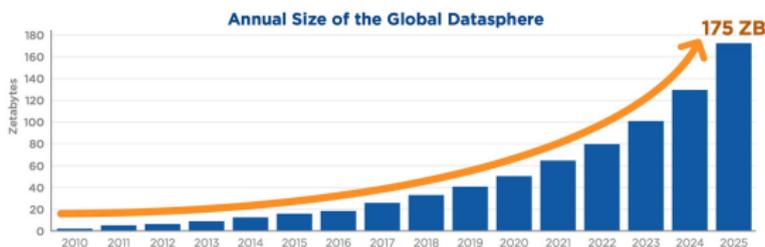
Workshop Per3S

13 Juin 2022

Agenda

- 1 Contexte
- 2 Motivation
- 3 Solution proposée

Contexte : la contrainte mémoire face à l'émergence de l'Edge Intelligence



- 50% des données sont produites sur les plateformes embarquées [1];
- La tendance actuelle consiste à traiter les données collectées directement sur les ces plateformes [2][3] pour répondre aux :
 - Contraintes de sécurité;
 - Coût de communication.
- Néanmoins, ces plateformes sont:
 - Limitées en terme d'espace de travail ;
 - Contraintes en énergie.

Forêts aléatoires (RF)

Algorithm Méthode de création d'un arbre de décision [4]

- 1: Création d'un bootstrap
- 2: **while** il existe un nœud impure **do**
- 3: Création aléatoire d'un sous-ensemble de propriétés F
- 4: **for** $f = 1$ to $|F| - 1$ **do**
- 5: Tentative de division du nœud n en deux nœuds enfants selon la propriété f
- 6: **end for**
- 7: Choix de la meilleure propriété f^* et création effective des nœuds enfants
- 8: **end while**

(1) Création du bootstrap ○ A C E C H A B F

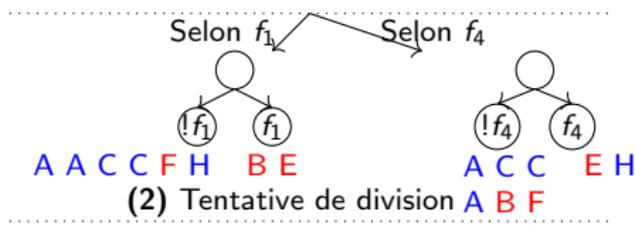
Bloc E/S	Label	f_1	f_2	f_3	f_4	Classe
(1)	A	0	0	0	0	0
	B	1	0	1	0	1
(2)	C	0	1	0	1	0
	D	0	0	0	0	0
(3)	E	1	1	1	1	1
	F	0	1	1	0	1
(4)	G	1	0	0	1	0
	H	0	1	0	1	0

Forêts aléatoires (RF)

Algorithm Méthode de création d'un arbre de décision [4]

- 1: Création d'un bootstrap
- 2: **while** il existe un nœud impure **do**
- 3: Création aléatoire d'un sous-ensemble de propriétés F
- 4: **for** $f = 1$ to $|F| - 1$ **do**
- 5: Tentative de division du nœud n en deux nœuds enfants selon la propriété f
- 6: **end for**
- 7: Choix de la meilleure propriété f^* et création effective des nœuds enfants
- 8: **end while**

(1) Création du bootstrap ○ A C E C H A B F



Bloc E/S	Label	f_1	f_2	f_3	f_4	Classe
(1)	A	0	0	0	0	0
	B	1	0	1	0	1
(2)	C	0	1	0	1	0
	D	0	0	0	0	0
(3)	E	1	1	1	1	1
	F	0	1	1	0	1
(4)	G	1	0	0	1	0
	H	0	1	0	1	0

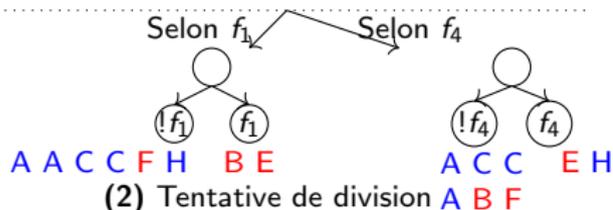
Forêts aléatoires (RF)

Algorithm Méthode de création d'un arbre de décision [4]

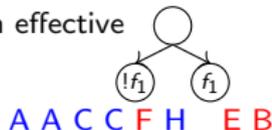
- 1: Création d'un bootstrap
- 2: **while** il existe un nœud impure **do**
- 3: Création aléatoire d'un sous-ensemble de propriétés F
- 4: **for** $f = 1$ to $|F| - 1$ **do**
- 5: Tentative de division du nœud n en deux nœuds enfants selon la propriété f
- 6: **end for**
- 7: Choix de la meilleure propriété f^* et création effective des nœuds enfants
- 8: **end while**

Bloc E/S	Label	f_1	f_2	f_3	f_4	Classe
(1)	A	0	0	0	0	0
	B	1	0	1	0	1
(2)	C	0	1	0	1	0
	D	0	0	0	0	0
(3)	E	1	1	1	1	1
	F	0	1	1	0	1
(4)	G	1	0	0	1	0
	H	0	1	0	1	0

(1) Création du bootstrap ○ A C E C H A B F



(3) Division effective



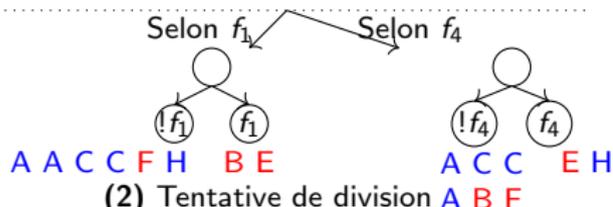
Forêts aléatoires (RF)

Algorithm Méthode de création d'un arbre de décision [4]

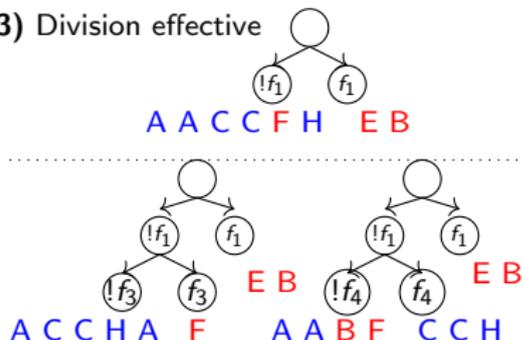
- 1: Création d'un bootstrap
- 2: **while** il existe un nœud impure **do**
- 3: Création aléatoire d'un sous-ensemble de propriétés F
- 4: **for** $f = 1$ to $|F| - 1$ **do**
- 5: Tentative de division du nœud n en deux nœuds enfants selon la propriété f
- 6: **end for**
- 7: Choix de la meilleure propriété f^* et création effective des nœuds enfants
- 8: **end while**

Bloc E/S	Label	f_1	f_2	f_3	f_4	Classe
(1)	A	0	0	0	0	0
	B	1	0	1	0	1
(2)	C	0	1	0	1	0
	D	0	0	0	0	0
(3)	E	1	1	1	1	1
	F	0	1	1	0	1
(4)	G	1	0	0	1	0
	H	0	1	0	1	0

(1) Création du bootstrap ○ A C E C H A B F



(3) Division effective



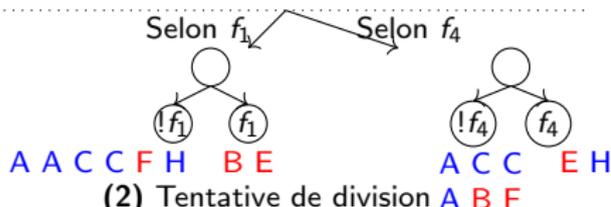
Forêts aléatoires (RF)

Algorithm Méthode de création d'un arbre de décision [4]

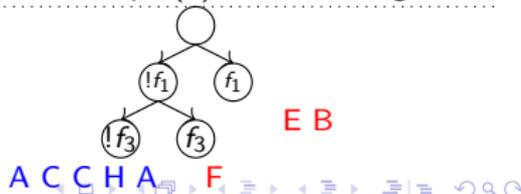
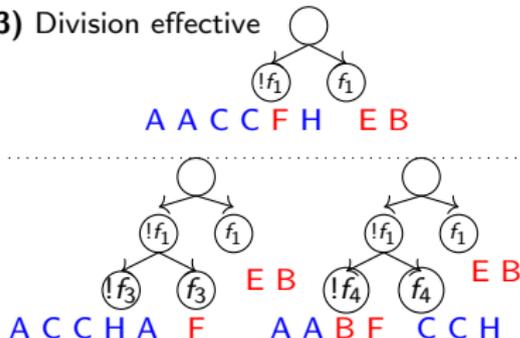
- 1: Création d'un bootstrap
- 2: **while** il existe un nœud impure **do**
- 3: Création aléatoire d'un sous-ensemble de propriétés F
- 4: **for** $f = 1$ to $|F| - 1$ **do**
- 5: Tentative de division du nœud n en deux nœuds enfants selon la propriété f
- 6: **end for**
- 7: Choix de la meilleure propriété f^* et création effective des nœuds enfants
- 8: **end while**

Bloc E/S	Label	f_1	f_2	f_3	f_4	Classe
(1)	A	0	0	0	0	0
	B	1	0	1	0	1
(2)	C	0	1	0	1	0
	D	0	0	0	0	0
(3)	E	1	1	1	1	1
	F	0	1	1	0	1
(4)	G	1	0	0	1	0
	H	0	1	0	1	0

(1) Création du bootstrap ○ A C E C H A B F



(3) Division effective



Motivation: l'algorithme des RF d'un point de vue E/S

On suppose un espace de travail qui peut contenir 4 éléments et un bloc d'E/S de 2 éléments.

Bloc E/S	Label	f_1	f_2	f_3	f_4	Classe
(1)	A	0	0	0	0	0
	B	1	0	1	0	1
(2)	C	0	1	0	1	0
	D	0	0	0	0	0
(3)	E	1	1	1	1	1
	F	0	1	1	0	1
(4)	G	1	0	0	1	0
	H	0	1	0	1	0

Donnée utile

Donnée inutile

Motivation: l'algorithme des RF d'un point de vue E/S

On suppose un espace de travail qui peut contenir 4 éléments et un bloc d'E/S de 2 éléments.

Nœud	Éléments	Blocs accédés	Pourcentage de données utilisées par bloc
N_0	{A, A, B, C, C, E, F, H}	(1), (2), (3), (4)	100%, 50%, 100%, 50%

Bloc E/S	Label	f_1	f_2	f_3	f_4	Classe
(1)	A	0	0	0	0	0
	B	1	0	1	0	1
(2)	C	0	1	0	1	0
	D	0	0	0	0	0
(3)	E	1	1	1	1	1
	F	0	1	1	0	1
(4)	G	1	0	0	1	0
	H	0	1	0	1	0

 Donnée utile

 Donnée inutile

Motivation: l'algorithme des RF d'un point de vue E/S

On suppose un espace de travail qui peut contenir 4 éléments et un bloc d'E/S de 2 éléments.

Nœud	Éléments	Blocs accédés	Pourcentage de données utilisées par bloc
N_0	{A, A, B, C, C, E, F, H}	(1), (2), (3), (4)	100%, 50%, 100%, 50%
N_1	{A, A, C, C, F, H}	(1), (2), (3), (4)	50%, 50%, 50%, 50%

Bloc E/S	Label	f_1	f_2	f_3	f_4	Classe
(1)	A	0	0	0	0	0
	B	1	0	1	0	1
(2)	C	0	1	0	1	0
	D	0	0	0	0	0
(3)	E	1	1	1	1	1
	F	0	1	1	0	1
(4)	G	1	0	0	1	0
	H	0	1	0	1	0

Donnée utile

Donnée inutile

Motivation: l'algorithme des RF d'un point de vue E/S

On suppose un espace de travail qui peut contenir 4 éléments et un bloc d'E/S de 2 éléments.

Nœud	Éléments	Blocs accédés	Pourcentage de données utilisées par bloc
N_0	{A, A, B, C, C, E, F, H}	(1), (2), (3), (4)	100%, 50%, 100%, 50%
N_1	{A, A, C, C, F, H}	(1), (2), (3), (4)	50%, 50%, 50%, 50%
N_2	{B, E}	(1), (3)	50%, 50%

Bloc E/S	Label	f_1	f_2	f_3	f_4	Classe
(1)	A	0	0	0	0	0
	B	1	0	1	0	1
(2)	C	0	1	0	1	0
	D	0	0	0	0	0
(3)	E	1	1	1	1	1
	F	0	1	1	0	1
(4)	G	1	0	0	1	0
	H	0	1	0	1	0

Donnée utile

Donnée inutile

Motivation: l'algorithme des RF d'un point de vue E/S

On suppose un espace de travail qui peut contenir 4 éléments et un bloc d'E/S de 2 éléments.

Nœud	Éléments	Blocs accédés	Pourcentage de données utilisées par bloc
N_0	{A, A, B, C, C, E, F, H}	(1), (2), (3), (4)	100%, 50%, 100%, 50%
N_1	{A, A, C, C, F, H}	(1), (2), (3), (4)	50%, 50%, 50%, 50%
N_2	{B, E}	(1), (3)	50%, 50%
N_3	{A, A, C, C, H}	(1), (2), (4)	50%, 50%, 50%

Bloc E/S	Label	f_1	f_2	f_3	f_4	Classe
(1)	A	0	0	0	0	0
	B	1	0	1	0	1
(2)	C	0	1	0	1	0
	D	0	0	0	0	0
(3)	E	1	1	1	1	1
	F	0	1	1	0	1
(4)	G	1	0	0	1	0
	H	0	1	0	1	0

Donnée utile

Donnée inutile

Motivation: l'algorithme des RF d'un point de vue E/S

On suppose un espace de travail qui peut contenir 4 éléments et un bloc d'E/S de 2 éléments.

Nœud	Éléments	Blocs accédés	Pourcentage de données utilisées par bloc
N_0	{A, A, B, C, C, E, F, H}	(1), (2), (3), (4)	100%, 50%, 100%, 50%
N_1	{A, A, C, C, F, H}	(1), (2), (3), (4)	50%, 50%, 50%, 50%
N_2	{B, E}	(1), (3)	50%, 50%
N_3	{A, A, C, C, H}	(1), (2), (4)	50%, 50%, 50%
N_4	{F}	(3)	50%

Bloc E/S	Label	f_1	f_2	f_3	f_4	Classe
(1)	A	0	0	0	0	0
	B	1	0	1	0	1
(2)	C	0	1	0	1	0
	D	0	0	0	0	0
(3)	E	1	1	1	1	1
	F	0	1	1	0	1
(4)	G	1	0	0	1	0
	H	0	1	0	1	0

Donnée utile

Donnée inutile

Motivation: l'algorithme des RF d'un point de vue E/S

On suppose un espace de travail qui peut contenir 4 éléments et un bloc d'E/S de 2 éléments.

Nœud	Éléments	Blocs accédés	Pourcentage de données utilisées par bloc
N_0	{A, A, B, C, C, E, F, H}	(1), (2), (3), (4)	100%, 50%, 100%, 50%
N_1	{A, A, C, C, F, H}	(1), (2), (3), (4)	50%, 50%, 50%, 50%
N_2	{B, E}	(1), (3)	50%, 50%
N_3	{A, A, C, C, H}	(1), (2), (4)	50%, 50%, 50%
N_4	{F}	(3)	50%

Bloc E/S	Label	f_1	f_2	f_3	f_4	Classe
(1)	A	0	0	0	0	0
	B	1	0	1	0	1
(2)	C	0	1	0	1	0
	D	0	0	0	0	0
(3)	E	1	1	1	1	1
	F	0	1	1	0	1
(4)	G	1	0	0	1	0
	H	0	1	0	1	0

Donnée utile

Donnée inutile

- La faible localité spatiale ;
- Les mouvements de données inutiles.

La solution proposée, RaFIO, est fondée sur deux mécanismes :

- 1 **Réorganisation du data-set** : l'objectif est d'augmenter la localité spatiale de l'algorithme ;

La solution proposée, RaFIO, est fondée sur deux mécanismes :

- 1 **Réorganisation du data-set** : l'objectif est d'augmenter la localité spatiale de l'algorithme ;
- 2 **Accès aux données à la demande** : l'objectif est d'accéder les données utiles uniquement.

References I

-  J. Gantz D. Reinsel and J. Rydning.
The Digitization of the World from Edge to Core.
page 28, 2018.
-  J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng.
A survey of machine learning for big data processing.
EURASIP Journal on Advances in Signal Processing, 2016.
-  O. Mutlu, S. Ghose, J. Gómez-Luna, and R. Ausavarungnirun.
Processing data where it makes sense: Enabling in-memory computation.
Microprocessors and Microsystems, 67, 2019.
-  L. Breiman.
Random Forests.
Machine Learning, 45(1), 2001.



M. Wright and A. Ziegler.

ranger: A fast implementation of random forests for high dimensional data in c++ and r.

Journal of Statistical Software, Articles, 77, 2017.



A. Anghel, N. Ioannou, T. P. Parnell, N. Papandreou, C. Mender-Dünner, and H. Pozidis.

Breadth-first, depth-next training of random forests.

ArXiv, abs/1910.06853, 2019.



Christophe Guyeux, Stéphane Chrétien, Gaby Bou Tayeh, Jacques Demerjian, and Jacques Bahi.

Introducing and comparing recent clustering methods for massive data management in the internet of things.

Journal of Sensor and Actuator Networks, 8(4), 2019.



Mayra Rodriguez, Cesar Comin, Dalcimar Casanova, Odemir Bruno, Diego Amancio, Francisco Rodrigues, and Luciano da F. Costa.
Clustering algorithms: A comparative approach.
PLOS ONE, 14, 12 2016.