



Une brève histoire de la parcimonie : *du traitement de signal à l'apprentissage profond*

R. Gribonval, E. Riccietti
Inria & ENS de Lyon, équipe Ockham

"Apprentissage frugal", un oxymore ?

■ Apprentissage = toujours plus ?

■ Frugalité = sobriété ?

- Compétition économique
- Maximiser performance
- Course au gigantisme
- Consommation effrénée



où placer le curseur ?

De quelles ressources parle-t-on ?

■ *Calcul* : bits & flops ...

- Coûts physiques : matériel (terres rares, eau ...), énergie, pollution ...
- ... mais aussi durée, rapidité, délais de traitement (avec impact sur le **service rendu**)

De quelles ressources parle-t-on ?

■ **Calcul** : bits & flops ...

- Coûts physiques : matériel (terres rares, eau ...), énergie, pollution ...
- ... mais aussi durée, rapidité, délais de traitement (avec impact sur le **service rendu**)

■ ... mais aussi **données**, annotées ou non

- Coûts économiques mais aussi sociaux (annotation), vie privée, ...

■ Deux régimes selon les cas d'usages



De quelles ressources parle-t-on ?

■ Calcul : bits & flops ...

- Coûts physiques : matériel (terres rares, eau ...), énergie, pollution ...
- ... mais aussi durée, rapidité, délais de traitement (avec impact sur le service rendu)

■ ... mais aussi *données*, annotées ou non

- Coûts économiques mais aussi sociaux (annotation), vie privée, ...

■ Deux régimes selon les cas d'usages

- opportunité = *beaucoup d'information* exploitable
- défi = *gourmandise en calcul et en mémoire*
 - enjeu = réduire les ressources nécessaires



De quelles ressources parle-t-on ?

■ Calcul : bits & flops ...

- Coûts physiques : matériel (terres rares, eau ...), énergie, pollution ...
- ... mais aussi durée, rapidité, délais de traitement (avec impact sur le service rendu)

■ ... mais aussi *données*, annotées ou non

- Coûts économiques mais aussi sociaux (annotation), vie privée, ...

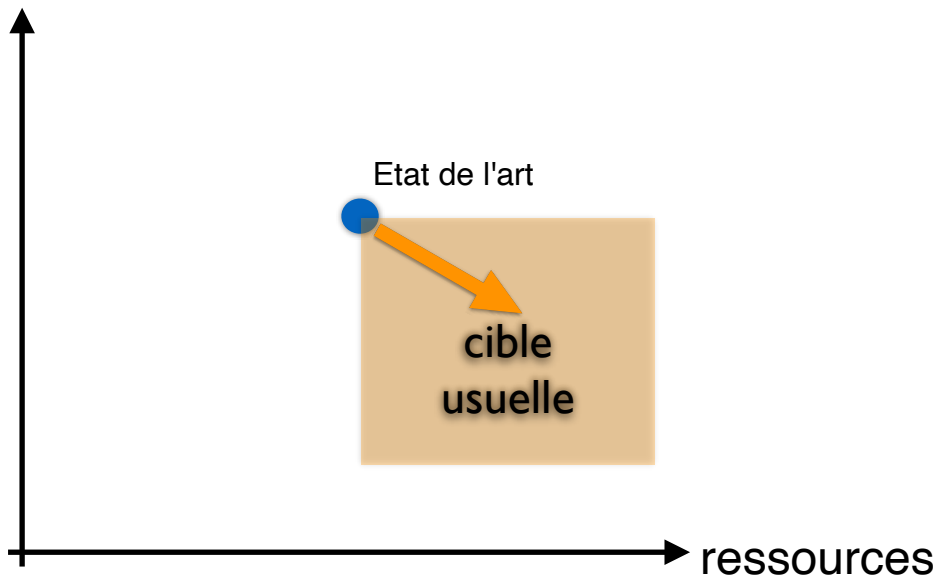
■ Deux régimes selon les cas d'usages

- défi = apprendre au mieux avec aussi peu de données que possible *malgré leur complexité*
- opportunité = *beaucoup d'information* exploitable
- défi = *gourmandise en calcul et en mémoire*
 - enjeu = réduire les ressources nécessaires



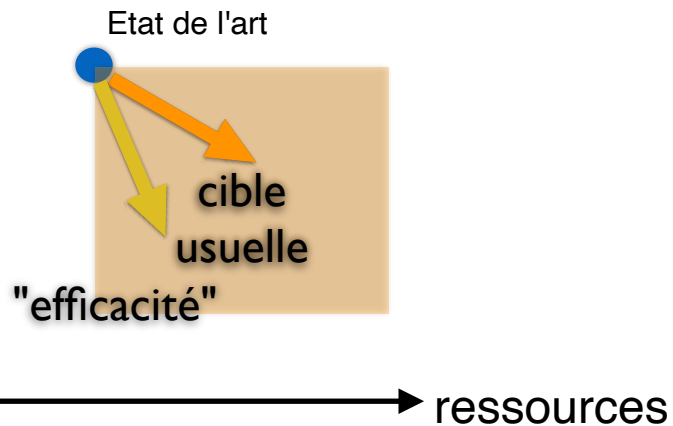
To be or not to be frugal ?

erreur de
prédiction/régression



To be or not to be frugal ?

erreur de
prédiction/régression

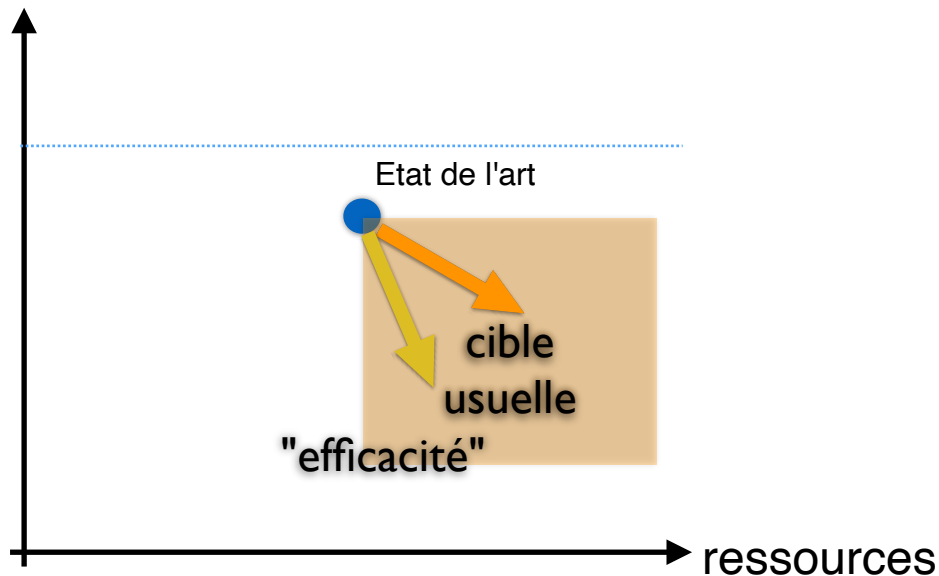


To be or not to be frugal ?

erreur de
prédiction/régression

erreur "acceptable" ?
performance "suffisante" ?

Choix / contexte
sociétal, économique,
géopolitique ...

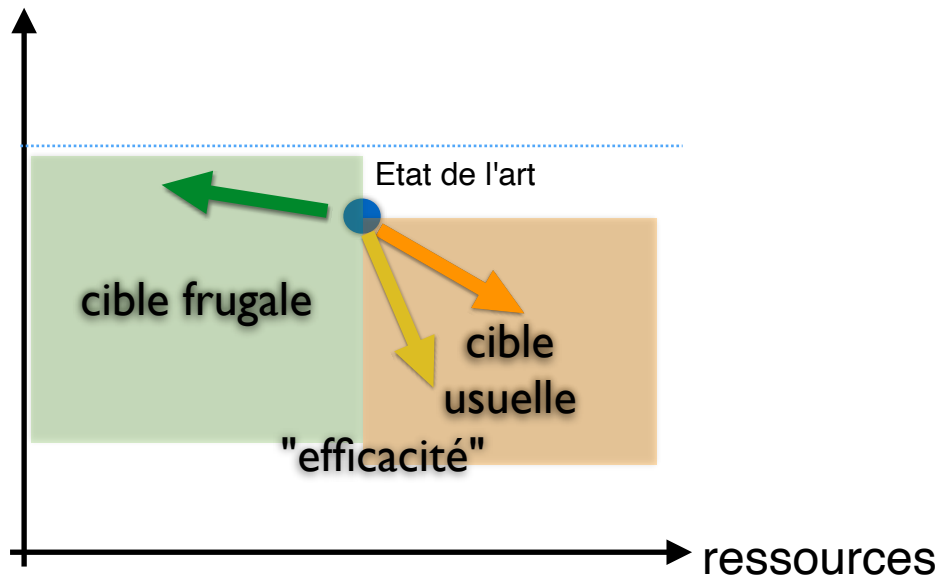


To be or not to be frugal ?

erreur de
prédiction/régression

erreur "acceptable" ?
performance "suffisante" ?

Choix / contexte
sociétal, économique,
géopolitique ...

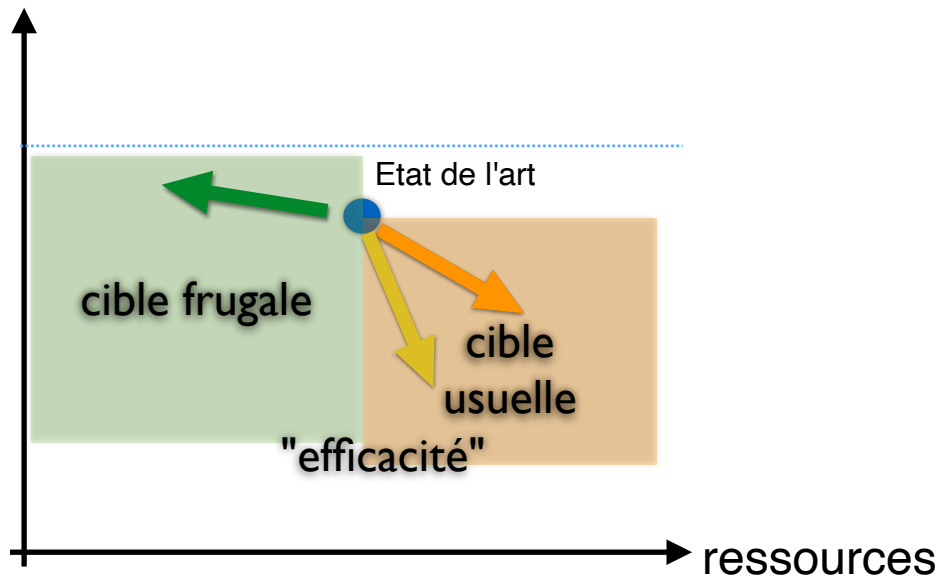


To be or not to be frugal ?

erreur de
prédiction/régression

erreur "acceptable" ?
performance "suffisante" ?

Choix / contexte
sociétal, économique,
géopolitique ...



Peut-on s'appuyer sur la notion de parcimonie ?

1. La parcimonie, une clé pour la frugalité ?

Un concept central : la parcimonie

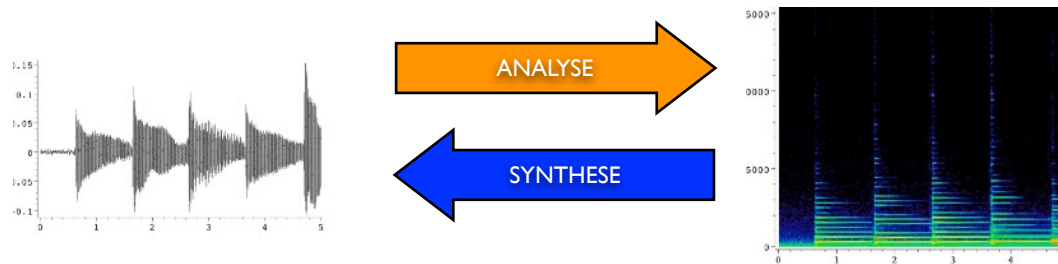
- **Parcimonie = la plupart des éléments sont nuls**
(dans un vecteur ou une matrice)
- On retrouve ce concept :
 - dans l'expression des *données* (variables explicatives)
 - dans l'expression des *paramètres* d'une méthode (degrés de liberté)
- Incarnation numérique du rasoir d'Ockham / Occam (XIVème siècle)



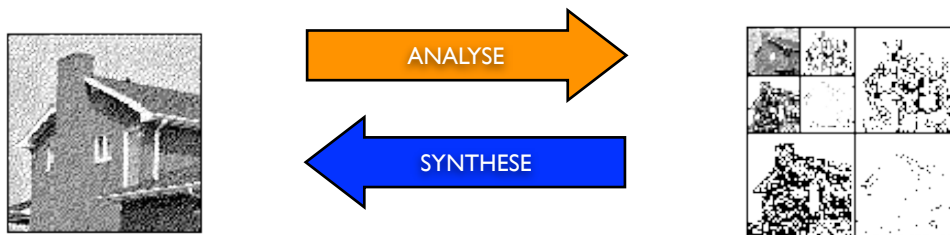
*Pluralitas non est ponenda
sine necessitate*

La parcimonie - un don de la nature nature

■ Audio : représentations temps-fréquence



■ Images : transformée en ondelettes

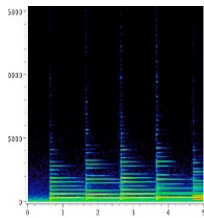


Parcimonie & frugalité

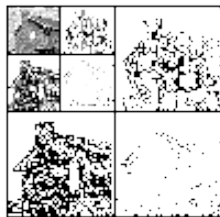
■ Parcimonie comme un objectif naturel:

- *bits (à la Shannon)*
- *ex: compression*

MP3, AAC



JPEG

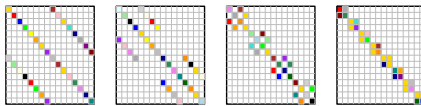


via **vecteur creux (= parcimonieux)**

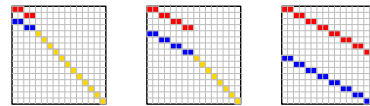
- *flops (à la Kolmogorov)*

- *ex: multiplication matricielle* $\mathbf{z} = \Psi \mathbf{x}$
 - coût générique en dim N : $O(N^2)$
 - transformées rapides $O(N)$ ou $O(N \log N)$

$\mathbf{F} =$



$\Psi =$



pour matrices (ondelettes, Fourier ...) qui sont
des **produits de quelques matrices creuses**

Parcimonie en traitement du signal : histoire et évolution

■ A l'origine :

■ Débruitage par seuillage

■ Problèmes inverses

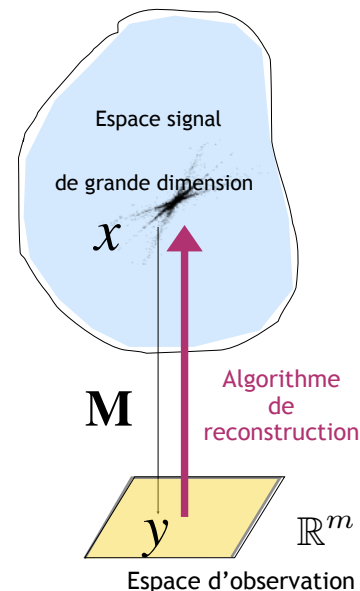
■ But = reconstruire à partir d'*observations partielles*

- Restauration de données (paquets manquants ...)
- Imagerie (tomographie, inpainting...)
- Exemple : défloutage en microscopie



■ Approche classique

- Modèle "parcimonieux" : hypothèse nécessaire
 - + variantes "modèle simple / de faible complexité / faible rang / faible dimension"
- Algorithmes de complexité bornée munis de solides garanties
 - typiquement : algorithmes gloutons, optimisation convexe



$$y = Mx$$

Parcimonie en traitement du signal : histoire et évolution

■ A l'origine :

■ Débruitage par seuillage

■ Problèmes inverses

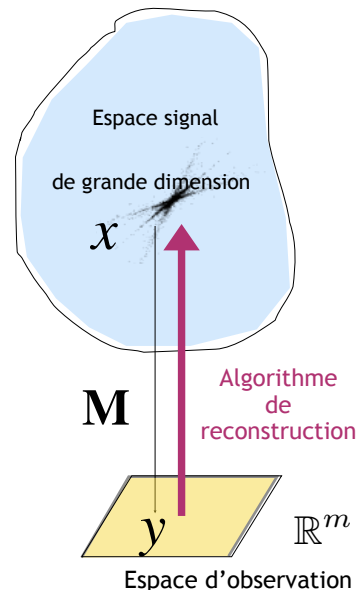
■ But = reconstruire à partir d'*observations partielles*

- Restauration de données (paquets manquants ...)
- Imagerie (tomographie, inpainting...)
- Exemple : défloutage en microscopie



■ Approche classique

- Modèle "parcimonieux" : hypothèse nécessaire
 - + variantes "modèle simple / de faible complexité / faible rang / faible dimension"
- Algorithmes de complexité bornée munis de solides garanties
 - typiquement : algorithmes gloutons, optimisation convexe



$$y = \mathbf{M}x$$

➔ Parcimonie comme connaissance *a priori* pour identifier des variables latentes



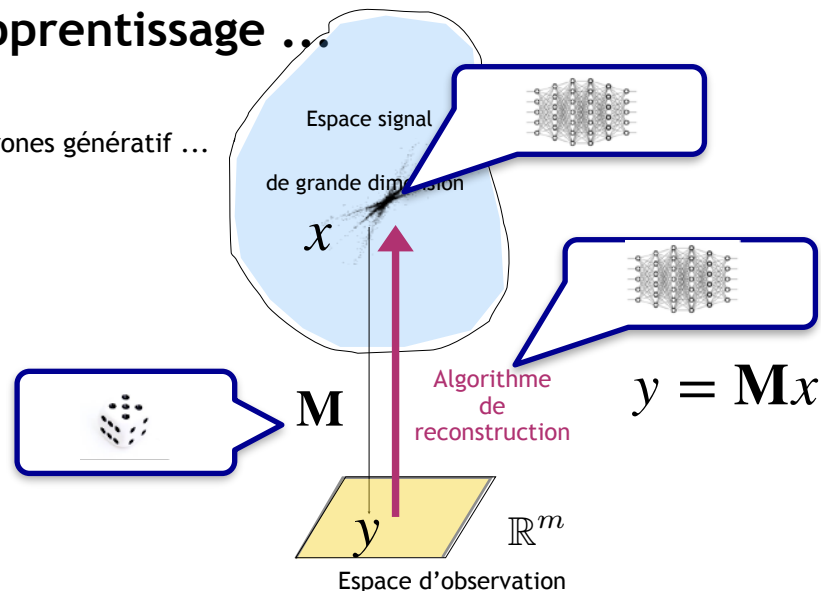
Parcimonie : histoire et évolution

■ Un peu plus récemment : accent sur l'apprentissage ...

- du modèle "parcimonieux"
 - Explicitement ou implicitement
 - Dictionnaire, variété de faible dimension, réseau de neurones génératif ...
- de l'algorithme de reconstruction
 - Débruiteur appris, algorithme déroulé, ``plug and play"

■ ... et la réduction de dimension

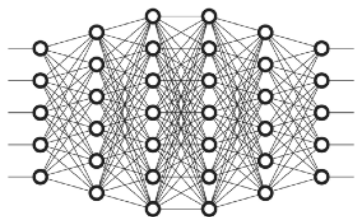
- Conception de l'opérateur M
 - **Compressive sensing** via projections aléatoires
 - Extension à l'apprentissage: **compressive learning**
 - via *sketching* et *random features*



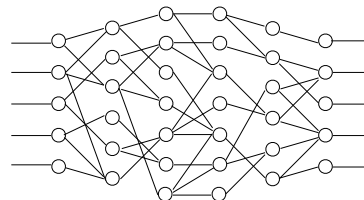
2. Parcimonie et apprentissage profond ?

Parcimonie et apprentissage profond ?

■ Réseau de l'état de l'art



■ Réseau compressé

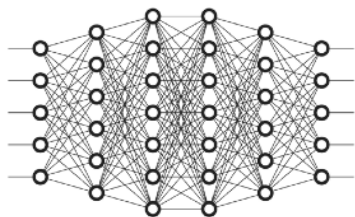


- beaucoup de paramètres
 - dont de grandes matrices
- format numérique haute précision
 - beaucoup de bits par paramètre

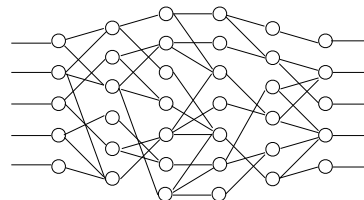
- connections parcimonieuses
- quantification grossière

Parcimonie et apprentissage profond ?

■ Réseau de l'état de l'art



■ Réseau compressé



- beaucoup de paramètres
 - dont de grandes matrices
- format numérique haute précision
 - beaucoup de bits par paramètre

- connections parcimonieuses
- quantification grossière

**Exploiter le savoir-faire
parcimonieux usuel ?**

Apprentissage et optimisation

■ Exemple : réseau ReLU

- Paramètres = poids et biais θ
- Entraînement sur données (x_i, y_i) = problème d'optimisation

$$\min_{\theta} \sum_i (y_i - f_{\theta}(x_i))^2$$

Apprentissage et optimisation

■ Exemple : réseau ReLU

- Paramètres = poids et biais θ
- Entraînement sur données (x_i, y_i) = problème d'optimisation

$$\min_{\theta} \sum_i (y_i - f_{\theta}(x_i))^2$$

■ Apprentissage avec parcimonie "dure" ?

- Imposer que les matrices de poids soient *creuses*
 - *notion de support : indices des coefficients autorisés à être non nuls*
- Approche naturelle en 2 étapes / alternée
 - trouver / mettre à jour le support
 - optimiser poids sous contrainte de support

Apprentissage et optimisation

■ Exemple : réseau ReLU

- Paramètres = poids et biais θ
- Entraînement sur données (x_i, y_i) = problème d'optimisation

$$\min_{\theta} \sum_i (y_i - f_{\theta}(x_i))^2 \quad \text{s.t. } \theta_{Ic} = 0$$

■ Apprentissage avec parcimonie "dure" ?

- Imposer que les matrices de poids soient *creuses*
 - *notion de support : indices des coefficients autorisés à être non nuls*
- Approche naturelle en 2 étapes / alternée
 - trouver / mettre à jour le support
 - optimiser poids sous contrainte de support

Apprentissage et optimisation

■ Exemple : réseau ReLU

- Paramètres = poids et biais θ
- Entraînement sur données (x_i, y_i) = problème d'optimisation

$$\inf_{\theta} \min_{\theta} \sum_i (y_i - f_{\theta}(x_i))^2 \quad \text{s.t. } \theta_{Ic} = 0$$

Oui mais ...

- NP-difficile
- et surtout : **Instable** (il n'existe pas toujours d'optimum)

■ Apprentissage avec parcimonie "dure" ?

- Imposer que les matrices de poids soient *creuses*
 - *notion de support : indices des coefficients autorisés à être non nuls*
- Approche naturelle en 2 étapes / alternée
 - trouver / mettre à jour le support
 - optimiser poids sous contrainte de support

SPURIOUS VALLEYS, NP-HARDNESS, AND TRACTABILITY OF SPARSE MATRIX FACTORIZATION WITH FIXED SUPPORT

QUOC-TUNG LE*, ELISA RICCIETTI*, AND REMI GRIBONVAL*

Does a sparse ReLU network training problem always admit an optimum?

Quoc-Tung Le

Elisa Riccietti

Rémi Gribonval

Apprentissage et optimisation

■ Exemple : réseau ReLU

- Paramètres = poids et biais θ
- Entraînement sur données (x_i, y_i) = problème d'optimisation

$$\inf_{\theta} \min_{\theta} \sum_i (y_i - f_{\theta}(x_i))^2 \quad \text{s.t. } \theta_{Ic} = 0$$

■ Apprentissage avec parcimonie "dure" ?

- Imposer que les matrices de poids soient *creuses*
 - *notion de support* : indices des coefficients autorisés à être non nuls
- Approche naturelle en 2 étapes / alternée
 - trouver / mettre à jour le support
 - optimiser poids sous contrainte de support

Oui mais ...

- NP-difficile
- et surtout : **Instable** (il n'existe pas toujours d'optimum)

Très différent du cas "classique"

$$\min_{x_I} \|z - A_I x_I\|_2^2$$

moindres carrés = facile

SPURIOUS VALLEYS, NP-HARDNESS, AND TRACTABILITY OF SPARSE MATRIX FACTORIZATION WITH FIXED SUPPORT

QUOC-TUNG LE*, ELISA RICCIETTI*, AND REMI GRIBONVAL*

Does a sparse ReLU network training problem always admit an optimum?

Quoc-Tung Le

Elisa Riccietti

Rémi Gribonval

Savoir faire parcimonieux ?

■ "Le support est la clé"

- **Difficile** = trouver *où* sont les coefficients non nuls
- **Facile** = trouver *les valeurs* des coefficients non nuls (moindres carrés)

■ Il suffit de seuiller

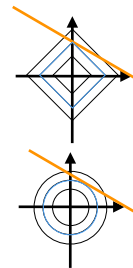
- Algorithmes gloutons ou proximaux : détection des "grands coefficients"

■ La minimisation L^1 promeut la parcimonie

- Basis Pursuit / Lasso

■ La minimisation L^2 n'induit pas de parcimonie

- Tikhonov / Ridge regression



Savoir faire parcimonieux ?

■ "Le support est la clé"

- **Difficile** = trouver *où* sont les coefficients non nuls
- **Facile** = trouver *les valeurs* des coefficients non nuls (moindres carrés)

■ Il suffit de seuiller

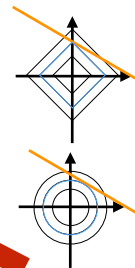
- Algorithmes gloutons et proximaux : détection des "grands coefficients"

■ La minimisation L^1 promeut la parcimonie

- Basis Pursuit / Lasso

■ La minimisation L^2 n'induit pas de parcimonie

- Tikhonov / Ridge regression



Remis en cause en contexte **profond / multilinéaire**

A FUNDAMENTAL PITFALL IN BLIND DECONVOLUTION
WITH SPARSE AND SHIFT-INVARIANT PRIORS

Alexis Benichoux¹, Emmanuel Vincent², Rémi Gribonval³

SPURIOUS VALLEYS, NP-HARDNESS, AND TRACTABILITY
OF SPARSE MATRIX FACTORIZATION WITH FIXED SUPPORT

QUOC-TUNG LE*, ELISA RICCIETTI*, AND REMI GRIBONVAL*

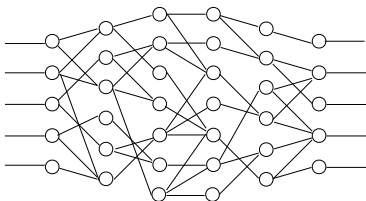
... mais fort potentiel de parcimonie *structurée*

3. Parcimonie et papillons

De la parcimonie profonde aux papillons

■ Parcimonie classique

- peu de connections *irrégulières*



■ parmi les défis :

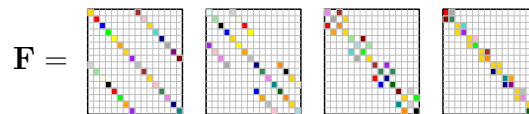
- ➔ NP-difficile + optimisation "instable"
(divergence possible de la descente de gradient / optimum "à l'infini")
- ➔ *format creux inefficace sur GPU*

see e.g. [Lim & al, 2019], [Petersen & al 2021] [Bereich & al 2023]

■ Parcimonie "papillon"

- schéma *structuré de parcimonie*

■ ex: DFT, ou transformée de Hadamard



- scénario: remplacer couches dense par produits creux structurés(

see e.g. [T. Dao & al, Monarch: Expressive Structured Matrices for Efficient and Accurate Training, ICML, 2022]

■ avancées récentes

- ✓ format butterfly **flexible**
- ✓ implémentation GPU **efficace**
- ✓ algorithme d'approximation **quasi-optimal**
- ✓ quantification spécifique : -30% de bits à précision égale comparé à l'arrondi usuel

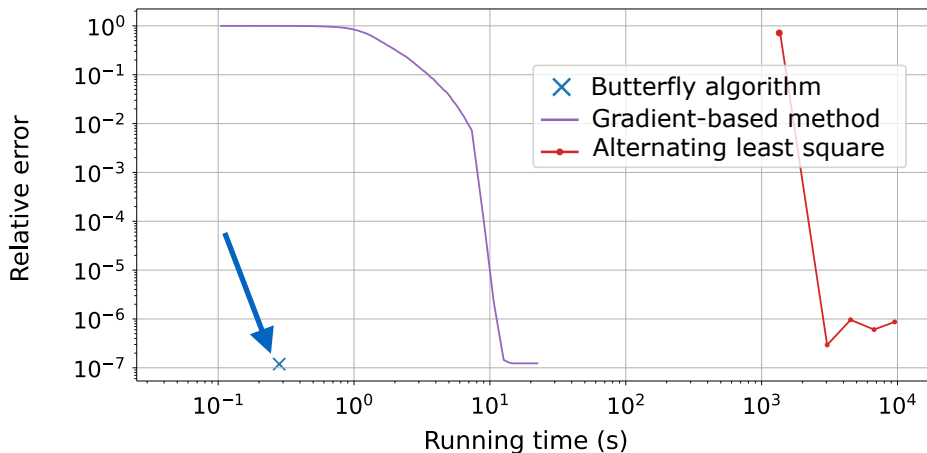
Efficacité pratique & garanties théoriques

■ Algorithme papillon

- But = solution approchée de

$$E(\mathbf{A}) := \inf_{\mathbf{X}_1, \dots, \mathbf{X}_L} \|\mathbf{A} - \mathbf{X}_1 \dots \mathbf{X}_L\|_F \quad (1)$$
 sous contraintes de parcimonie structurée "papillon"

- Approche : SVD tronquée de blocs extraits hiérarchiquement de $\mathbf{A}$



■ Garanties théoriques

- Quasi-optimalité de $(\hat{\mathbf{X}}_1, \dots, \hat{\mathbf{X}}_L) = \text{butterfly}(\mathbf{A})$

$$\|\mathbf{A} - \hat{\mathbf{X}}_1 \dots \hat{\mathbf{X}}_L\| \leq \sqrt{L-1} E(\mathbf{A})$$

- Corollaire: existence d'un optimum de (1)

■ Implémentation en Python

- *approximation* de n'importe quelle matrice $n \times n$ par une implémentation spécialisée (CPU/GPU) en $O(n \log n)$

→ `butterfly(A)`

✓ efficacité : comparable à la FFT pytorch

✓ flexibilité : applicable à toute matrice $\mathbf{A}$

`pip install lazylinop`

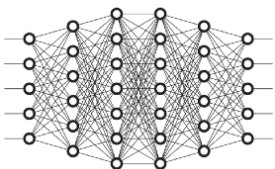
4. La parcimonie, une valeur d'avenir ?



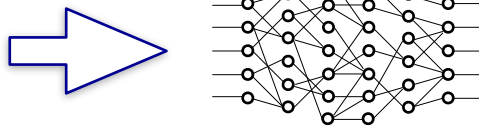
Parcimonie pour la frugalité ...

Idee naturelle : moins de paramètres = moins de bits / flops / watts / temps / matériau ...

Dense



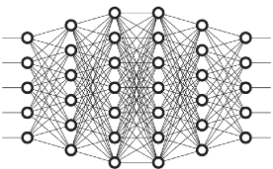
Parcimonie



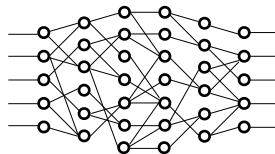
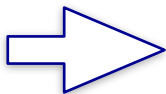
Parcimonie pour la frugalité ...

Idee naturelle : moins de paramètres = moins de bits / flops / watts / temps / matériau ...

Dense



Parcimonie



Le diable est dans les détails

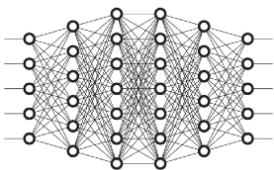
- peu adapté aux GPU
- instable, NP-difficile ...
- savoir-faire usuel remis en cause

Nouvelles questions scientifiques :-)

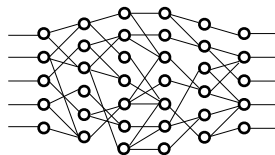
Parcimonie pour la frugalité ...

Idee naturelle : moins de paramètres = moins de bits / flops / watts / temps / matériau ...

Dense



Parcimonie



Parcimonie *structurée*

ex : architectures papillon
pour des matrices / couches linéaires



Le diable est dans les détails

- peu adapté aux GPU
- instable, NP-difficile ...
- savoir-faire usuel remis en cause

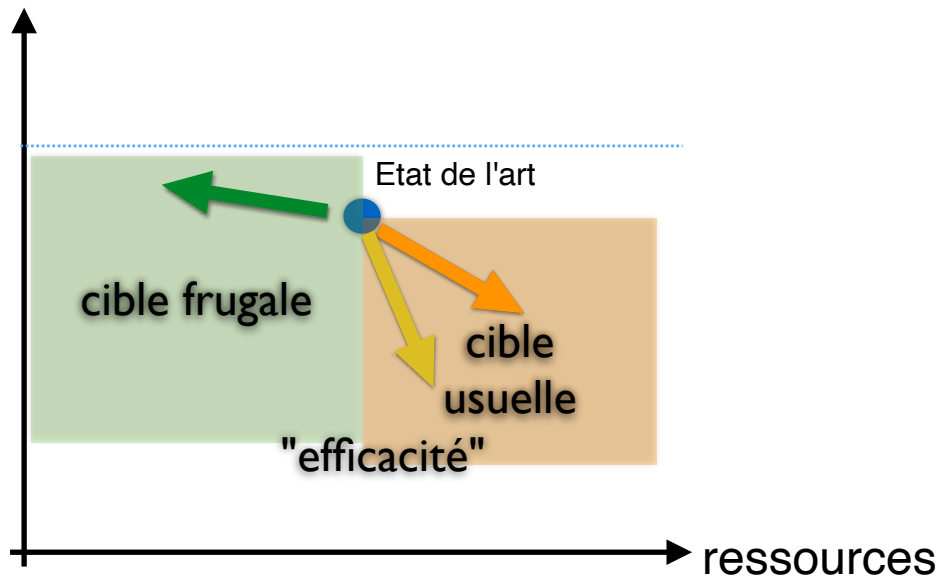
- ✓ Optimisé pour GPU & quantification
- ✓ Algorithme papillon
 - meilleur que descente de gradient
 - garanties théoriques

Nouvelles questions scientifiques :-)

To be or not to be frugal ?

erreur de
prédiction/régression

erreur "acceptable" ?
performance "suffisante" ?



To be or not to be frugal ?

erreur de
prédiction/régression

erreur "acceptable" ?
performance "suffisante" ?

Choix / contexte
sociétal, économique,
géopolitique ...

cible

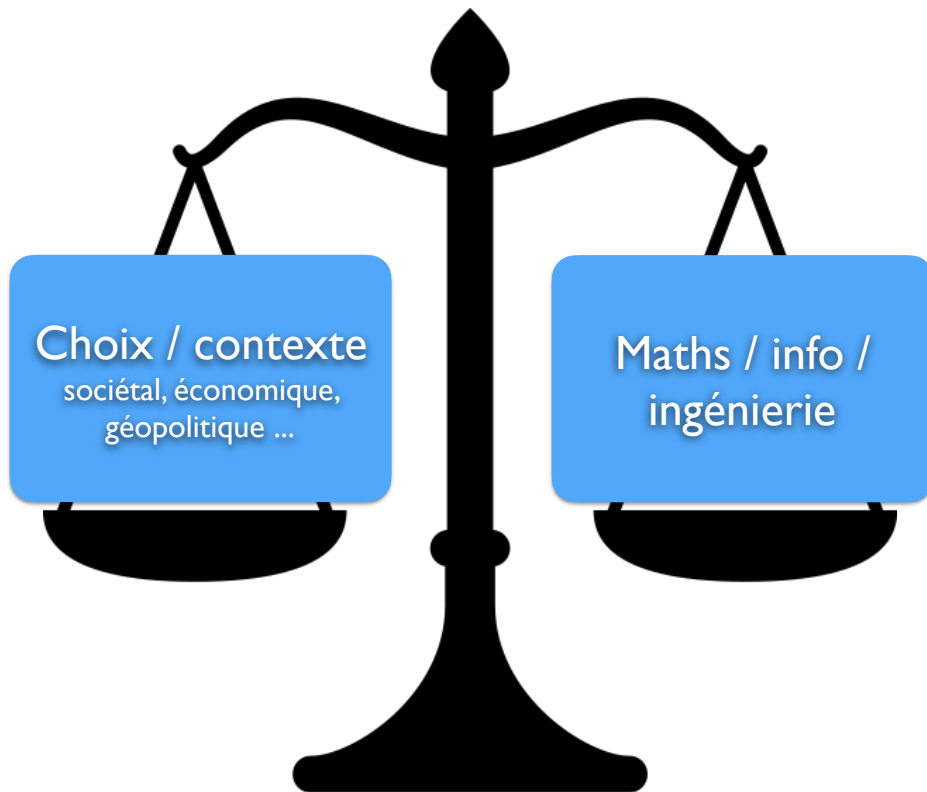
Maths / info /
ingénierie

Etat de l'art

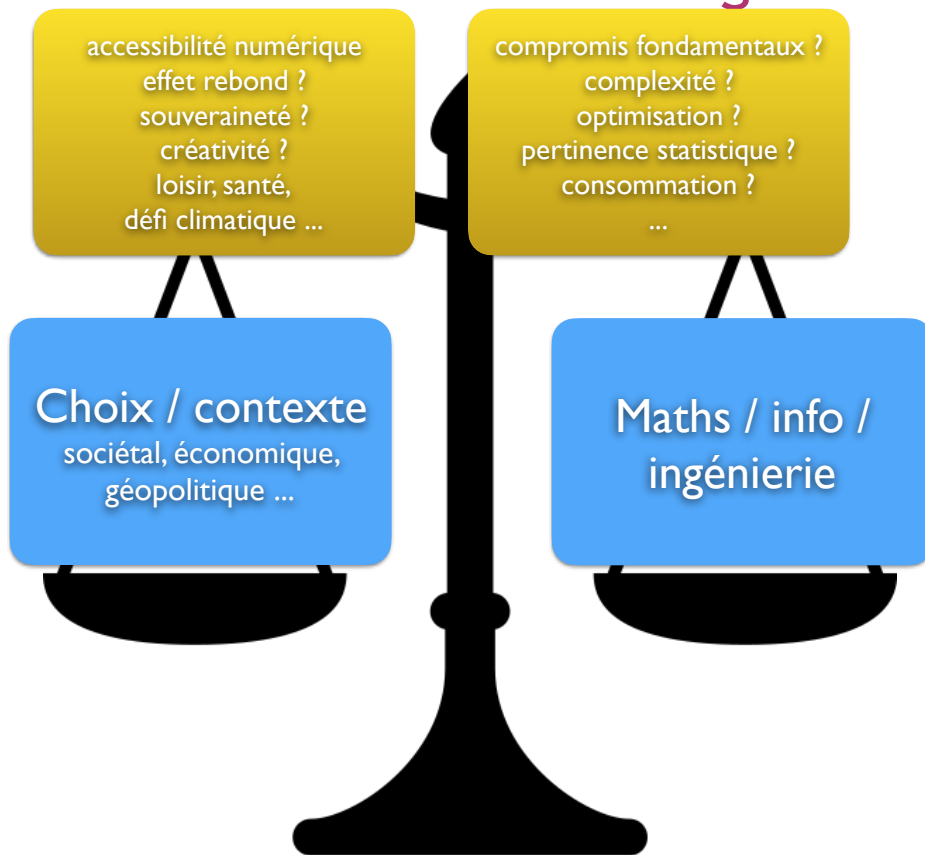
efficacité

ressources

To be or not to be frugal ?



To be or not to be frugal ?



References

Survey on deep sparsity

	Rapture of the deep: highs and lows of sparsity in a world of depths Rémi Gribonval, Elisa Riccietti, Quoc-Tung Le, Léon Zheng 2025   under revision in IEEE Signal Processing Magazine, special issue on The Mathematics of Deep Learning
---	--

Butterfly algorithm

	Butterfly factorization with error guarantees Quoc-Tung Le, Léon Zheng, Elisa Riccietti, Rémi Gribonval 2024   to appear in SIAM Journal on Matrix Analysis (SIMAX)
---	---

Quantization

	Optimal quantization of rank-one matrices in floating-point arithmetic—with applications to butterfly factorizations Rémi Gribonval, Theo Mary, Elisa Riccietti 2023   under revision in SIAM Journal on Scientific Computing (SISC)
---	--