La parcimonie, une valeur d'avenir pour l'apprentissage frugal ?

Rémi Gribonval, Equipe-projet Ockham

Journées de Recherche en Apprentissage Frugal
Grenoble, *13-14 décembre 2023*

"Apprentissage frugal", un oxymore ?

Apprentissage = toujours plus ?

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

Frugalité = sobriété ?

où placer le curseur ?

sources: https://commons.wikimedia.org/wiki/File:Cern_datacenter.jpg
https://commons.wikimedia.org/wiki/File:Be%C5%82chat%C3%B3w_Elektrownia.jpg
https://commons.wikimedia.org/wiki/File:Cimeti%C3%A8re_de_Loyasse_-_Tombes.jpg

De quelles ressources parle-t-on ?

- ***Calcul*** **: bits & flops ...**
  - Coûts physiques: matériel (terres rares, eau ...), énergie, pollution ...
  - Ainsi que: durée, rapidité, délais de traitement

De quelles ressources parle-t-on ?

- ***Calcul* : bits & flops ...**
  - Coûts physiques: matériel (terres rares, eau ...), énergie, pollution ...
  - Ainsi que: durée, rapidité, délais de traitement
- **... 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 ...
  - Ainsi que: durée, rapidité, délais de traitement
- **... 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 ...
  - Ainsi que: durée, rapidité, délais de traitement
- **... mais aussi *données*, annotées ou non**
  - Coûts économiques mais aussi sociaux (annotation), vie privée, ...

Deux régimes selon les cas d'usages

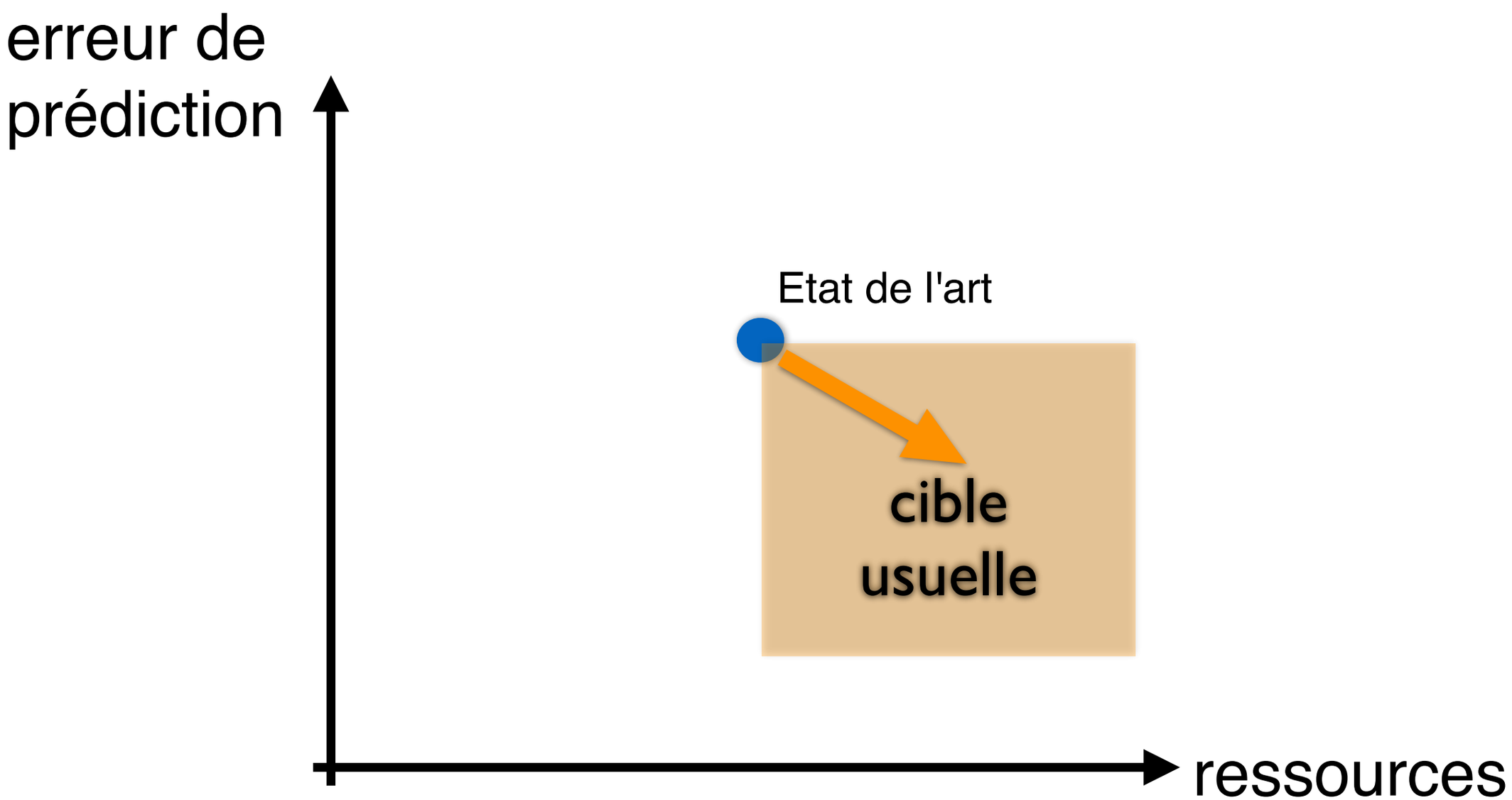
- **défi** = apprendre au mieux avec aussi peu de données que possible *malgré leur complexité*

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

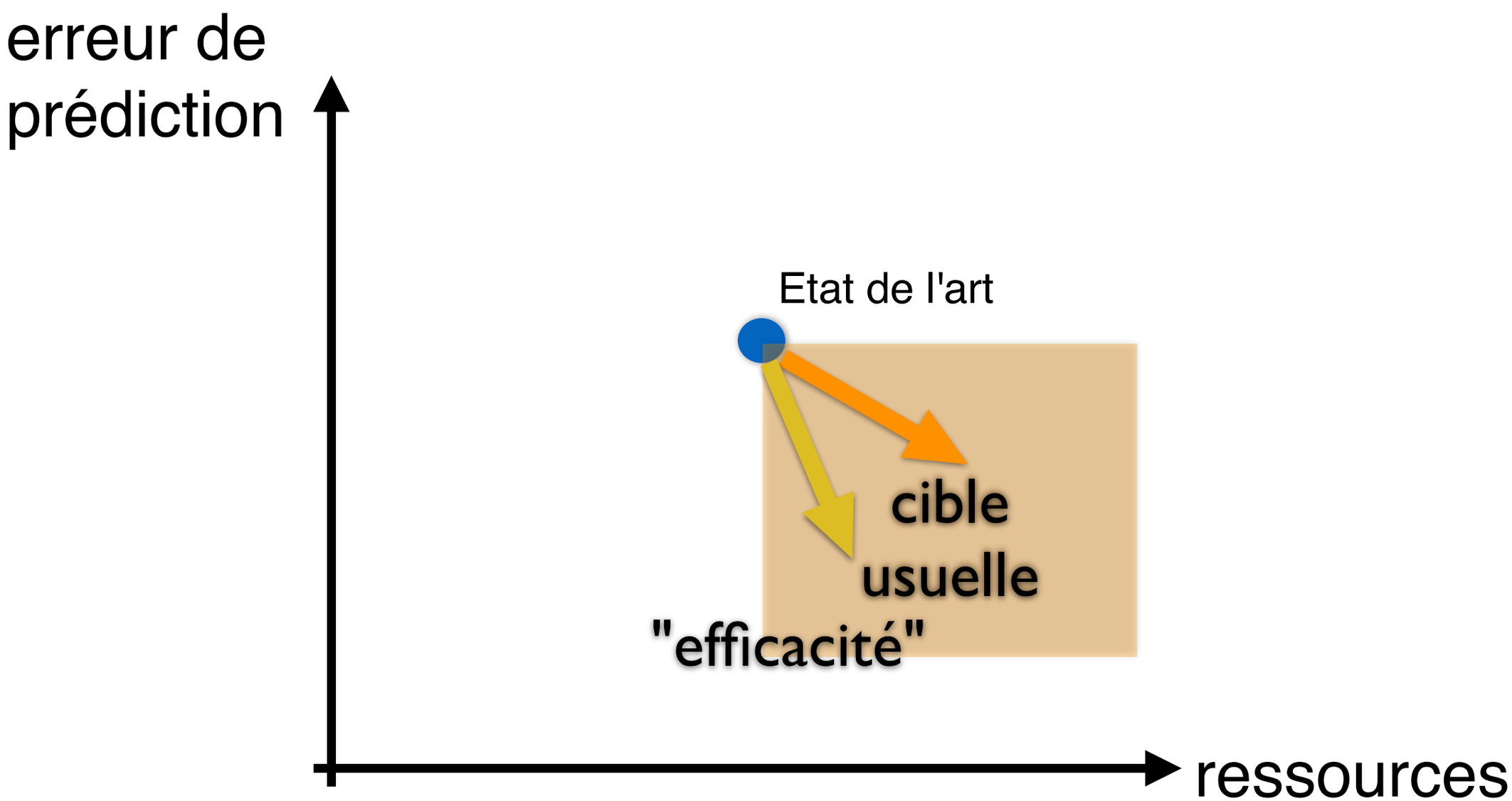
rareté ⟷ abondance

ressources **en données** disponibles

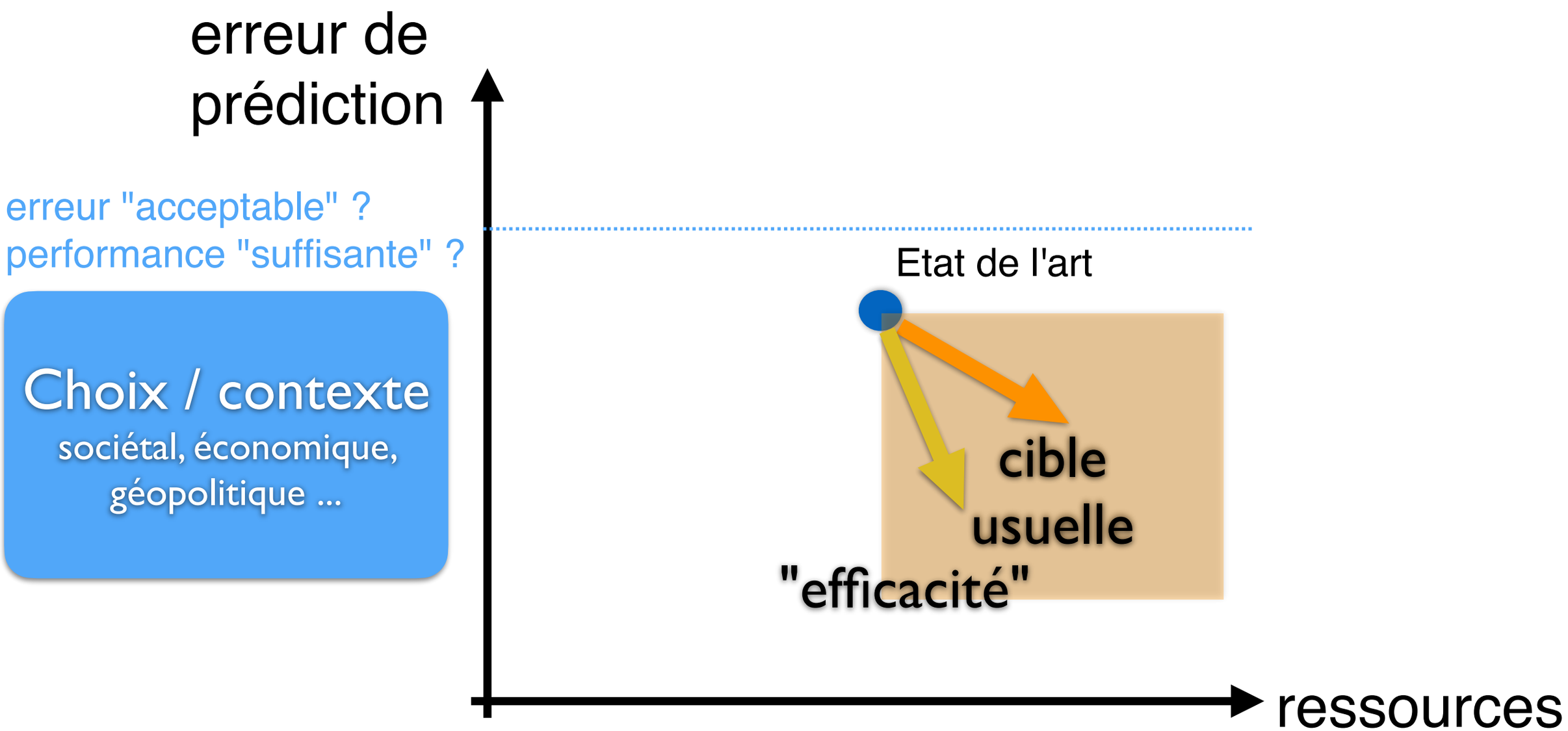
To be or not to be frugal ?



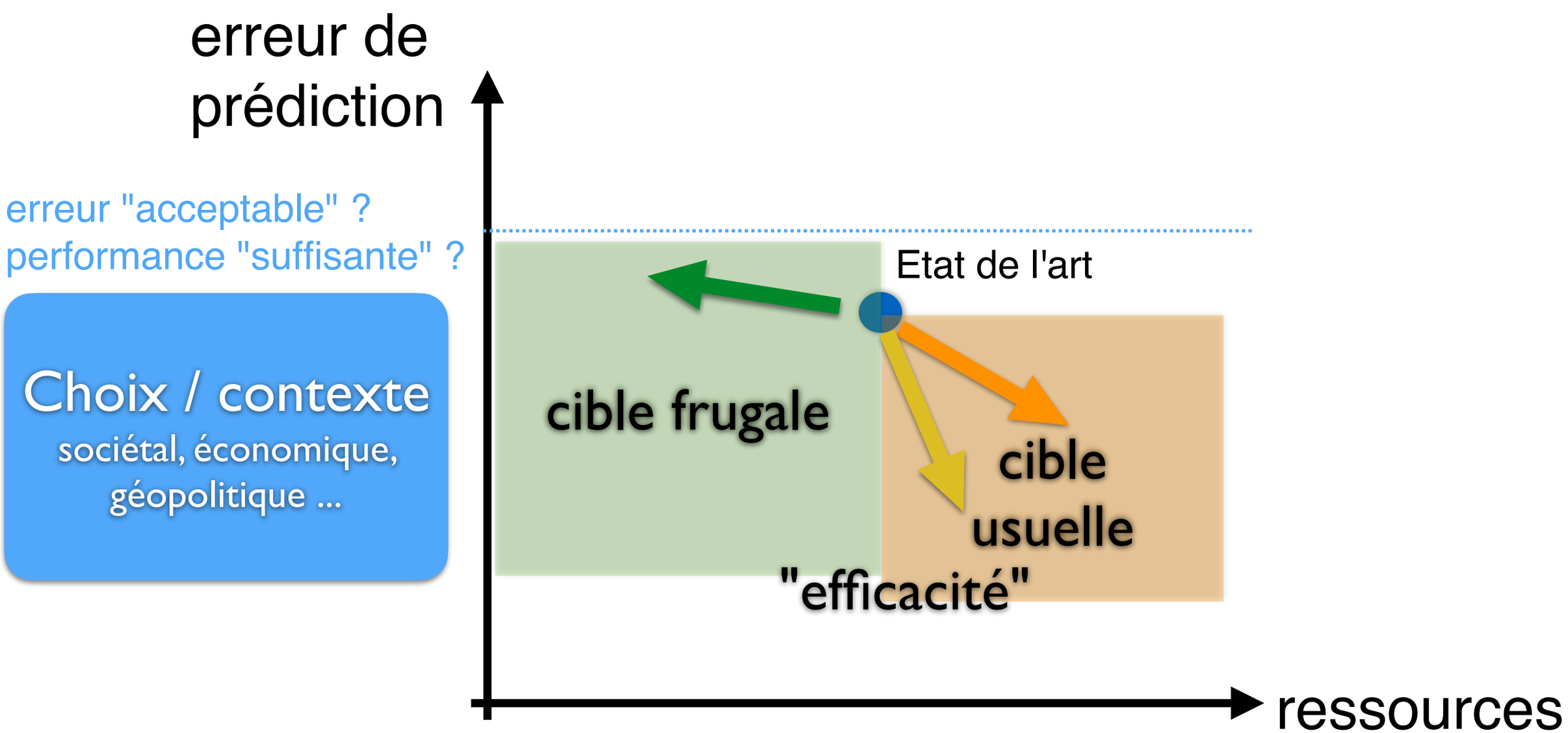
To be or not to be frugal ?



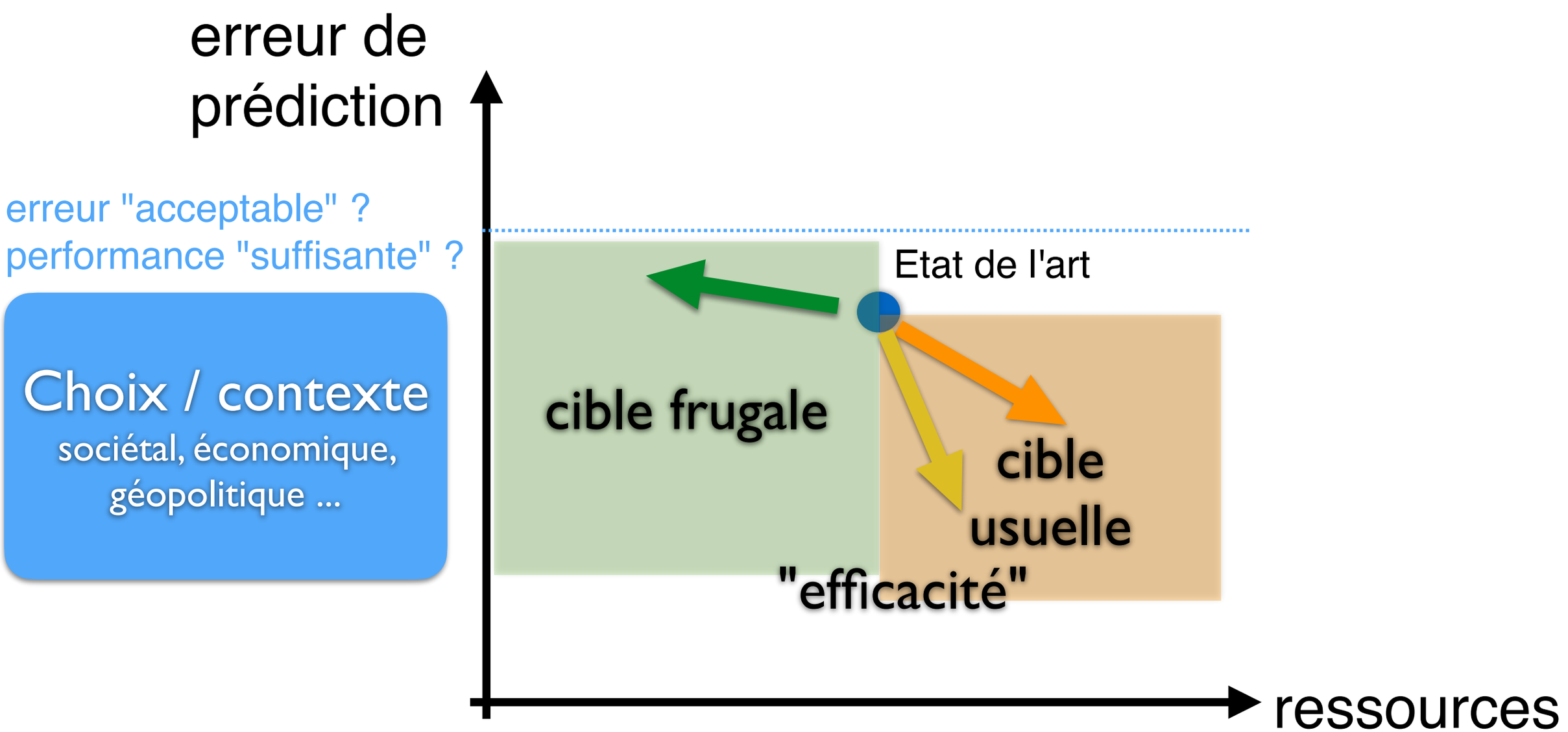
To be or not to be frugal ?



To be or not to be frugal ?



To be or not to be frugal ?

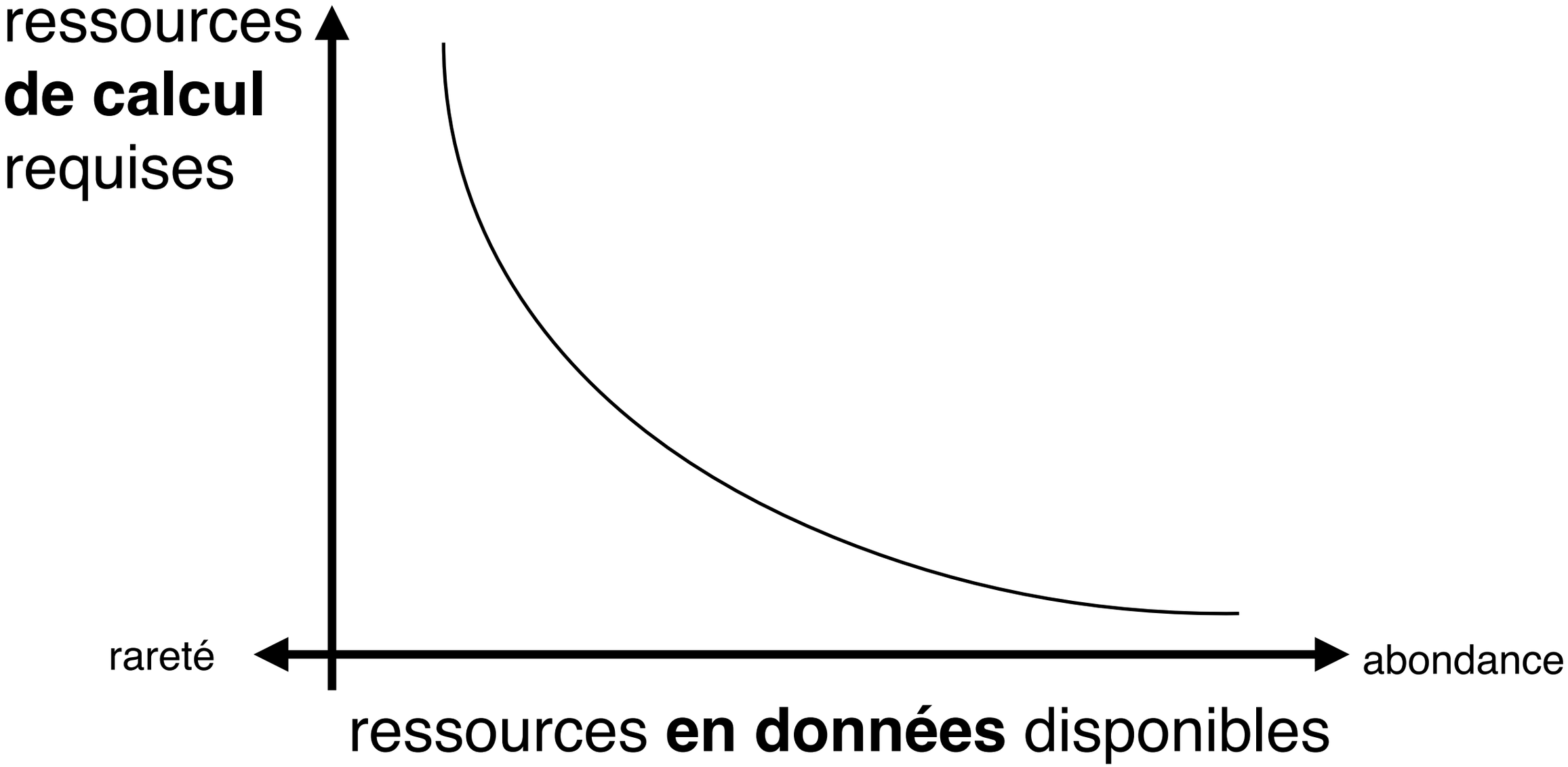


Défi I = maîtriser les compromis fondamentaux ressources / performance (courbe de Pareto)

Compromis données / calcul

[1] S. Shalev-Shwartz and N. Srebro. SVM optimization: inverse dependence on training set size. In *Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008)*, pages 928–935, Helsinki, Finland, 2008. ACM.

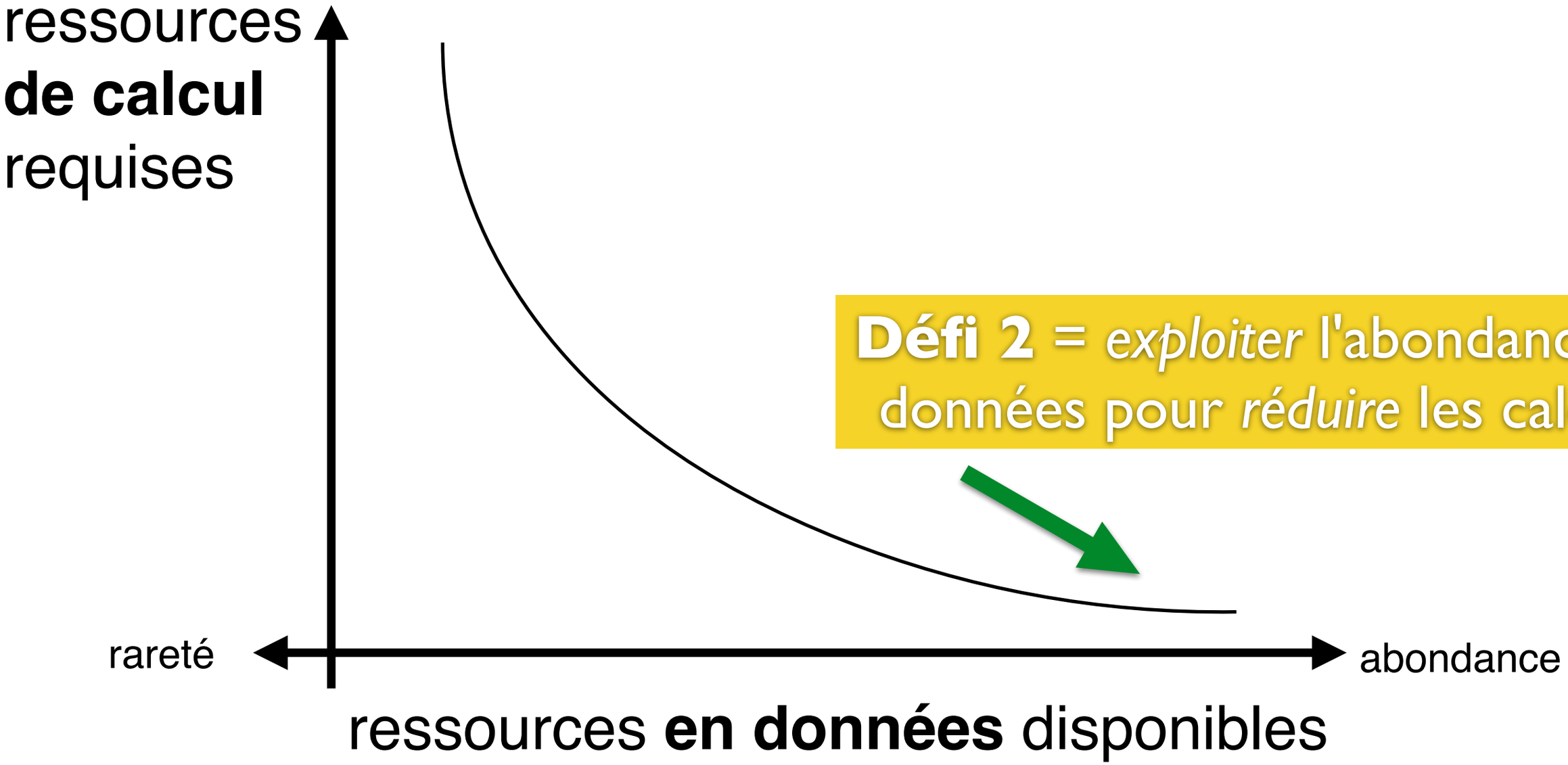
- **Pour atteindre une performance cible *fixée***



Compromis données / calcul

[1] S. Shalev-Shwartz and N. Srebro. SVM optimization: inverse dependence on training set size. In *Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008)*, pages 928–935, Helsinki, Finland, 2008. ACM.

- **Pour atteindre une performance cible *fixée***

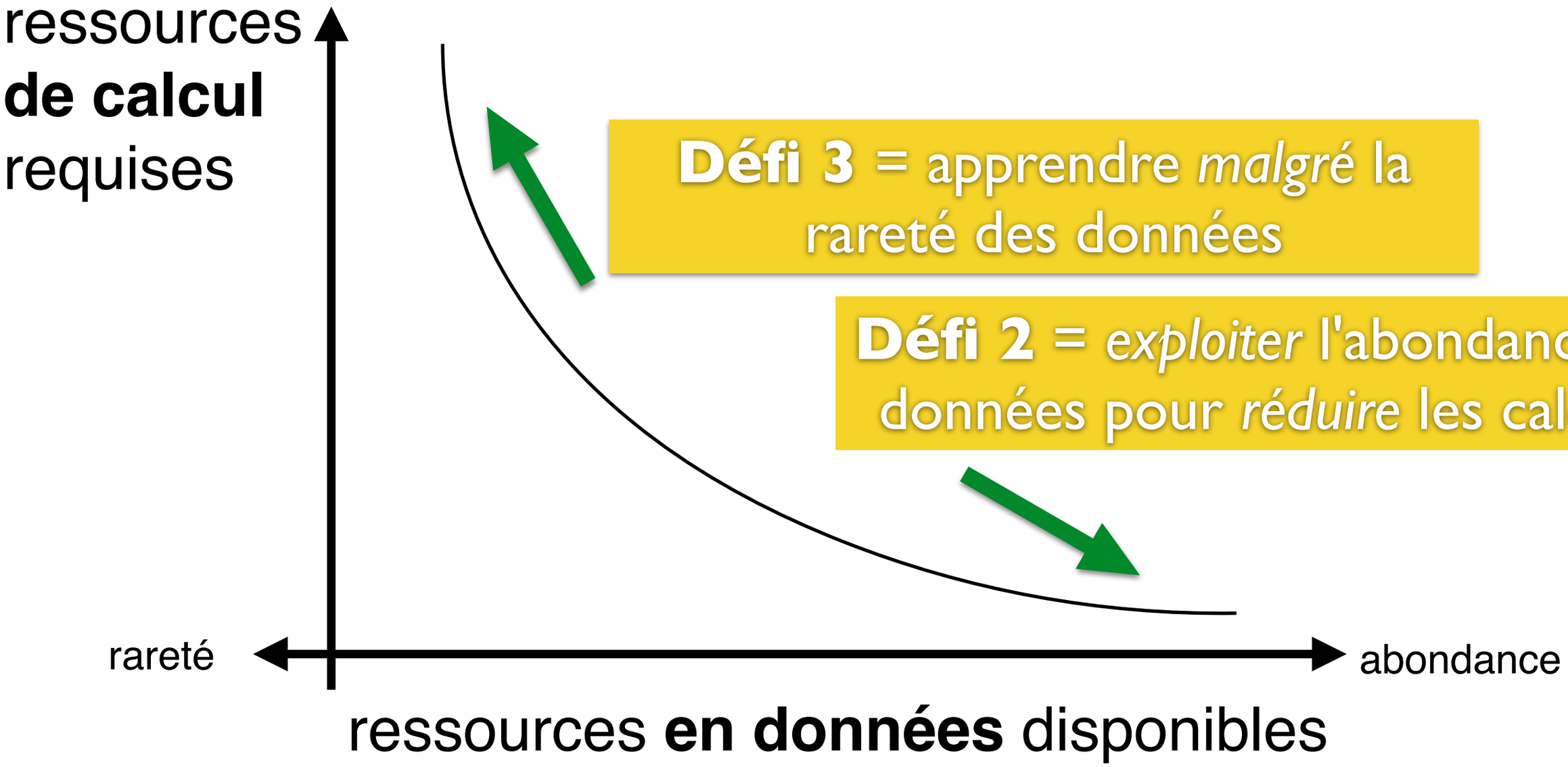


Défi 2 = *exploiter* l'abondance de données pour *réduire* les calculs

Compromis données / calcul

[1] S. Shalev-Shwartz and N. Srebro. SVM optimization: inverse dependence on training set size. In *Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008)*, pages 928–935, Helsinki, Finland, 2008. ACM.

- **Pour atteindre une performance cible *fixée***



Défi 3 = apprendre *malgré* la rareté des données

Défi 2 = *exploiter* l'abondance de données pour *réduire* les calculs

La parcimonie une clé pour la frugalité ?

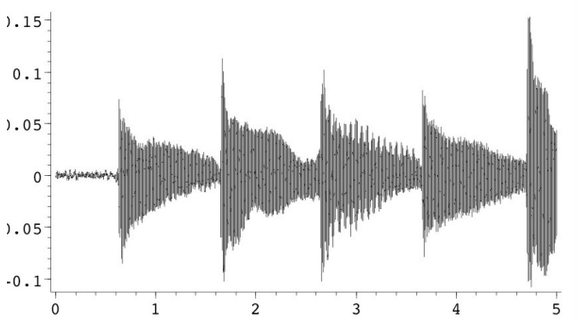
Un concept central : la parcimonie

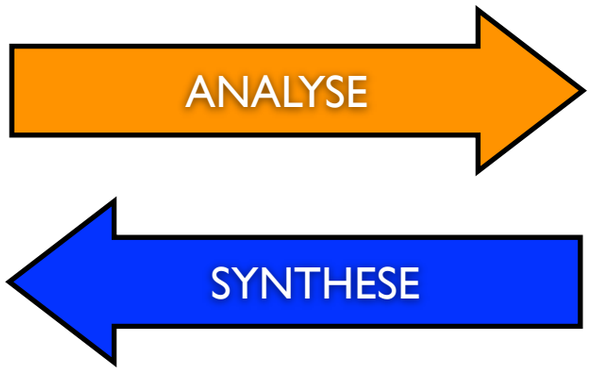
- **Parcimonie = la plupart des éléments sont nuls (dans un vecteur ou une matrice)**
  - On retrouve ce concept :
    - dans l'expression des *données* (variables explicatives)
    - dans l'expression des *paramètres* d'une méthode (degrés de liberté)
  - Incarnation numérique du rasoir d'Ockham / Occam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

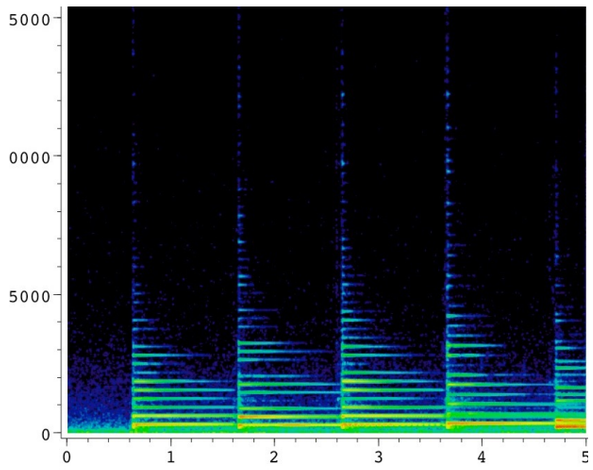
Pluralitas non est ponenda sine necessitate

La parcimonie - un don de la nature

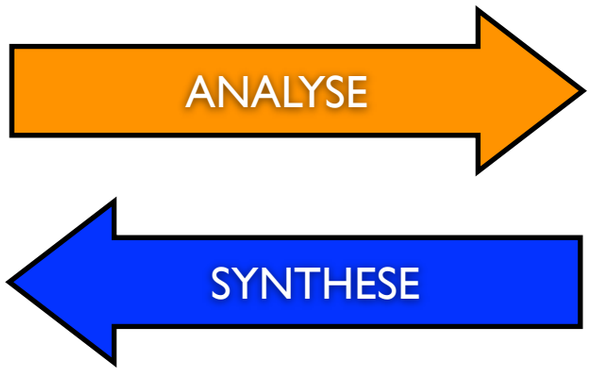
- Audio : représentations temps-fréquence

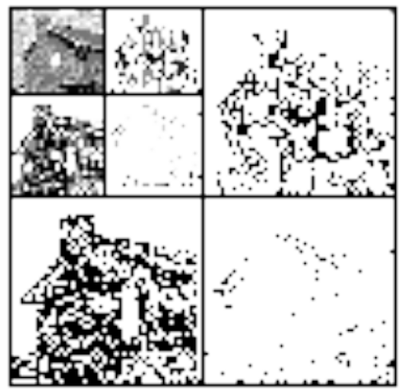






- Images : transformée en ondelettes

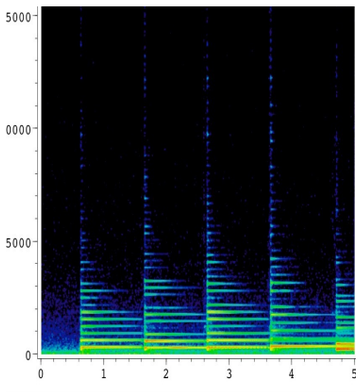




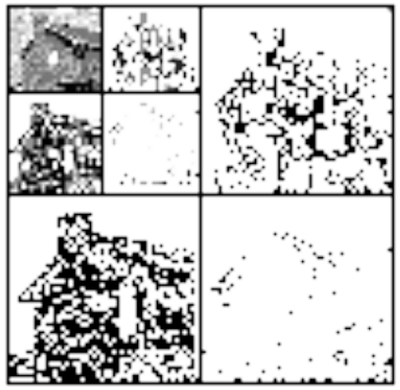
Parcimonie & frugalité

- Parcimonie comme un **objectif naturel**:
  - *bits*
    - *ex: compression*

MP3, AAC



JPEG



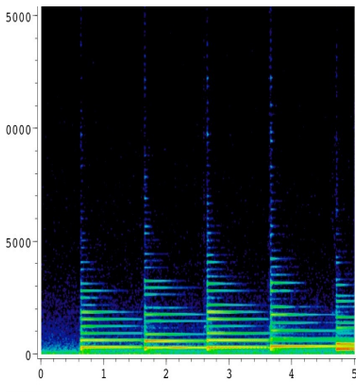
via ***vecteur creux (= parcimonieux)***

Parcimonie & frugalité

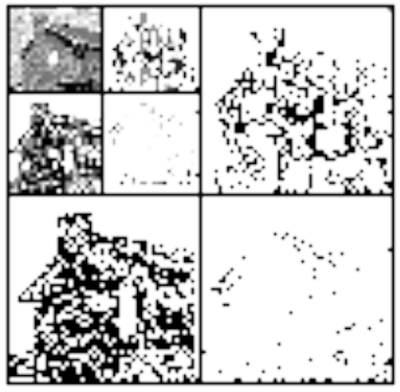
- Parcimonie comme un **objectif naturel**:
  - *bits*
    - *ex: compression*
  - *flops*
    - *ex: multiplication matricielle*

$$\mathbf{z} = \mathbf{\Psi x}$$

MP3, AAC



JPEG



- coût générique en dim N: $O(N^2)$
- transformées rapides $O(N)$ ou $O(N \log N)$

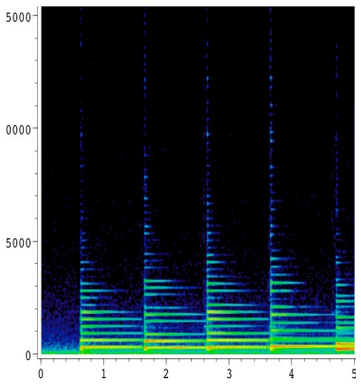
via ***vecteur creux (= parcimonieux)***

Parcimonie & frugalité

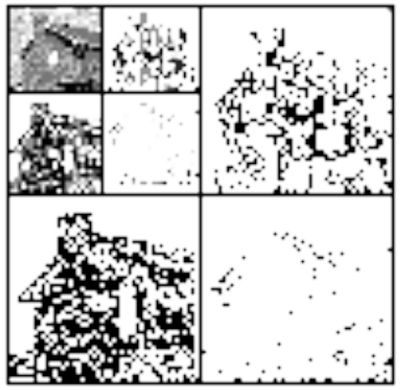
- Parcimonie comme un **objectif naturel**:
  - *bits*
    - *ex: compression*
  - *flops*
    - *ex: multiplication matricielle*

$$\mathbf{z} = \mathbf{\Psi x}$$

MP3, AAC

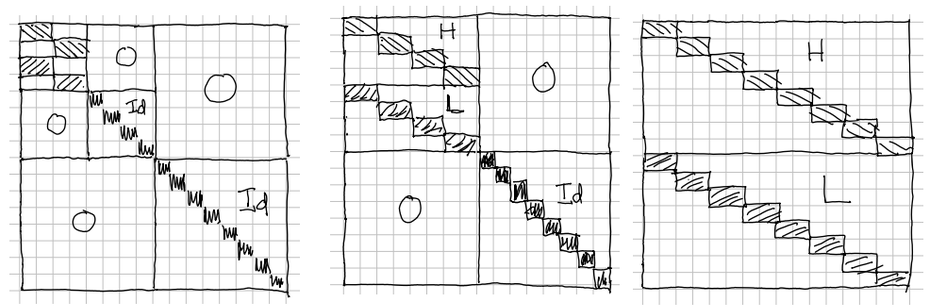


JPEG



- coût générique en dim N: $O(N^2)$
- transformées rapides $O(N)$ ou $O(N \log N)$

$\mathbf{\Psi} =$



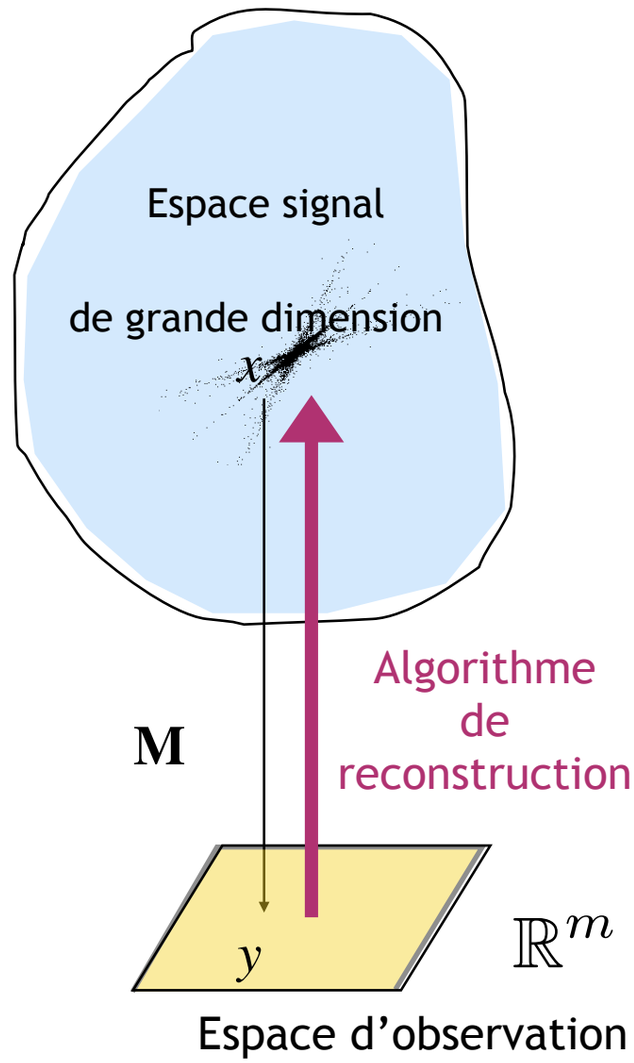
via ***vecteur creux (= parcimonieux)***

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

Parcimonie : histoire et évolution

A l'origine : problèmes inverses

- **But** = reconstruire à partir d'*observations partielles*
  - Restauration de données (paquets manquants …)
  - Imagerie (tomographie, inpainting…)
    - *Exemple* : défloutage en microscopie

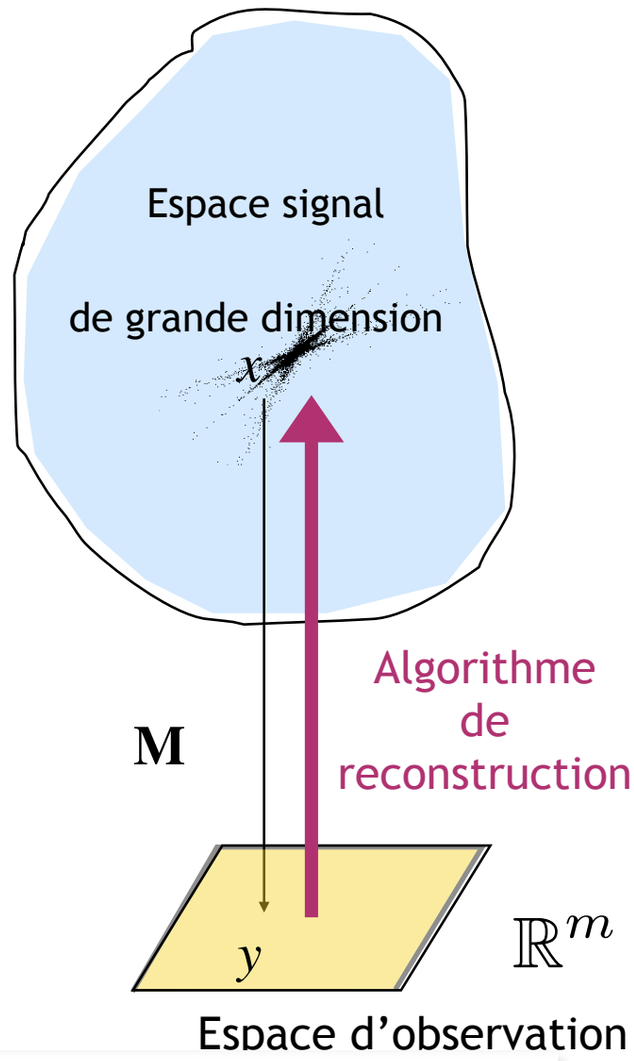


- **Approche classique**
  - Modèle "parcimonieux" : hypothèse nécessaire
    - + variantes "modèle simple / de faible complexité / de faible dimension"
  - Algorithmes de complexité bornée munis de solides garanties
    - typiquement : algorithmes gloutons, optimisation convexe

Parcimonie : histoire et évolution

A l'origine : problèmes inverses

- **But** = reconstruire à partir d'*observations partielles*
  - Restauration de données (paquets manquants …)
  - Imagerie (tomographie, inpainting…)
    - *Exemple* : défloutage en microscopie



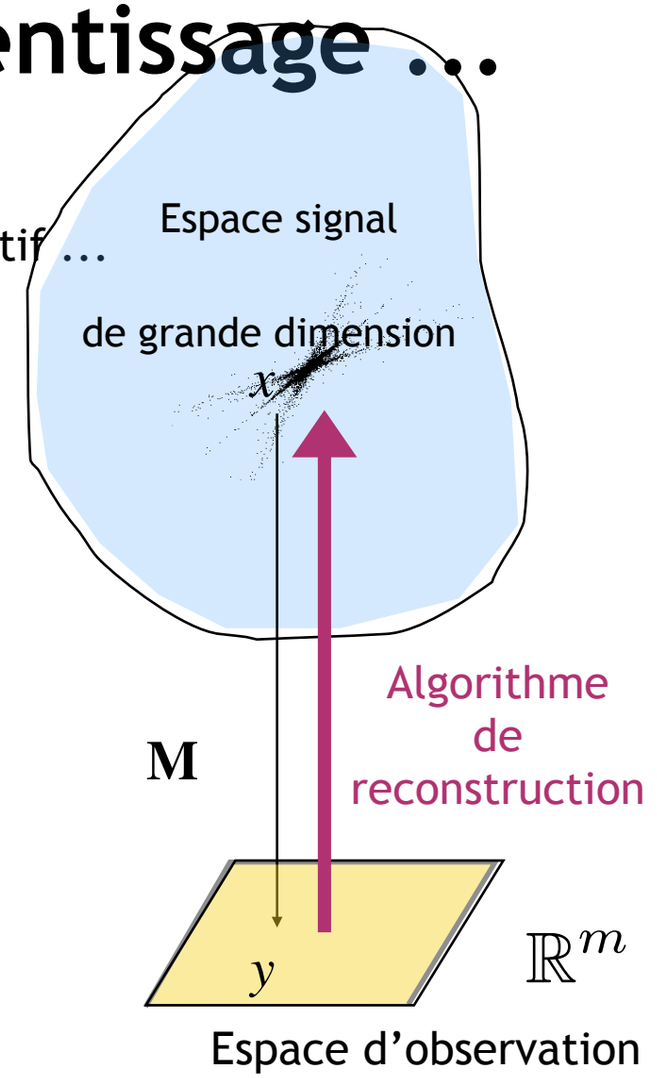
- **Approche classique**
  - Modèle "parcimonieux" : hypothèse nécessaire
    - + variantes "modèle simple / de faible complexité / de faible dimension"
  - Algorithmes de complexité bornée munis de solides garanties
    - typiquement : algorithmes gloutons, optimisation convexe

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

Parcimonie : histoire et évolution

- **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 $\mathbf{M}$
    - ***Compressive sensing*** via projections aléatoires
    - Extension à l'apprentissage: c***ompressive learning***
      - via *sketching* et *random features*

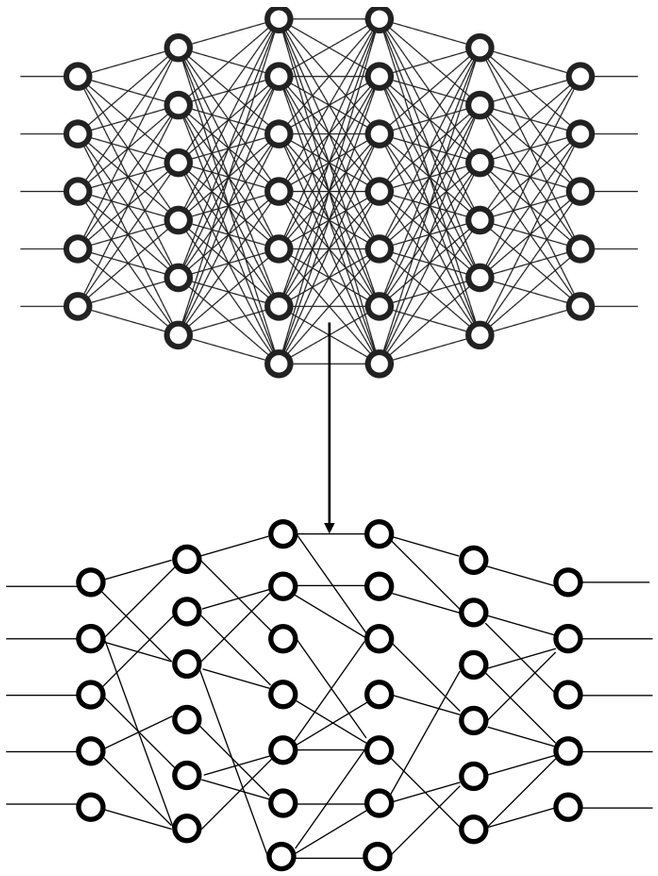


Parcimonie et apprentissage (profond)

Parcimonie et apprentissage (profond) ?

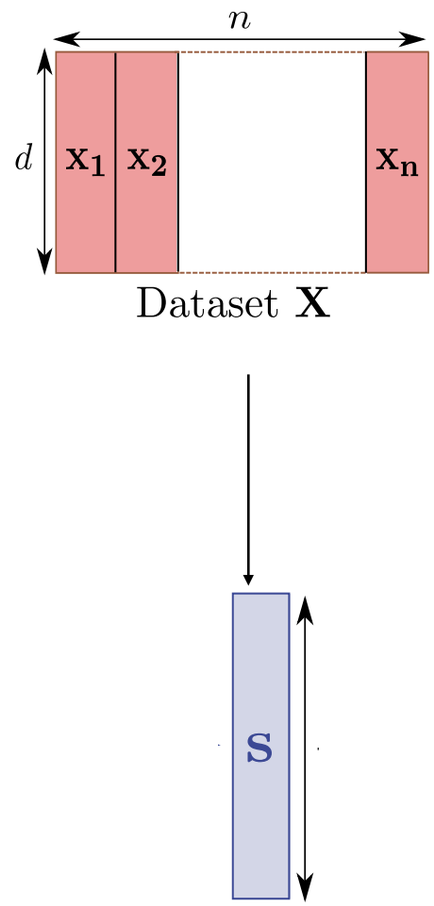
Réseaux parcimonieux

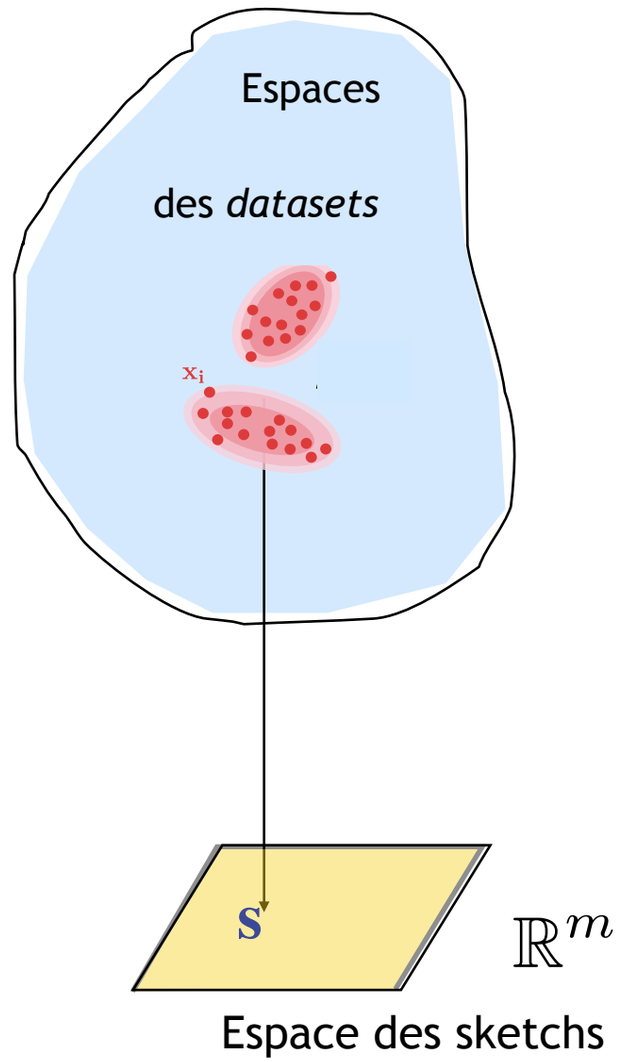
- la parcimonie comme objectif



Apprentissage compressif

- la parcimonie comme connaissance

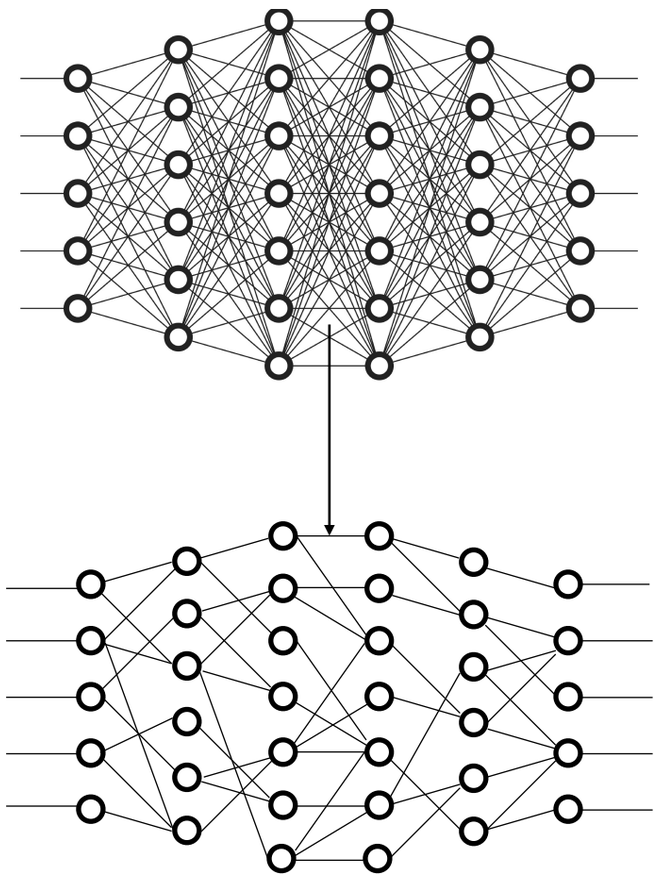




Parcimonie et apprentissage (profond) ?

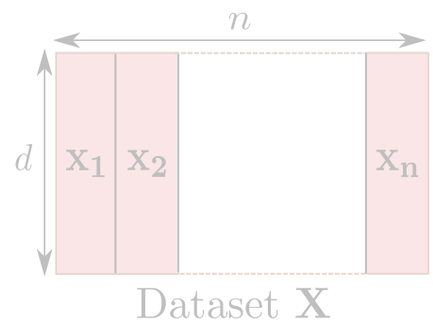
Réseaux parcimonieux

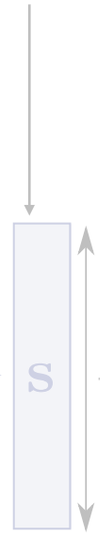
- la parcimonie comme objectif

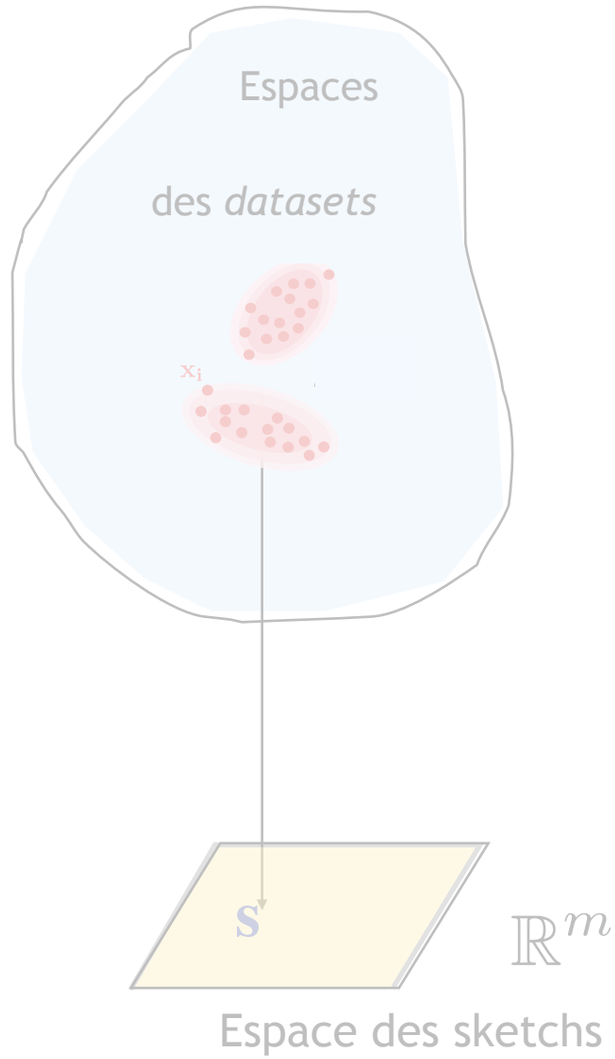


Apprentissage compressif

- la parcimonie comme connaissance



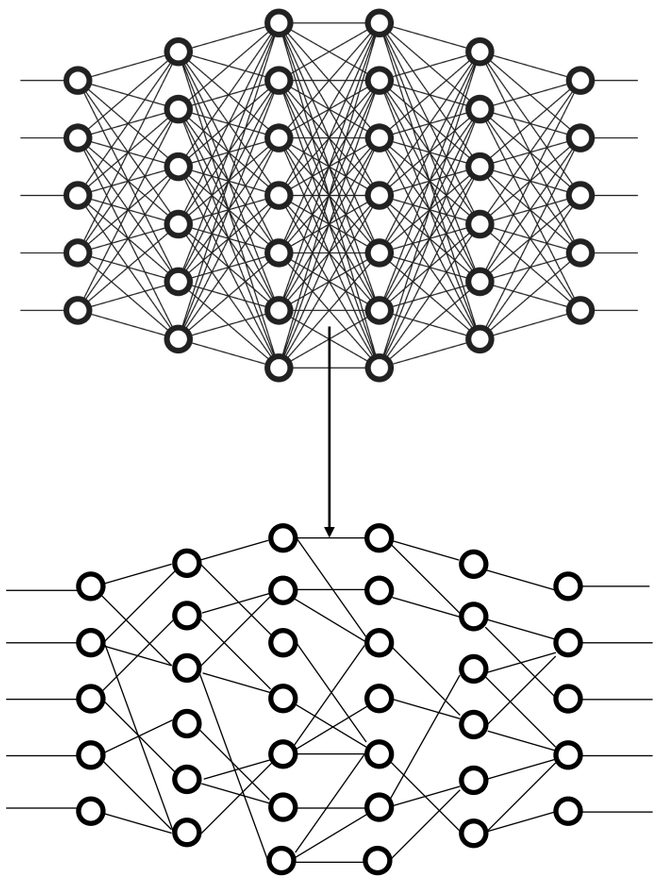




Parcimonie et apprentissage (profond) ?

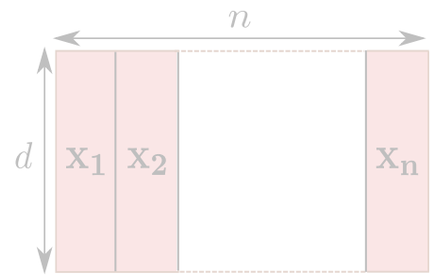
Réseaux parcimonieux

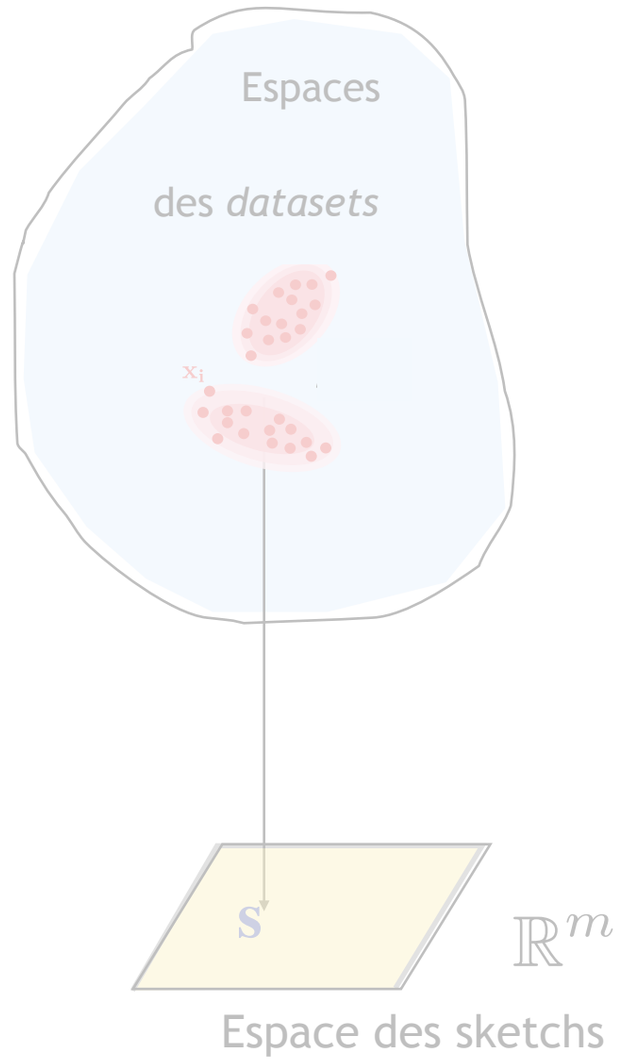
- la parcimonie comme objectif
  - élagage (ou croissance d'architecture !)
  - quantification
  - couches denses → structurées
    - rang-faible, tenseurs, papillons



Apprentissage compressif

- la parcimonie comme connaissance





Apprentissage et optimisation

Exemple : réseau ReLU à une couche cachée

- Paramètres = poids = $\theta = (\mathbf{W}_1, \mathbf{W}_2)$, fonction $f_\theta(x) = \mathbf{W}_2 \mathrm{ReLU}(\mathbf{W}_1 x)$
- Entrainement 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 à une couche cachée

- Paramètres = poids = $\theta = (\mathbf{W}_1, \mathbf{W}_2)$, fonction $f_\theta(x) = \mathbf{W}_2 \mathrm{ReLU}(\mathbf{W}_1 x)$
- Entrainement 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 $\mathbf{W}_i$ 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 à une couche cachée

- Paramètres = poids = $\theta = (\mathbf{W}_1, \mathbf{W}_2)$, fonction $f_\theta(x) = \mathbf{W}_2 \mathrm{ReLU}(\mathbf{W}_1 x)$
- Entrainement sur données (x_i, y_i) = problème d'optimisation

$$\min_\theta \sum_i (y_i - f_\theta(x_i))^2 \qquad \text{s.t. } \mathrm{supp}(\theta) \subset I$$

Apprentissage avec parcimonie "dure" ?

- Imposer que les matrices $\mathbf{W}_i$ 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 à une couche cachée

- Paramètres = poids = $\theta = (\mathbf{W}_1, \mathbf{W}_2)$, fonction $f_\theta(x) = \mathbf{W}_2 \mathrm{ReLU}(\mathbf{W}_1 x)$
- Entrainement sur données (x_i, y_i) = problème d'optimisation

$$\inf_\theta \cancel{\min_\theta} \sum_i (y_i - f_\theta(x_i))^2 \qquad \text{s.t. } \mathrm{supp}(\theta) \subset I$$

Apprentissage avec parcimonie "dure" ?

- Imposer que les matrices $\mathbf{W}_i$ 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)


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 à une couche cachée

- Paramètres = poids = $\theta = (\mathbf{W}_1, \mathbf{W}_2)$, fonction $f_\theta(x) = \mathbf{W}_2 \mathrm{ReLU}(\mathbf{W}_1 x)$
- Entrainement sur données (x_i, y_i) = problème d'optimisation

$$\inf_\theta \cancel{\min_\theta} \sum_i (y_i - f_\theta(x_i))^2 \qquad \text{s.t. } \mathrm{supp}(\theta) \subset I$$

Apprentissage avec parcimonie "dure" ?

- Imposer que les matrices $\mathbf{W}_i$ 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

Cela dépend-il du support ?

Oui mais ...

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


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

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

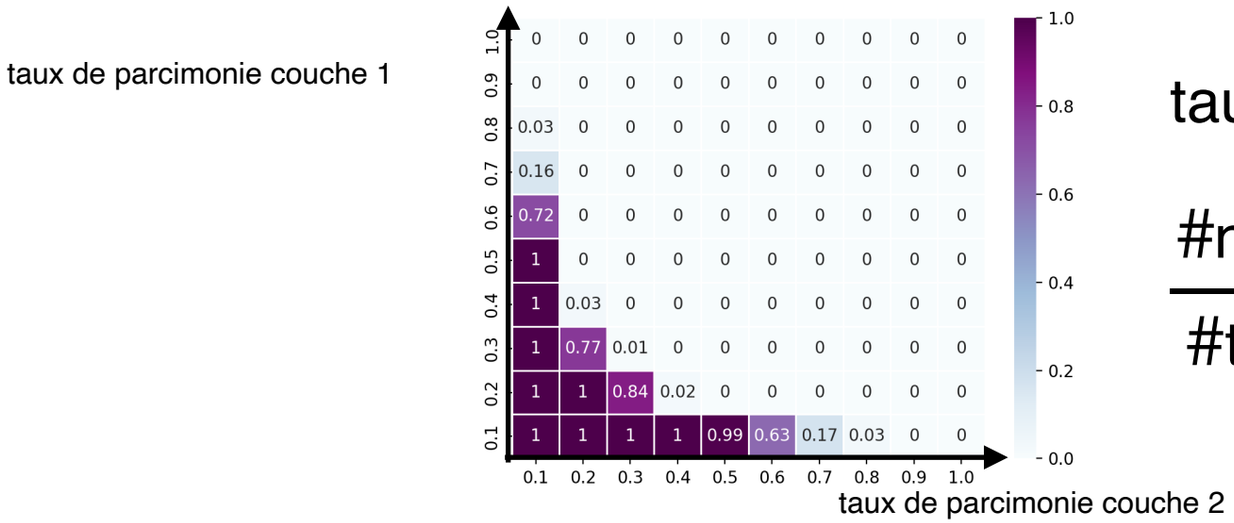

Does a sparse ReLU network training problem always admit an optimum?

Quoc-Tung Le Elisa Riccietti Rémi Gribonval

Les mauvais supports sont-ils courants ?

Experience

- Réseau à 2 couches $f_\theta(x) = \mathbf{W}_2 \mathrm{ReLU}(\mathbf{W}_1 x)$
- Supports aléatoires avec divers degrés de parcimonie
- Algorithme pour détecter les mauvais supports
  - Garantie: pas de faux positif (faux négatifs possibles)
  - Probabilité empirique de mauvais support



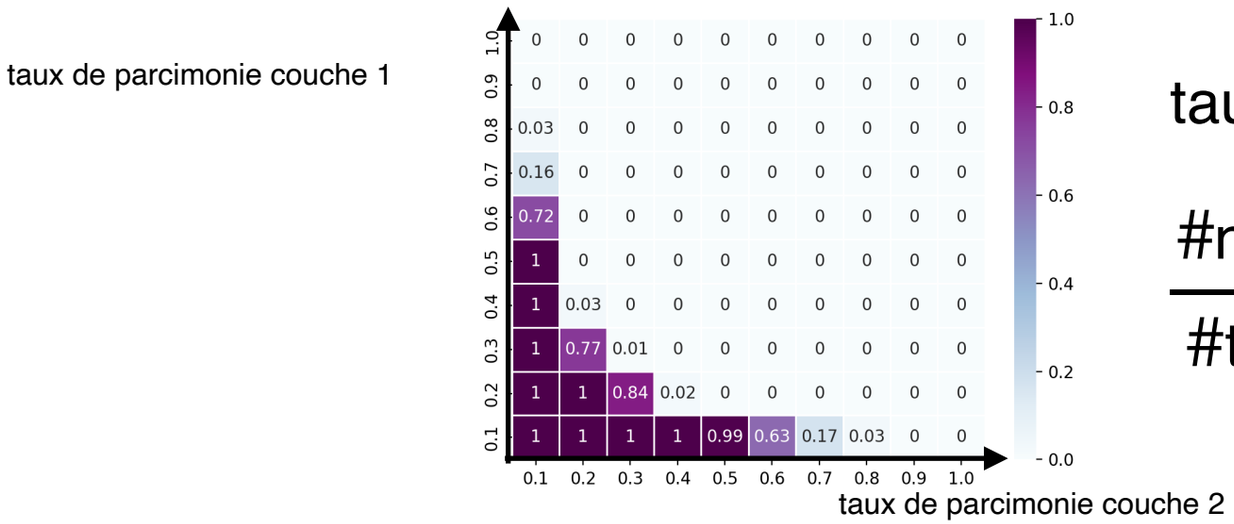
taux de parcimonie =

$$\frac{\#\text{nonzero entries}}{\#\text{total nb entries}}$$

Les mauvais supports sont-ils courants ?

Experience

- Réseau à 2 couches $f_\theta(x) = \mathbf{W}_2 \mathrm{ReLU}(\mathbf{W}_1 x)$
- Supports aléatoires avec divers degrés de parcimonie
- Algorithme pour détecter les mauvais supports
  - Garantie: pas de faux positif (faux négatifs possibles)
  - Probabilité empirique de mauvais support



taux de parcimonie =

$$\frac{\#\text{nonzero entries}}{\#\text{total nb entries}}$$

Deux alternatives:

- Régularisation $\min_\theta \sum_i (y_i - f_\theta(x_i))^2 \quad + \lambda \|\theta\|_p^p$
  - quelle norme Lp ?
- Restriction à de "bons" supports ?

Savoir faire parcimonieux ?

- La minimisation L^1 promeut la parcimonie
  - Basis Pursuit / Lasso
- La minimisation L^2 n'induit pas de parcimonie
  - Tikhonov / Ridge regression
- "Le support est la clé"
  - **Difficile** = trouver ***où*** sont les coefficients non nuls
  - **Facile** = trouver ***les valeurs*** des coefficients non nuls (moindres carrrés)
- Il suffit de seuiller
  - Algorithmes gloutons ou proximaux : détection des "grands coefficients"

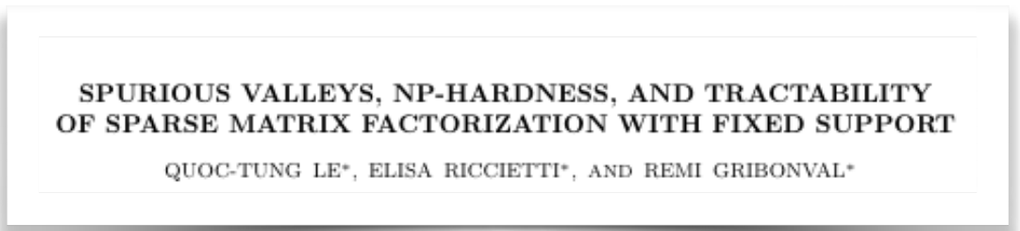
Savoir faire parcimonieux ?

- La minimisation L^1 promeut la parcimonie
  - Basis Pursuit / Lasso
- La minimisation L^2 n'induit pas de parcimonie
  - Tikhonov / Ridge regression
- "Le support est la clé"
  - **Difficile** = trouver ***où*** sont les coefficients non nuls
  - **Facile** = trouver ***les valeurs*** des coefficients non nuls (moindres carrrés)
- Il suffit de seuiller
  - Algorithmes gloutons ou proximaux : détection des "grands coefficients"

Effondrement en contexte **profond / multilinéaire**


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

Alexis Benichoux[1], Emmanuel Vincent[2], Rémi Gribonval[3]

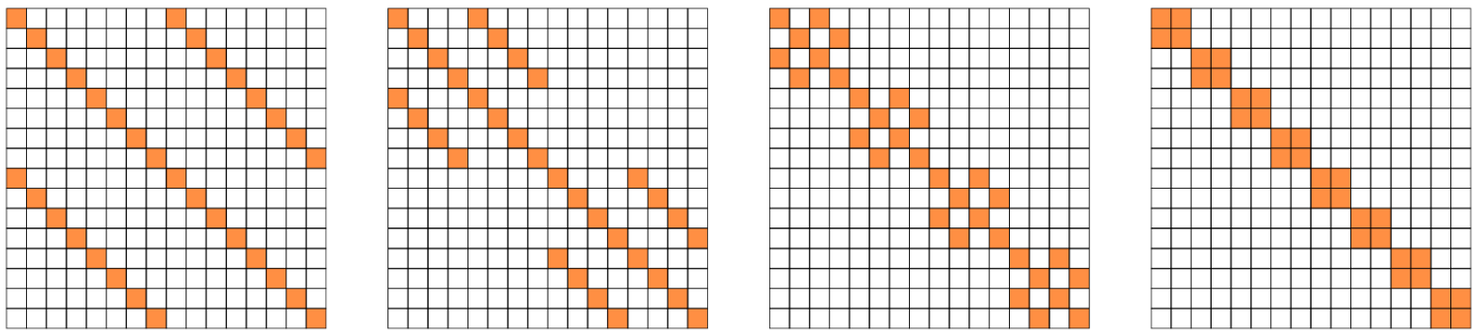

SPURIOUS VALLEYS, NP-HARDNESS, AND TRACTABILITY OF SPARSE MATRIX FACTORIZATION WITH FIXED SUPPORT

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

... mais fort potentiel de parcimonie *structurée*

Structure parcimonieuse "papillon"

Ubiquitaire pour transformées rapides (FFT ...)



(a) $\mathbf{S}_{\mathtt{bf}}^{(1)}$ (b) $\mathbf{S}_{\mathtt{bf}}^{(2)}$ (c) $\mathbf{S}_{\mathtt{bf}}^{(3)}$ (d) $\mathbf{S}_{\mathtt{bf}}^{(4)}$

- solide ancrage en analyse numérique (méthodes multipoles, matrices hiérarchiques)
- implémentation efficace & expressivité du modèle

Emergence rapide pour l'apprentissage profond

- pour remplacer couches denses des réseaux
- see e.g. [T. Dao & al, Learning Fast Algorithms for Linear Transforms Using Butterfly Factorizations, ICML, 2019]

Résultats récents sur les papillons

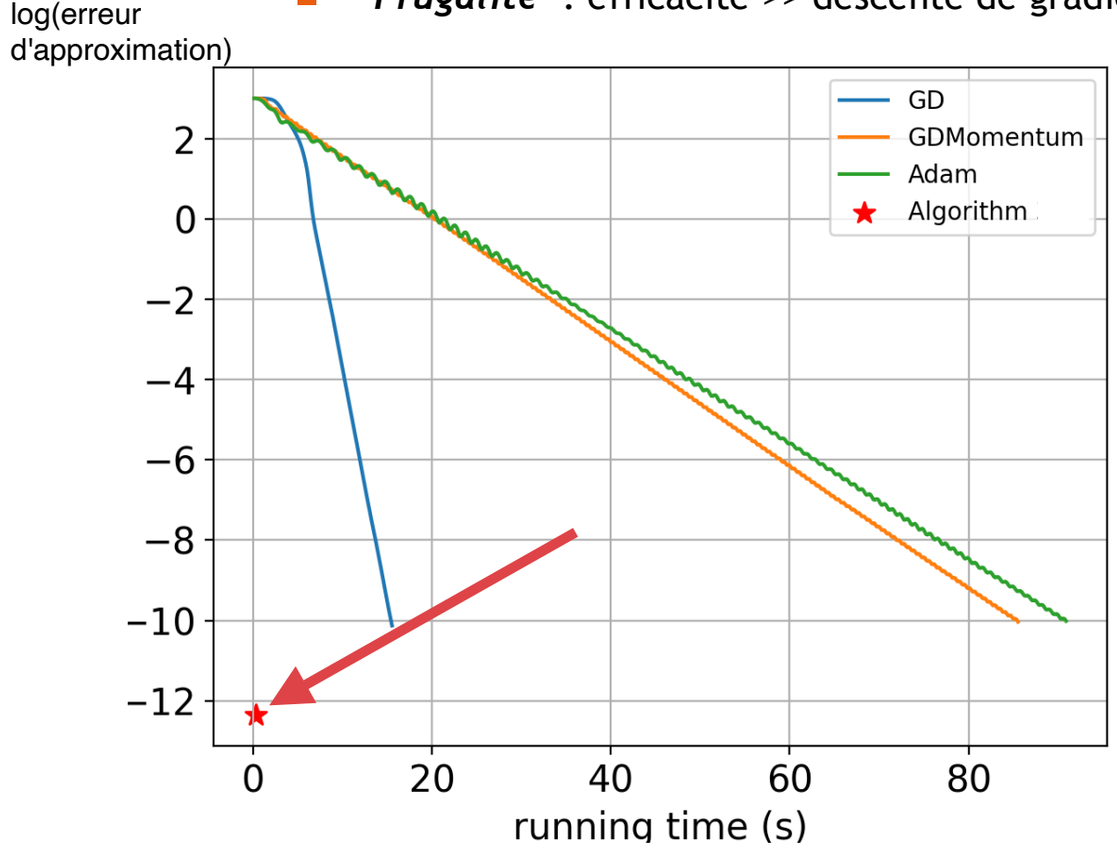
Quoc-Tung Le

Léon Zheng

Elisa Riccietti

Factorisation hiérarchique

- **Entrée** : une matrice (dense) $\mathbf{A}$
- **Sortie :** facteurs papillons $\mathbf{W}_\ell$
- **Approche** : un facteur à la fois
  - en ***exploitant la structure***
- **Résultats**
  - ***Identifiabilité*** (+en cours : stabilité au bruit)
  - ***Frugalité***" : efficacité >> descente de gradient

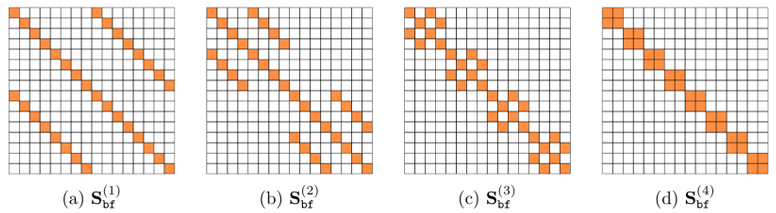


"réseau linéaire profond"

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

$$\min_{\mathbf{W}_\ell} \|\mathbf{A} - \Pi_{\ell=1}^{L} \mathbf{W}_\ell\|$$

avec facteurs papillons



(a) $\mathbf{S}_{\text{bf}}^{(1)}$ (b) $\mathbf{S}_{\text{bf}}^{(2)}$ (c) $\mathbf{S}_{\text{bf}}^{(3)}$ (d) $\mathbf{S}_{\text{bf}}^{(4)}$

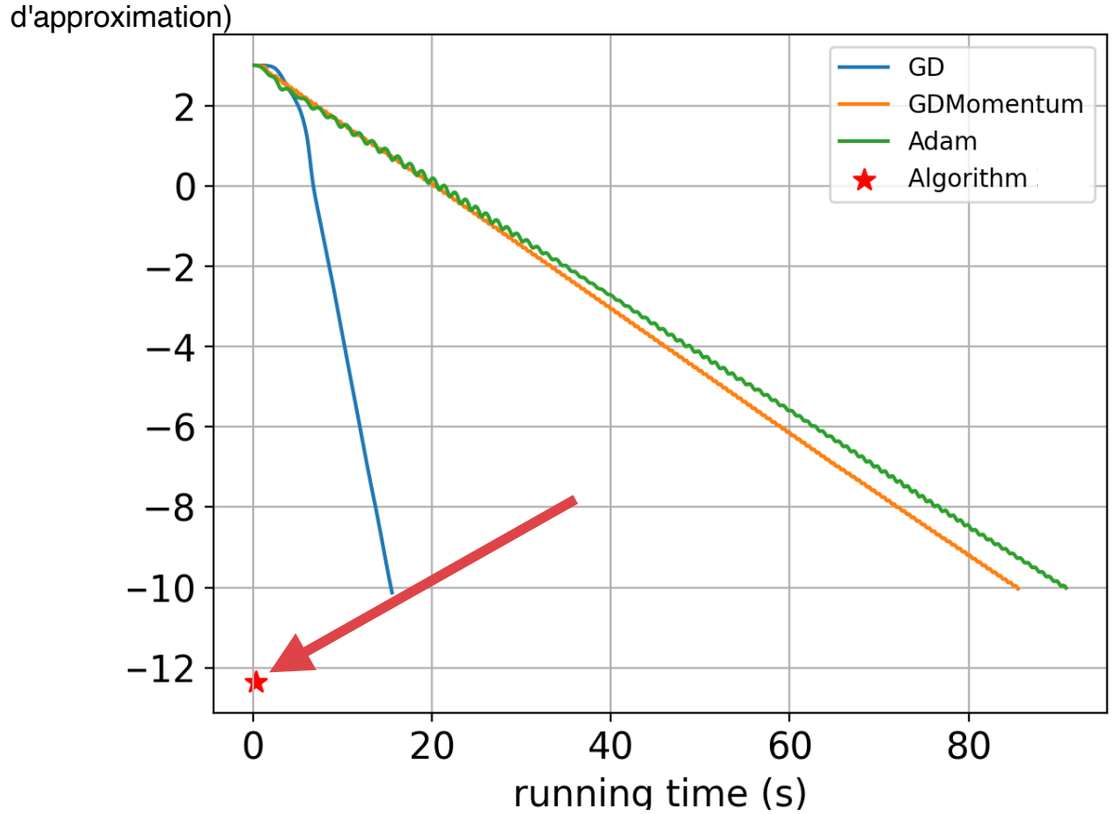
Résultats récents sur les papillons

Elisa Riccietti

Théo Mary

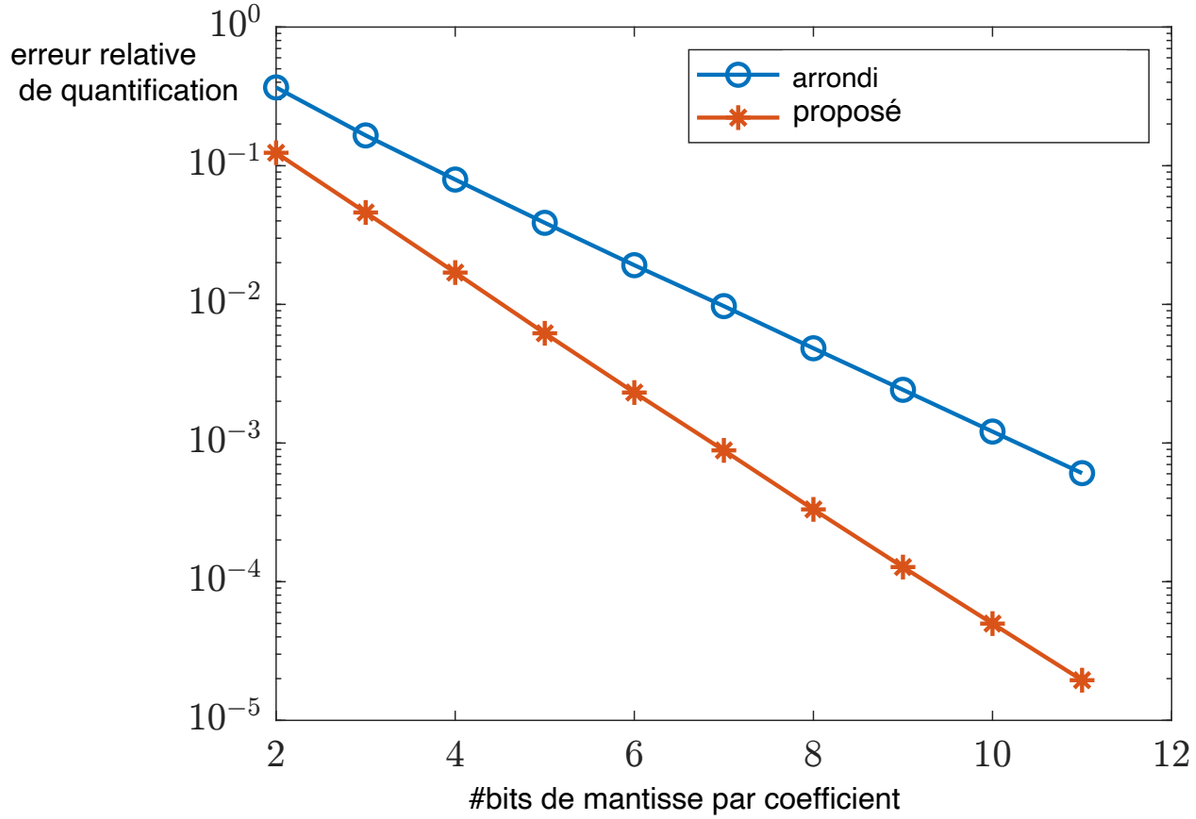
Factorisation hiérarchique

- **Entrée** : une matrice (dense) $\mathbf{A}$
- **Sortie** : facteurs papillons $\mathbf{W}_\ell$
- **Approche** : un facteur à la fois
  - en ***exploitant la structure***
- **Résultats**
  - ***Identifiabilité*** (+en cours : stabilité au bruit)
  - ***Frugalité***" : efficacité >> descente de gradient



Quantification efficace

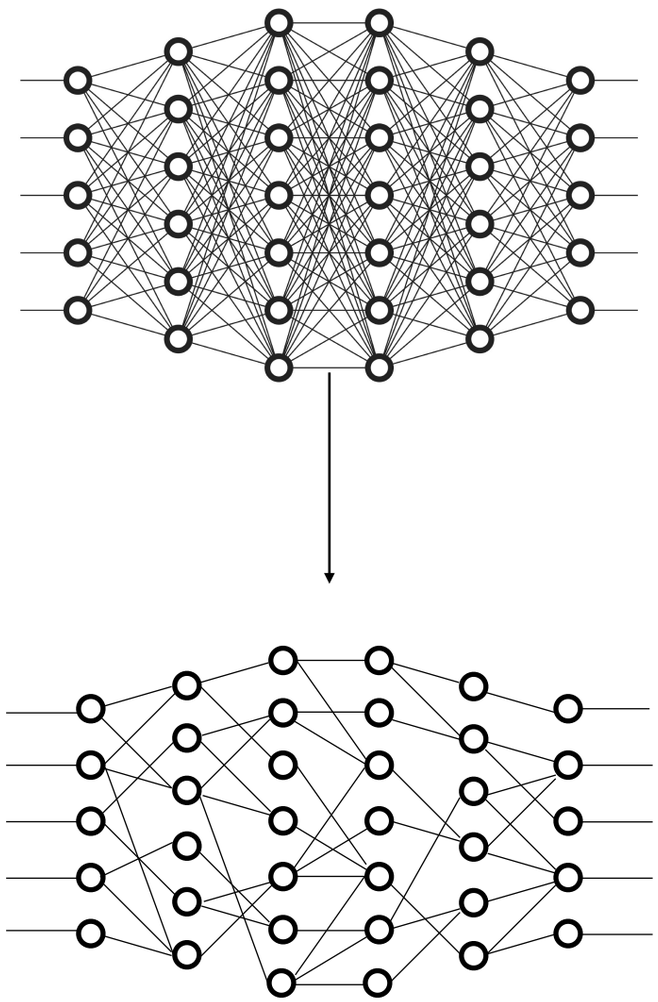
- **Entrée** : facteurs papillons $\mathbf{W}_\ell$
- **Sortie** : facteurs papillons *quantifiés*
- **Approche** : un facteur à la fois
  - en ***exploitant invariance par remise à l'échelle***
- **Résultats**
  - Algorithme **optimal** à deux facteurs
  - **-30% de bits** comparé à arrondi usuel



Parcimonie et apprentissage (profond) ?

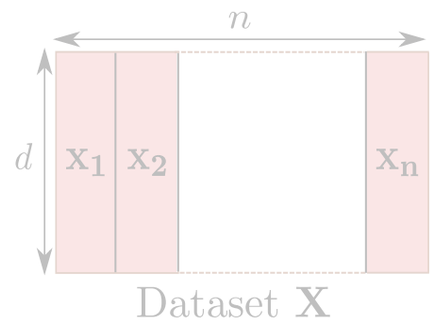
Réseaux parcimonieux

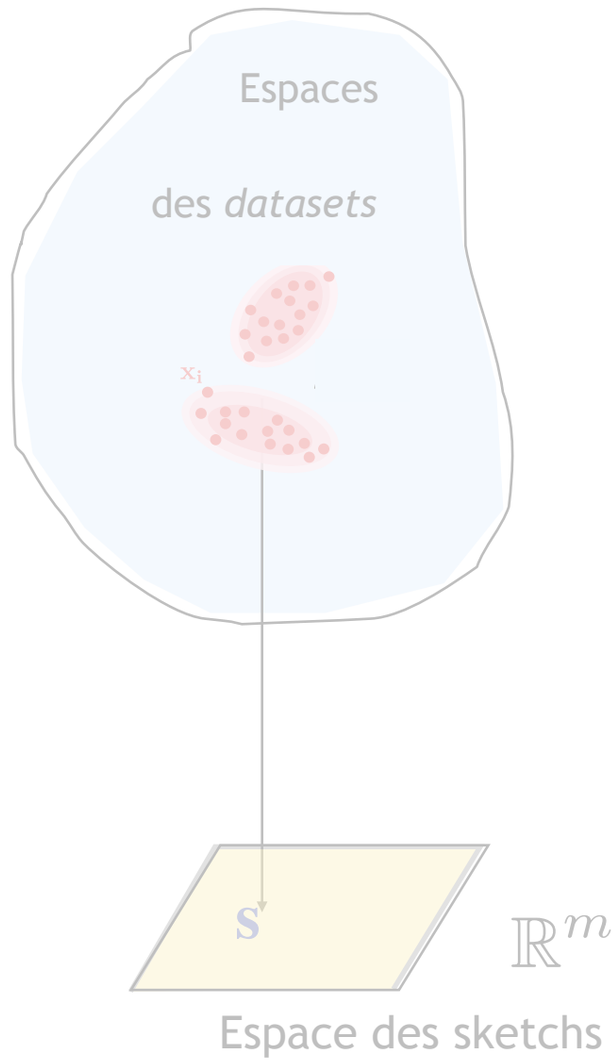
- la parcimonie comme objectif
  - + quantification des poids



Apprentissage compressif

- la parcimonie comme connaissance

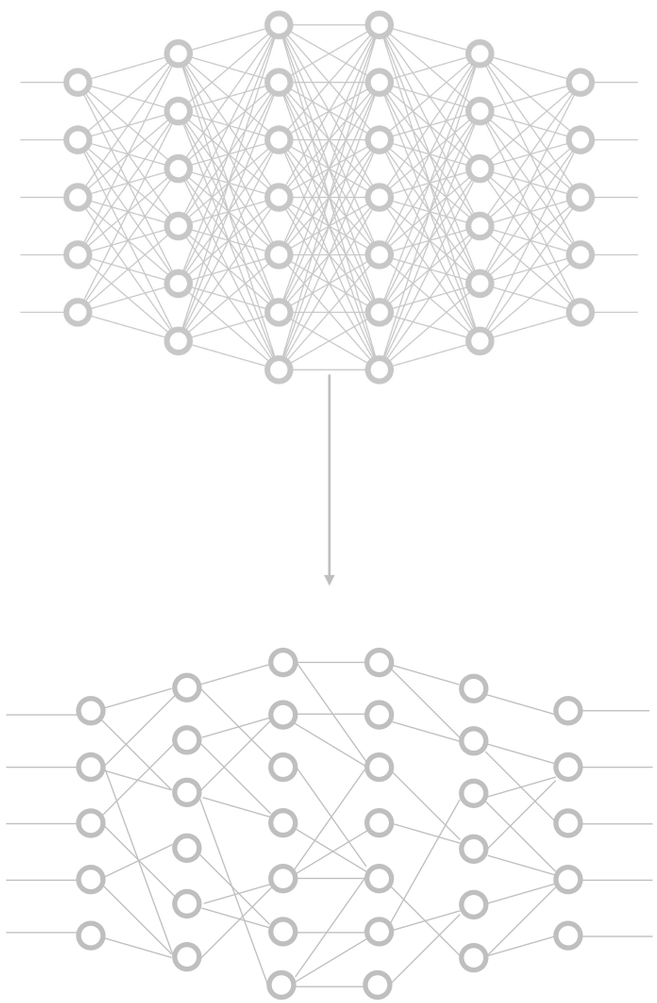




Parcimonie et apprentissage (profond) ?

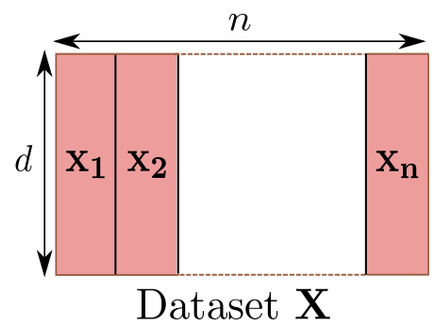
Réseaux parcimonieux

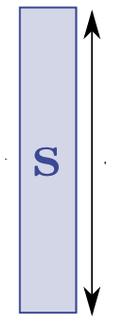
- la parcimonie comme objectif
  - + quantification des poids

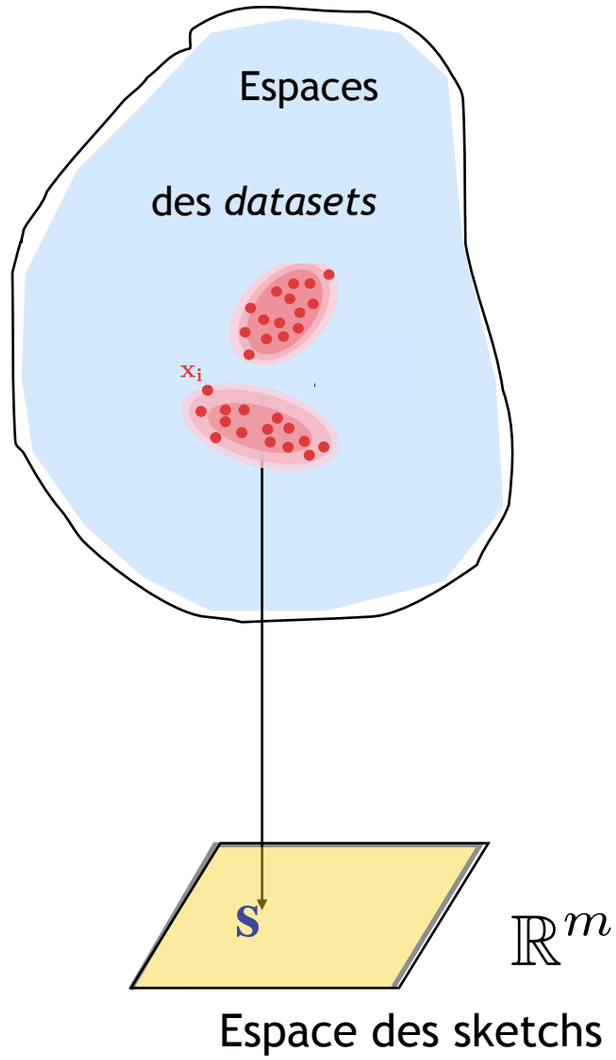


Apprentissage compressif

- la parcimonie comme connaissance

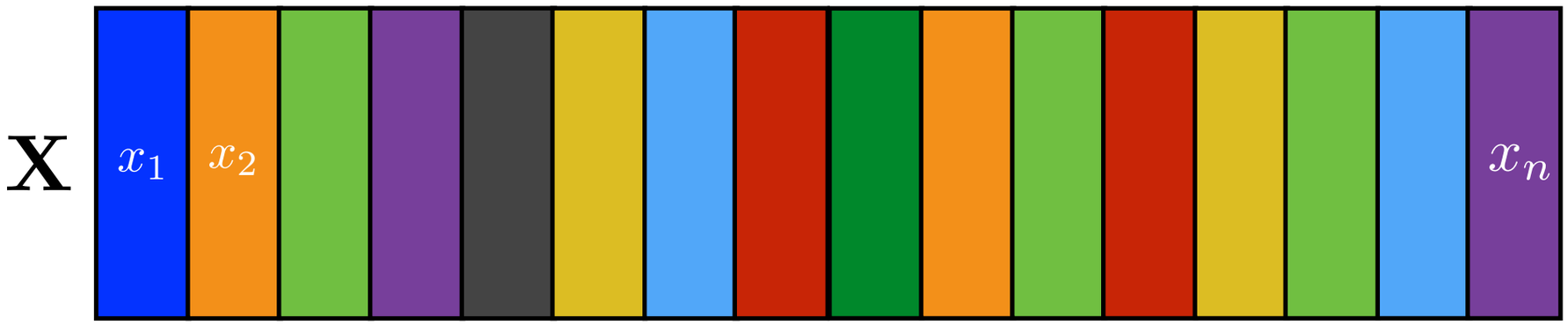




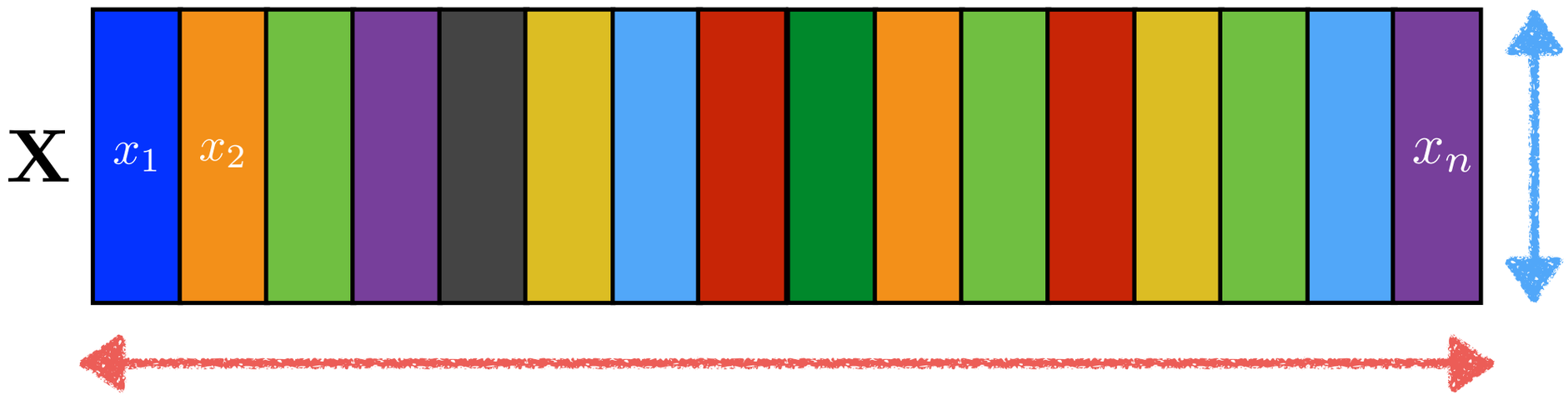


Apprentissage compressif (survol)

Large-scale learning

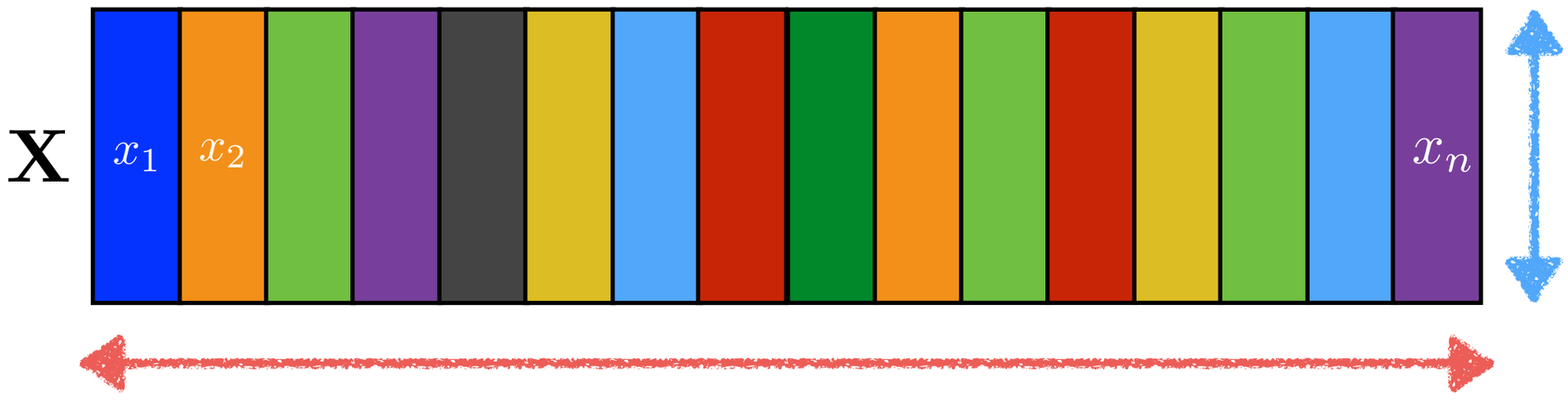


Large-scale learning



- High feature dimension *d*
- Large collection size *n* = *“volume”*

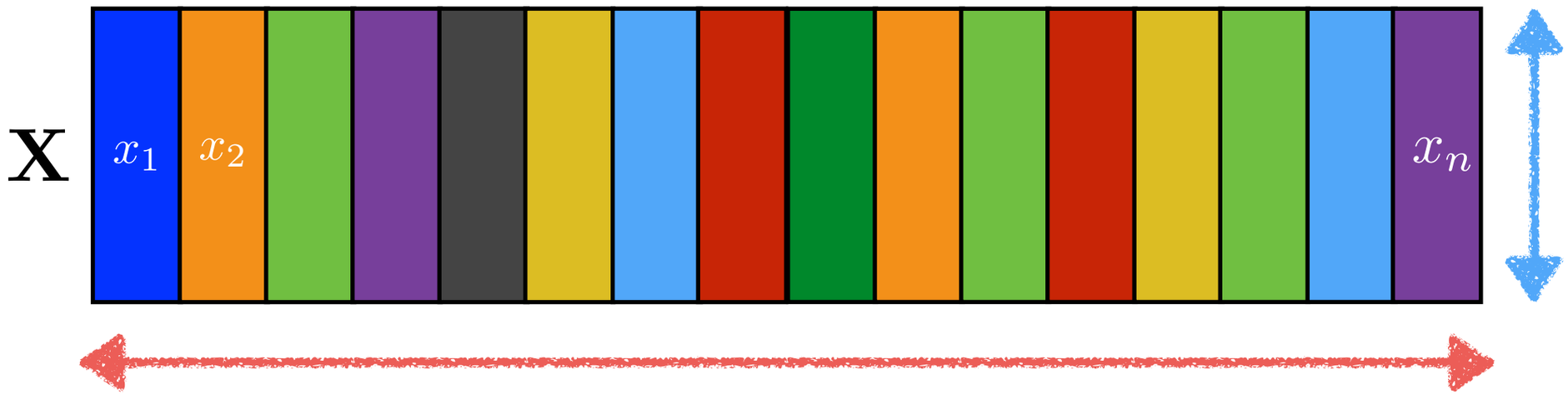
Large-scale learning



- High feature dimension *d*
- Large collection size *n* = *“volume”*

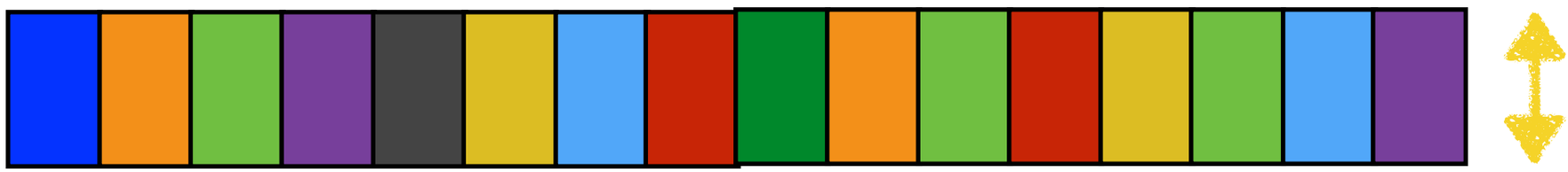
Challenge: compress $\mathcal{X}$ before learning ?

Compressive learning: three routes



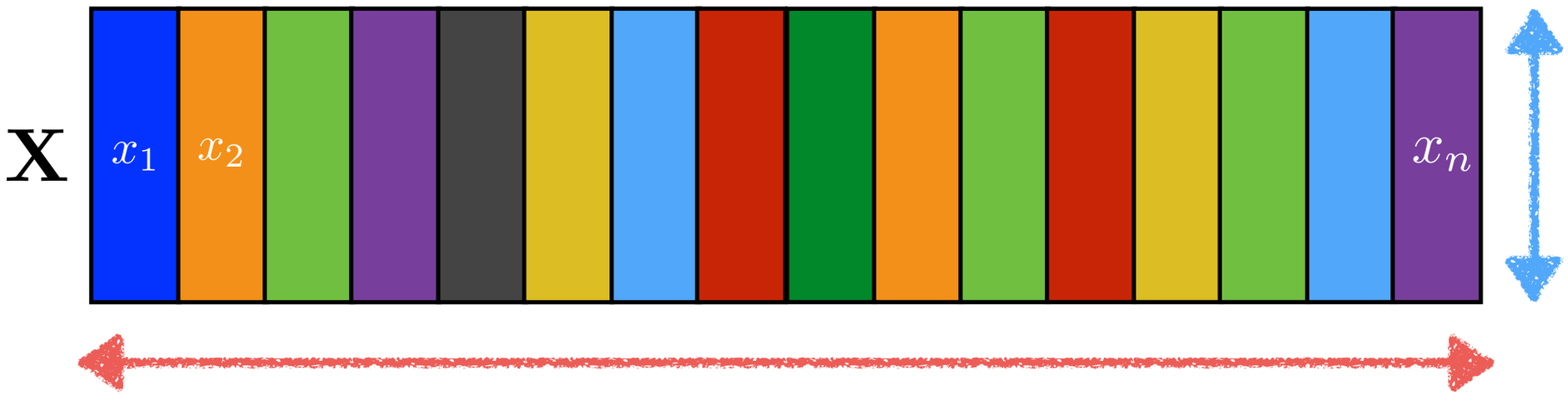
- dimension reduction
- subsampling
- **sketching**

$\mathbf{Y} = \mathbf{M}\mathbf{X}$

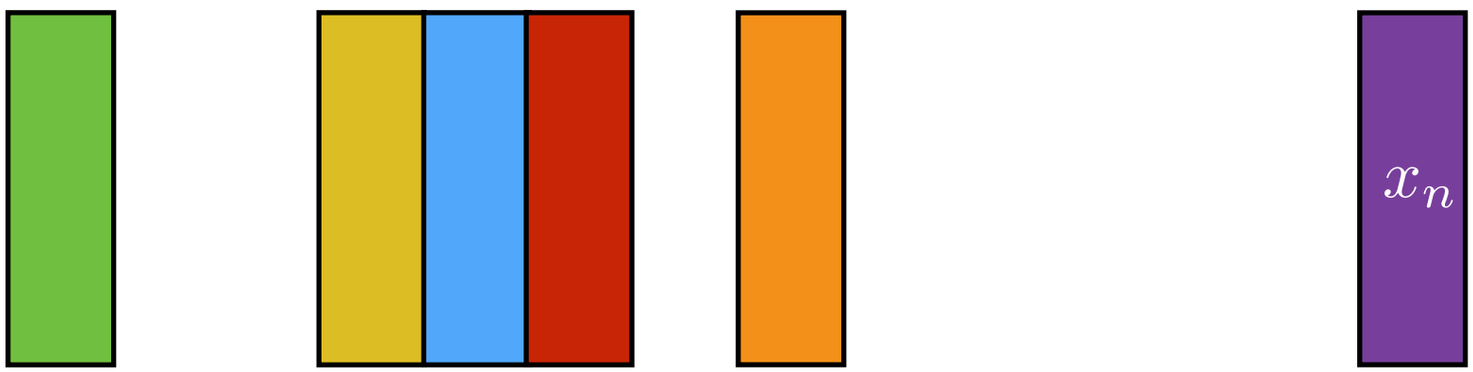


random projections - Johnson Lindenstrauss lemma
see e.g. [Calderbank & al 2009, Reboredo & al 2013]

Compressive learning: three routes



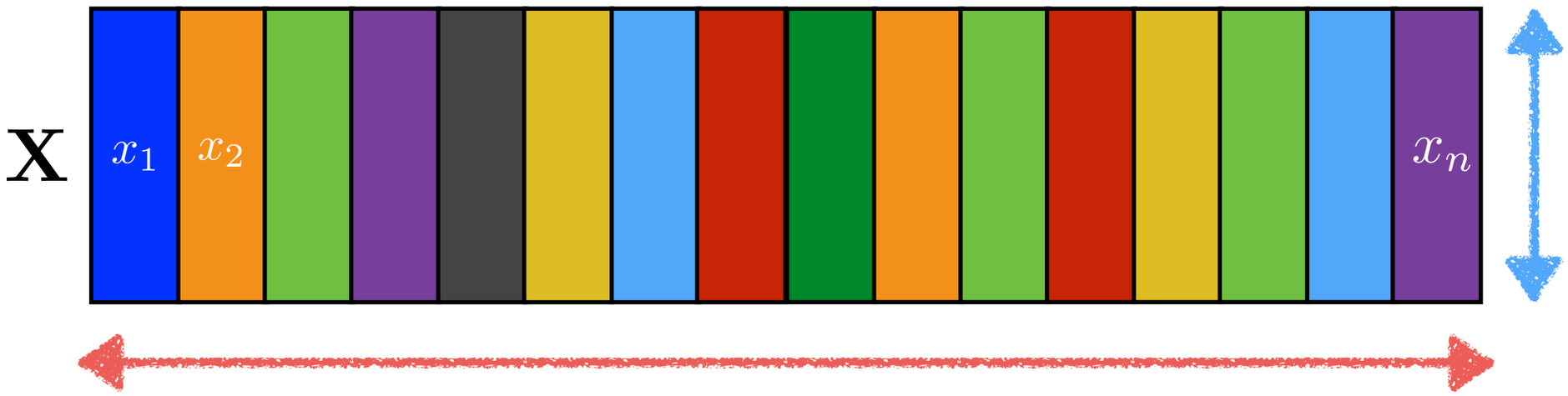
- dimension reduction
- subsampling
- **sketching**



Nyström method & coresets

see e.g. [Williams&Seeger 2000, Agarwal & al 2003, Felman 2010]

Compressive learning: three routes



- dimension reduction
- subsampling
- **sketching**

  with random moments

$$\begin{matrix}\mathbb{E}\Phi_1(X)\\ \dots \\ \mathbb{E}\Phi_m(X)\end{matrix} \quad z \in \mathbb{R}^m$$

Inspiration: *compressive sensing* *[Foucart & Rauhut 2013]*
sketching/hashing *[Thaper & al 2002, Cormode & al 2005]*
Connections with: *generalized method of moments* *[Hall 2005]*
kernel mean embeddings *[Smola & al 2007, Sriperimbudur & al 2010]*

Compressive Statistical Learning

learning task $\longrightarrow$ $\arg\min_{\theta} \mathbb{E}_X \ell(X, \theta)$

Compressive Statistical Learning

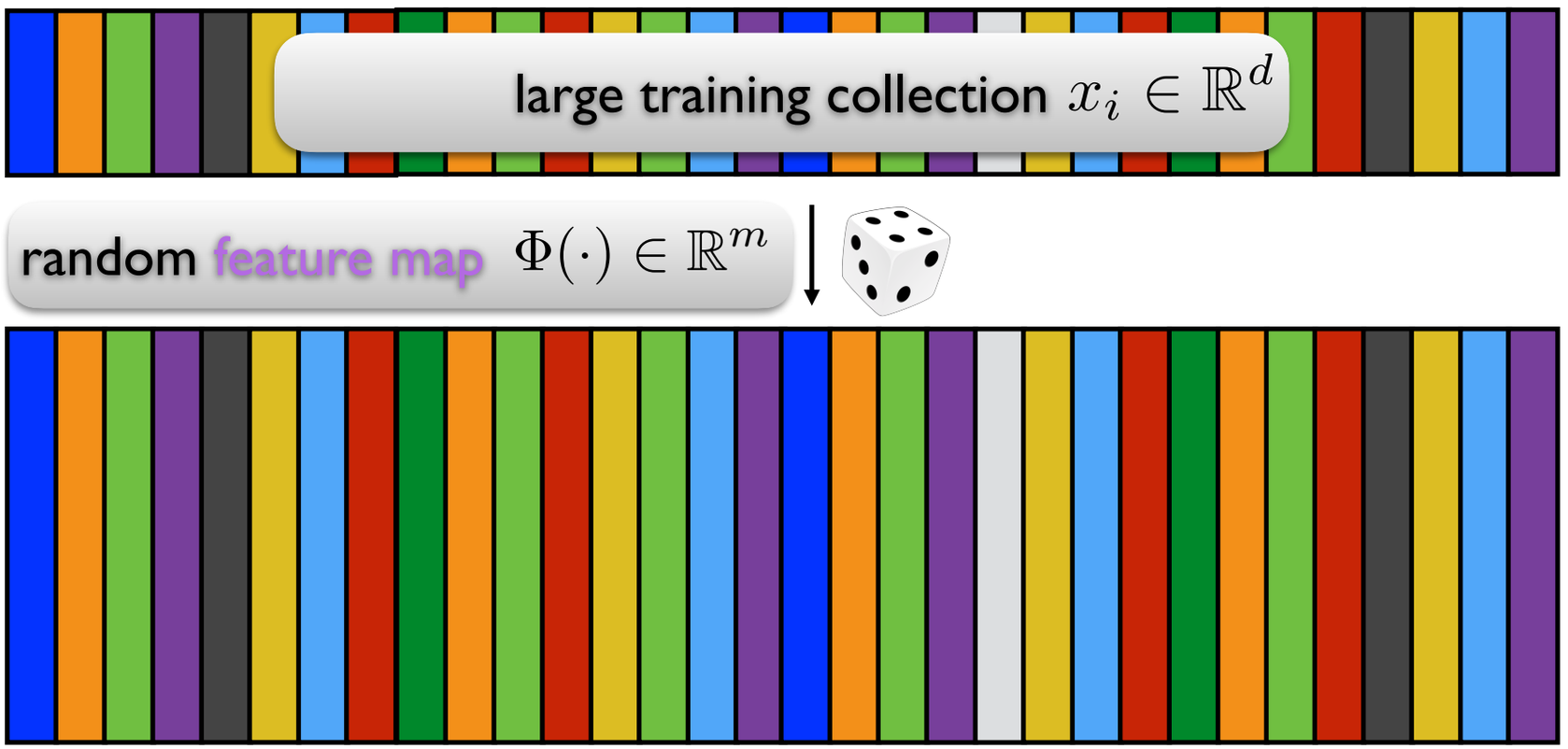
learning task $\longrightarrow \arg\min_\theta \mathbb{E}_X \ell(X, \theta)$

- **Traditional approach:**
  - (Convex) optimization
  - (Stochastic) gradient descent

$$\hat{\theta} \approx \arg\min_\theta \frac{1}{n} \sum_{i=1}^{n} \ell(x_i, \theta)$$

- **Several passes on the training set**
- **Resource hungry at large scale**

Compressive Statistical Learning



large training collection $x_i \in \mathbb{R}^d$

learning task $\longrightarrow \arg\min_\theta \mathbb{E}_X \ell(X, \theta)$

random feature map $\Phi(\cdot) \in \mathbb{R}^m$

empirical average

small sketch vector

$$z = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i) \in \mathbb{R}^m$$

Compressive Statistical Learning

large training collection $x_i \in \mathbb{R}^d$

learning task $\longrightarrow \arg\min_\theta \mathbb{E}_X \ell(X, \theta)$

random feature map $\Phi(\cdot) \in \mathbb{R}^m$

ex: random Fourier features

$$\Phi(x) = \rho(\mathbf{W}x)$$

scalar nonlinearity

random projection

empirical average

small sketch vector

$$z = \frac{1}{n}\sum_{i=1}^{n} \Phi(x_i) \in \mathbb{R}^m$$

Compressive Statistical Learning

large training collection $x_i \in \mathbb{R}^d$

learning task $\longrightarrow \arg\min_\theta \mathbb{E}_X \ell(X, \theta)$

random feature map $\Phi(\cdot) \in \mathbb{R}^m$

ex: random Fourier features

$$\Phi(x) = \rho(\mathbf{W}x)$$

scalar nonlinearity

random projection

empirical average

small sketch vector

$$z = \frac{1}{n}\sum_{i=1}^{n} \Phi(x_i) \in \mathbb{R}^m$$

- = Random feature moments
  - Small memory footprint
- *One pass* on the training set
- *Distributed / streamed* computation

Compressive Statistical Learning

learning task $\longrightarrow$ $\arg\min_\theta \mathbb{E}_X \ell(X, \theta)$

small sketch vector

$$z = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i) \in \mathbb{R}^m$$

- = Random feature moments
  - Small memory footprint
  - Privacy
- *One pass* on the training set
- *Distributed / streamed* computation

Compressive Statistical Learning

large training collection $x_i \in \mathbb{R}^d$

learning task → $\arg\min_\theta \mathbb{E}_X \ell(X, \theta)$

$$\hat{\theta} \approx \arg\min_\theta R(z, \theta)$$

Learning by moment fitting

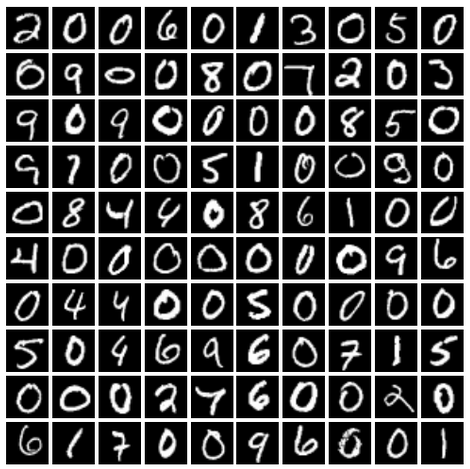
small sketch vector

$$z = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i) \in \mathbb{R}^m$$

- = Random feature moments
  - Small memory footprint
  - Privacy
- *One pass* on the training set
- *Distributed / streamed* computation

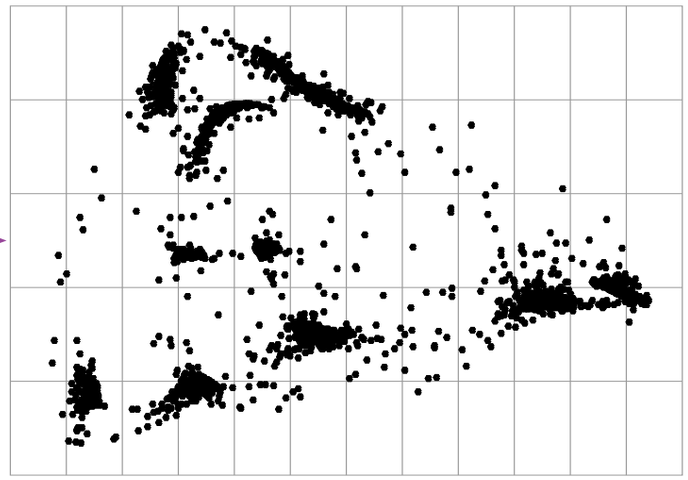
Example: clustering MNIST

Handwritten digits

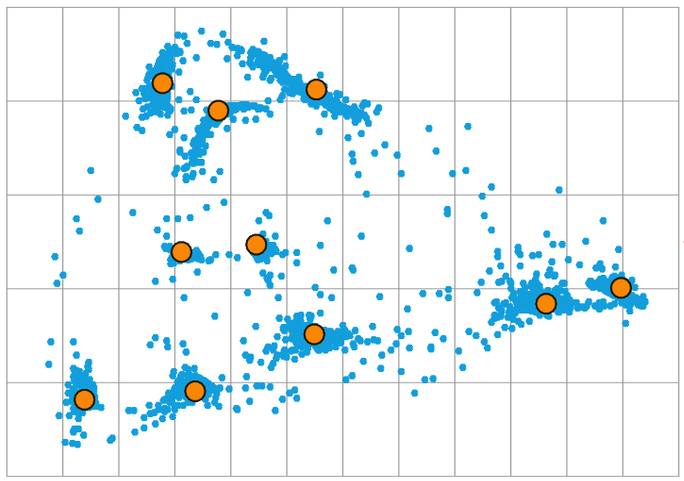


Pre-processing

Spectral embedding



n=70 000 points
d=10 dimension
k=10 clusters



k centroids, each of dimension d

Sketched Clustering

$\widetilde{z}$

Sketch

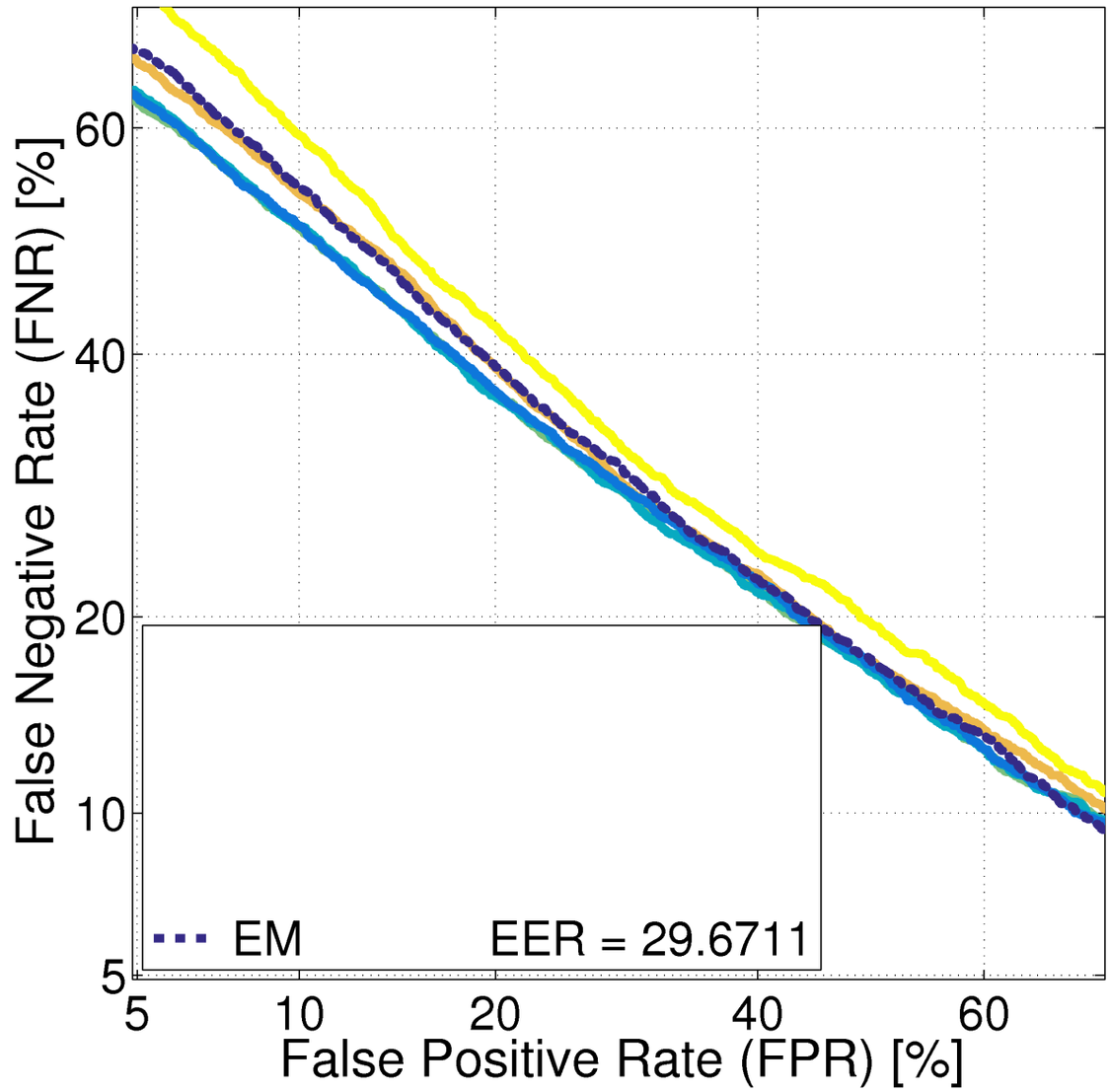
m ~ kd sketch dimension ~ 200

Proof of Concept: Speaker Verification with Compressive GMM-UBM

~ 50 Gbytes
~ 1000 hours of speech

$\mathcal{X}$

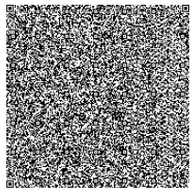
$k = 64; n_{\text{Sketch}} = 200 \times n_{\text{EM}} = 6.10^7$



Proof of Concept: Speaker Verification with Compressive GMM-UBM

~ 50 Gbytes
~ 1000 hours of speech

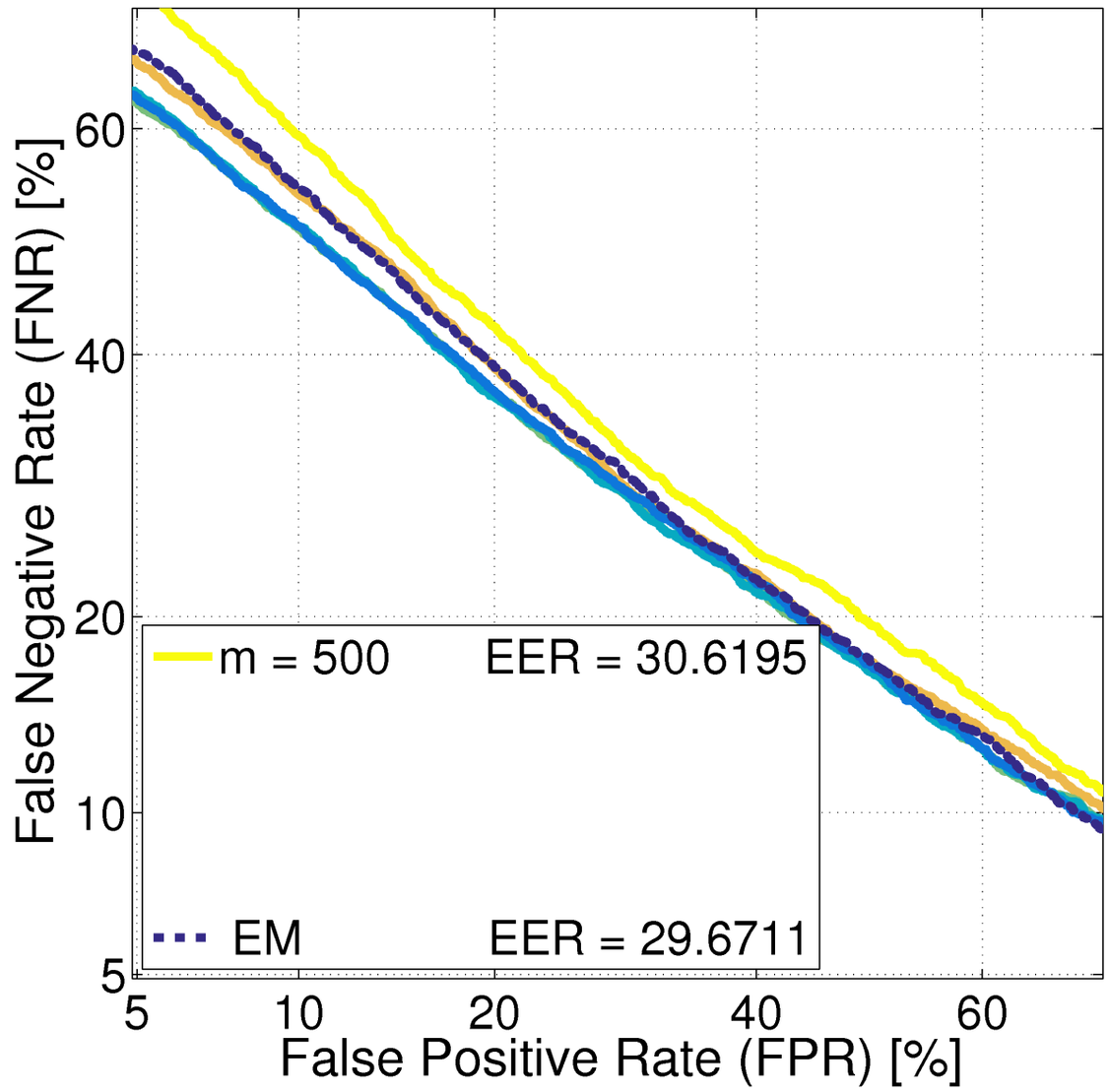
$\mathcal{X}$



- **m= 500**
  - 7 200 000-fold compression

$kd = 768; m/kd < 1$

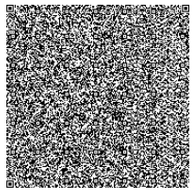
$k = 64; n_{\text{Sketch}} = 200 \times n_{\text{EM}} = 6.10^7$



Proof of Concept: Speaker Verification with Compressive GMM-UBM

~ 50 Gbytes
~ 1000 hours of speech

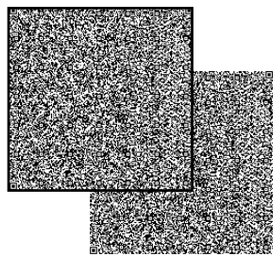
$\mathcal{X}$



m= 500

7 200 000-fold compression

$kd = 768; m/kd < 1$

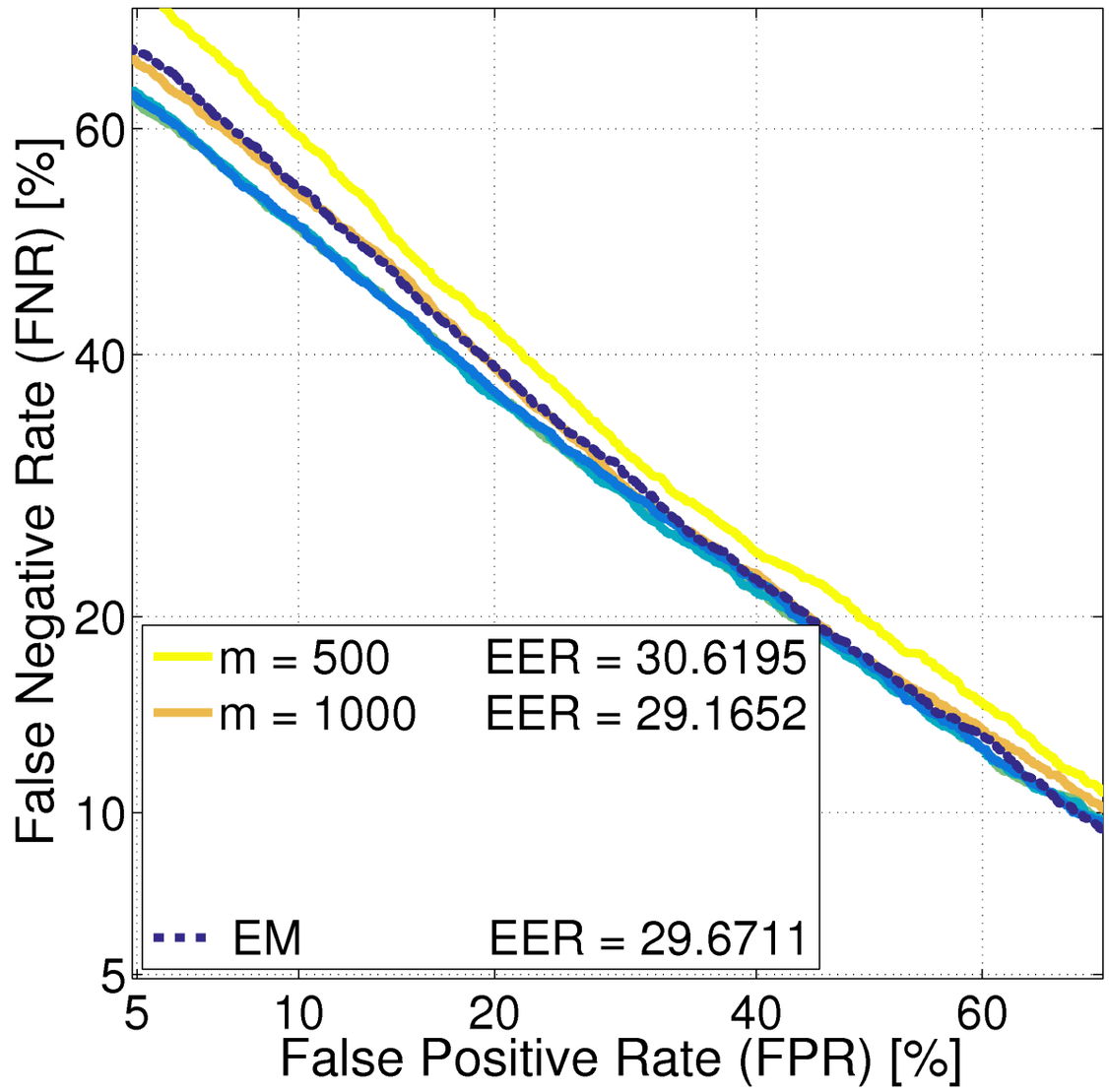


m= 1000

3 600 000-fold compression

$kd = 768; m/kd \approx 1.3$

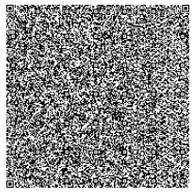
$k = 64; n_{\text{Sketch}} = 200 \times n_{\text{EM}} = 6.10^7$



Proof of Concept: Speaker Verification with Compressive GMM-UBM

~ 50 Gbytes
~ 1000 hours of speech

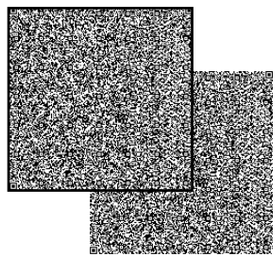
$\mathcal{X}$



m= 500

- 7 200 000-fold compression

$kd = 768; m/kd < 1$



m= 1000

- 3 600 000-fold compression

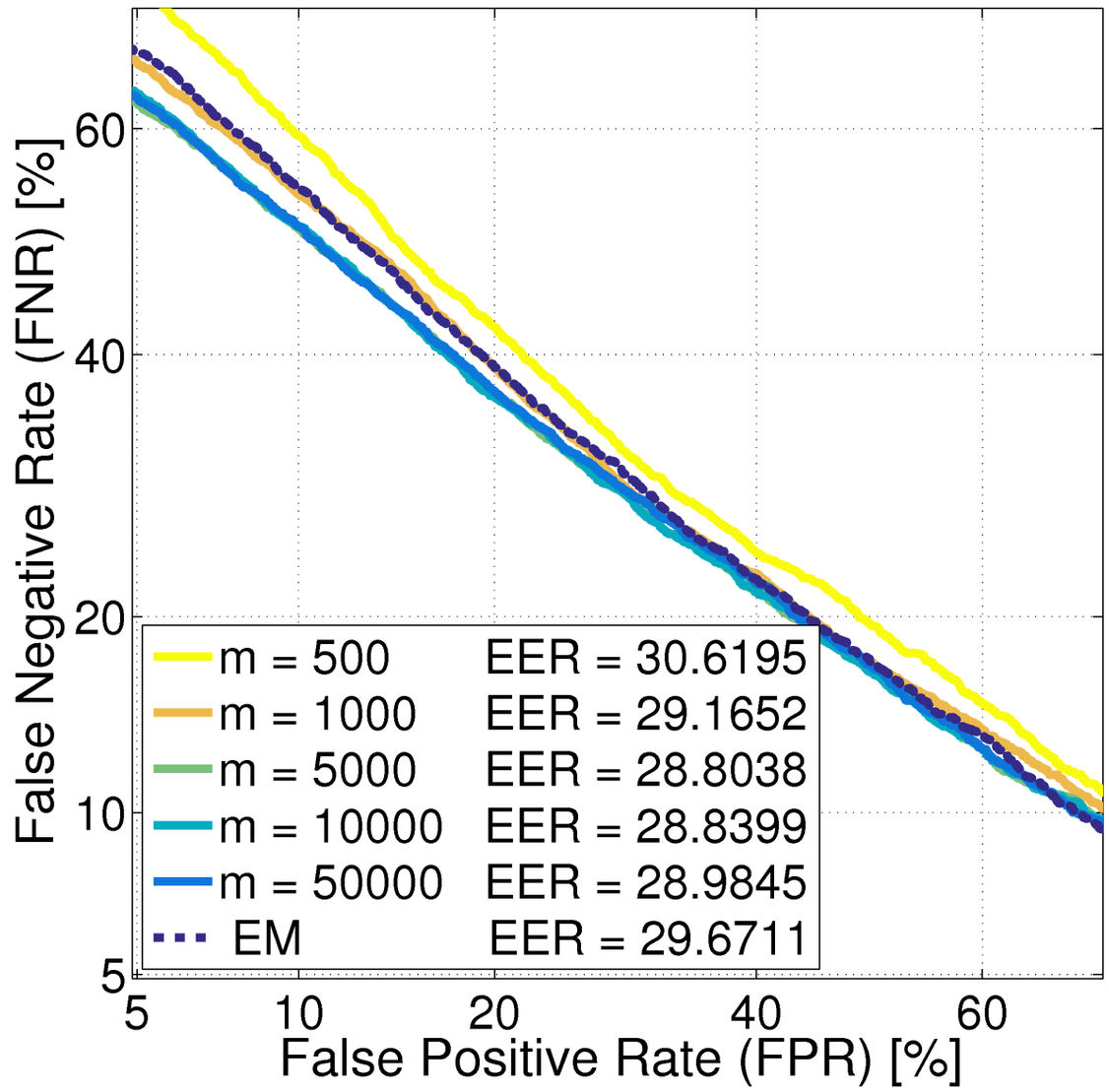
$kd = 768; m/kd \approx 1.3$

m= 5 000

- 720 000-fold compression
- *exploit whole collection*
- *improved performance*

$kd = 768; m/kd \approx 7$

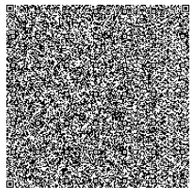
$k = 64; n_{\text{Sketch}} = 200 \times n_{\text{EM}} = 6.10^7$



Proof of Concept: Speaker Verification with Compressive GMM-UBM

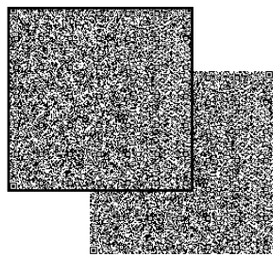
~ 50 Gbytes
~ 1000 hours of speech

$\mathcal{X}$



m= 500
- 7 200 000-fold compression

$kd = 768; m/kd < 1$



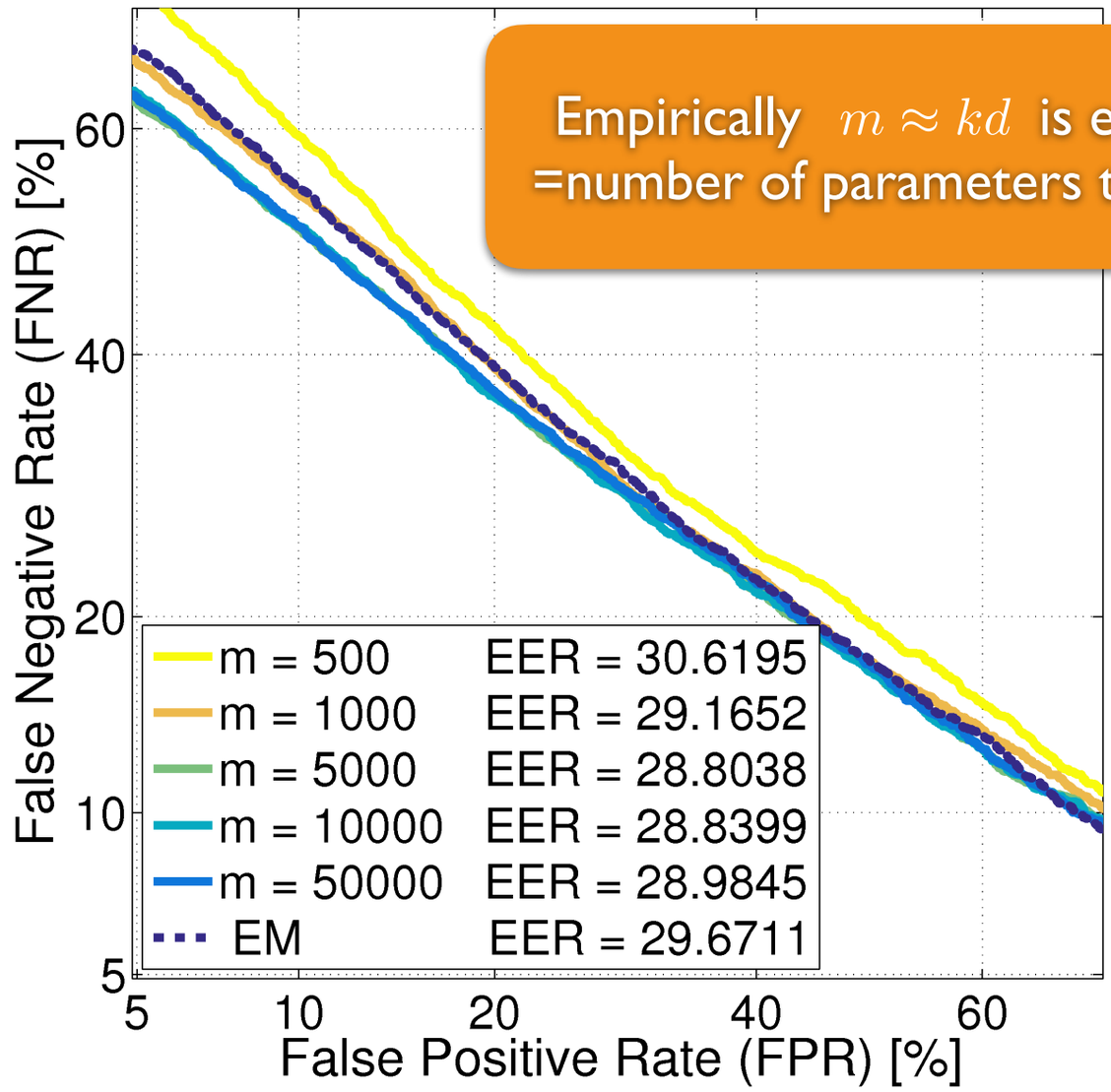
m= 1000
- 3 600 000-fold compression

$kd = 768; m/kd \approx 1.3$

m= 5 000
- 720 000-fold compression
- *exploit whole collection*
- *improved performance*

$kd = 768; m/kd \approx 7$

$k = 64; n_{\text{Sketch}} = 200 \times n_{\text{EM}} = 6.10^7$



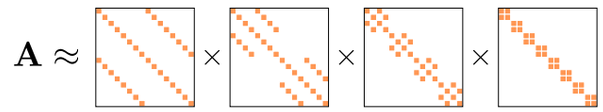
Empirically $m \approx kd$ is enough
=number of parameters to learn

Et ensuite ?

Quelques autres pistes

Frugalité via les *architectures* d'apprentissage

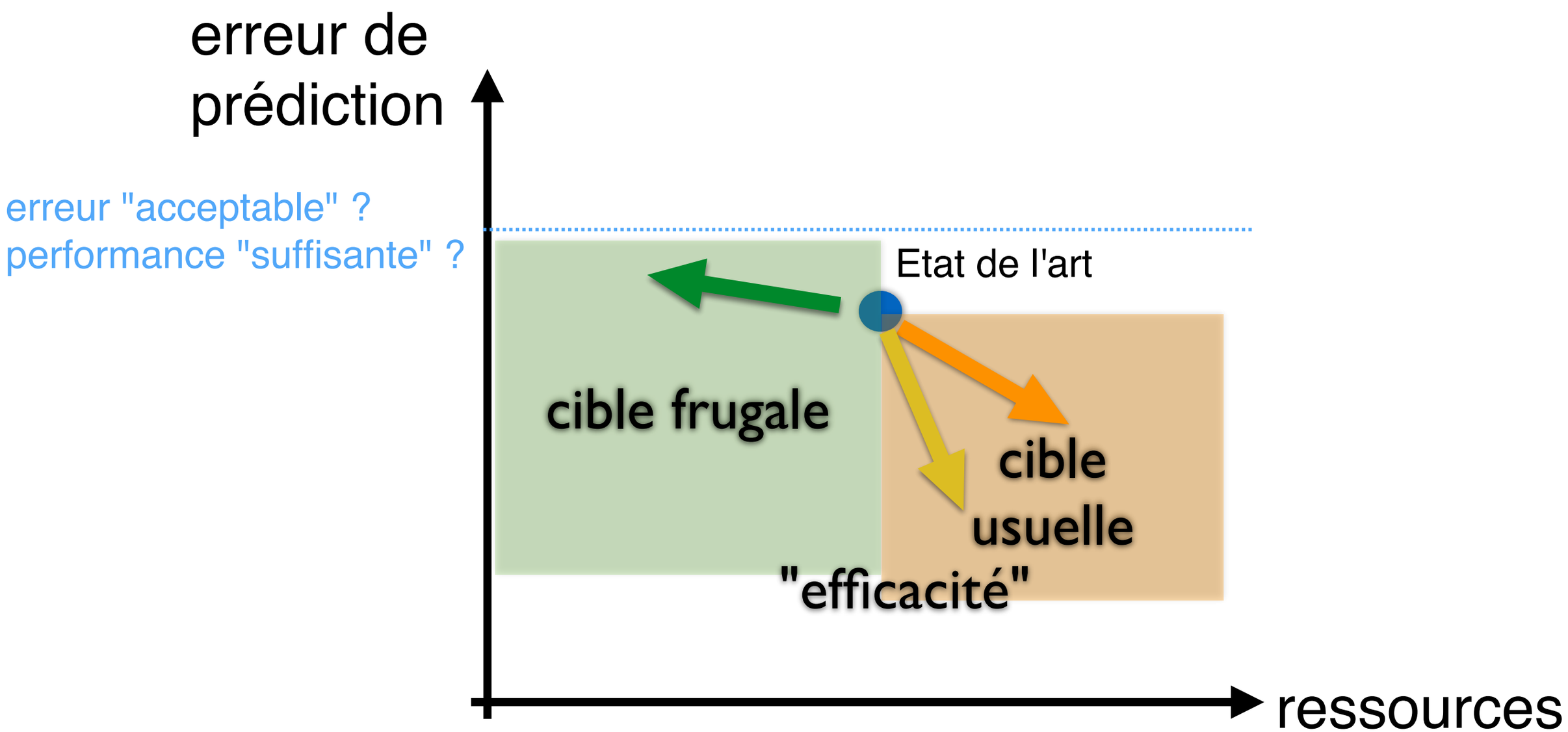
- inférence intrinsèquement frugale ...
  - notamment via parcimonie & butterflies
- ... mais aussi entraînement intrinsèquement frugal
  - distribué, binarisation, sketching (of gradients, vectors, datasets) ...



Frugalité via des *principes*

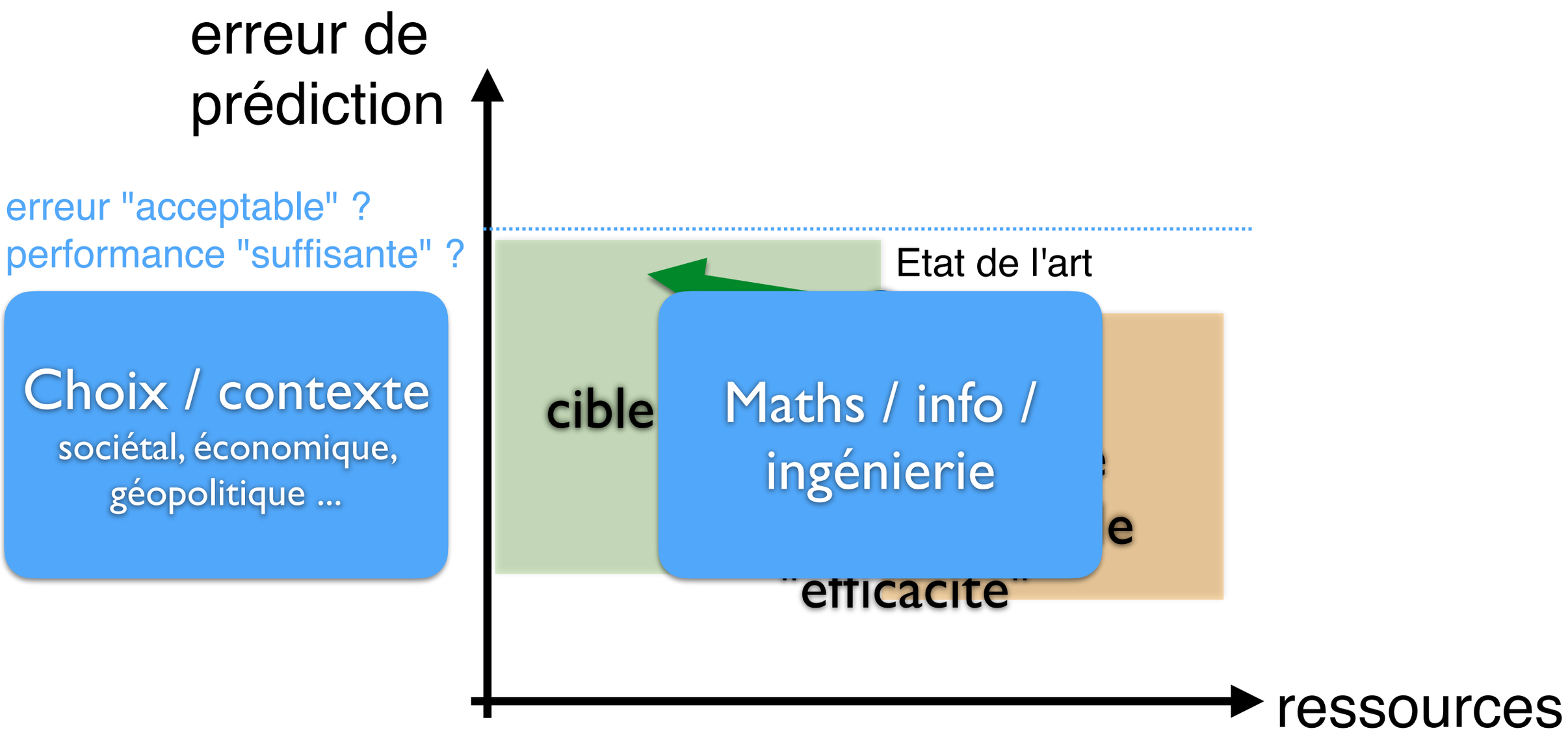
- limiter le gâchis du "end-to-end learning" !
- **incorporer nos connaissances durement acquises**
  - modèles physiques, connus ou paramétrés, e.g. via des EDP
  - symétries et invariances, au-delà de l'augmentation des données
  - graphes, bases de connaissance
  - apprentissage "quality-aware"; sélection des données pertinentes
- ***effets bénéfiques sur*** "interprétabilité", confidentialité, robustesse

To be or not to be frugal ?



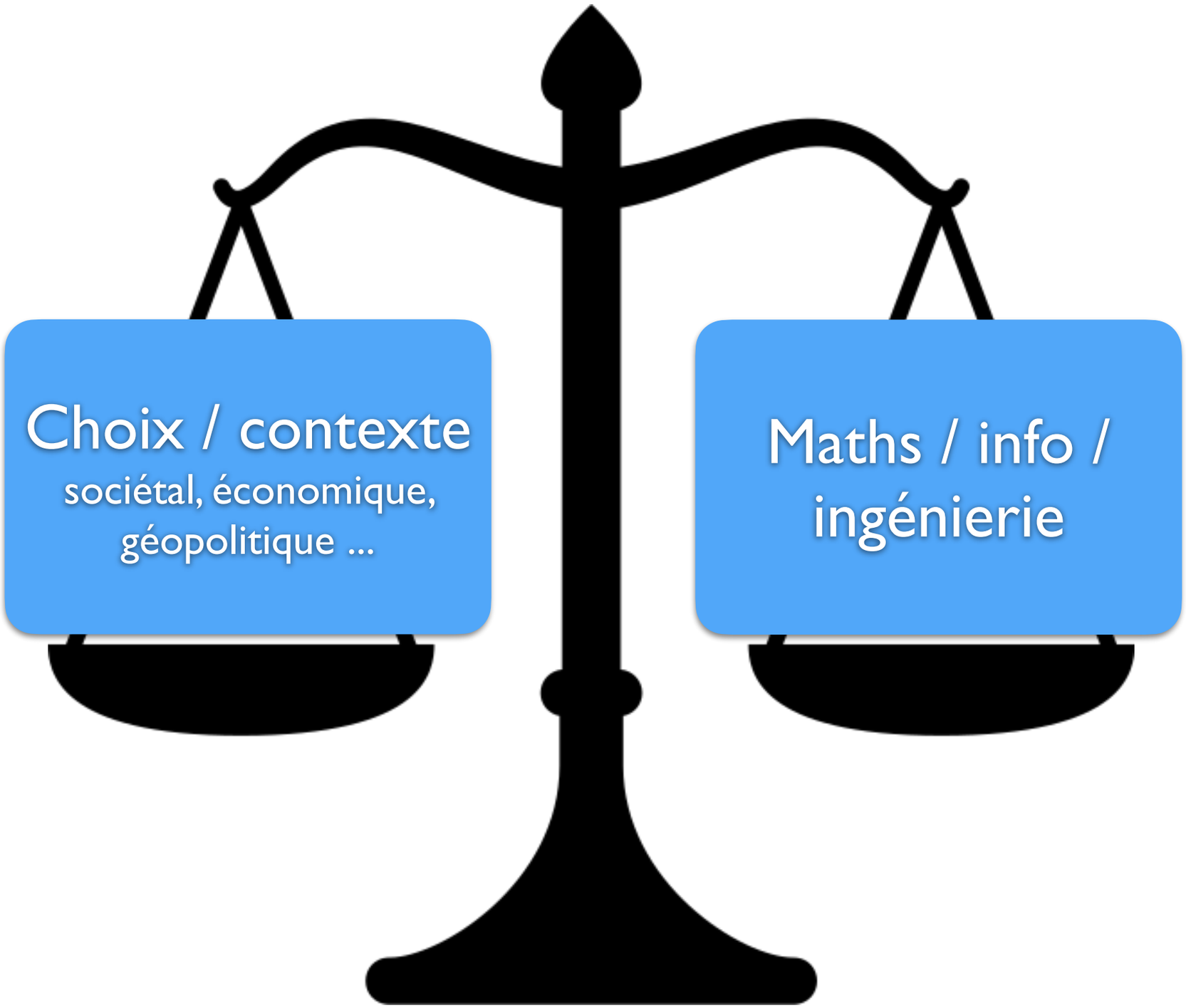
source image: https://fr.m.wikipedia.org/wiki/Fichier:Balance,_by_David.svg, CC Attribution 3.0

To be or not to be frugal ?



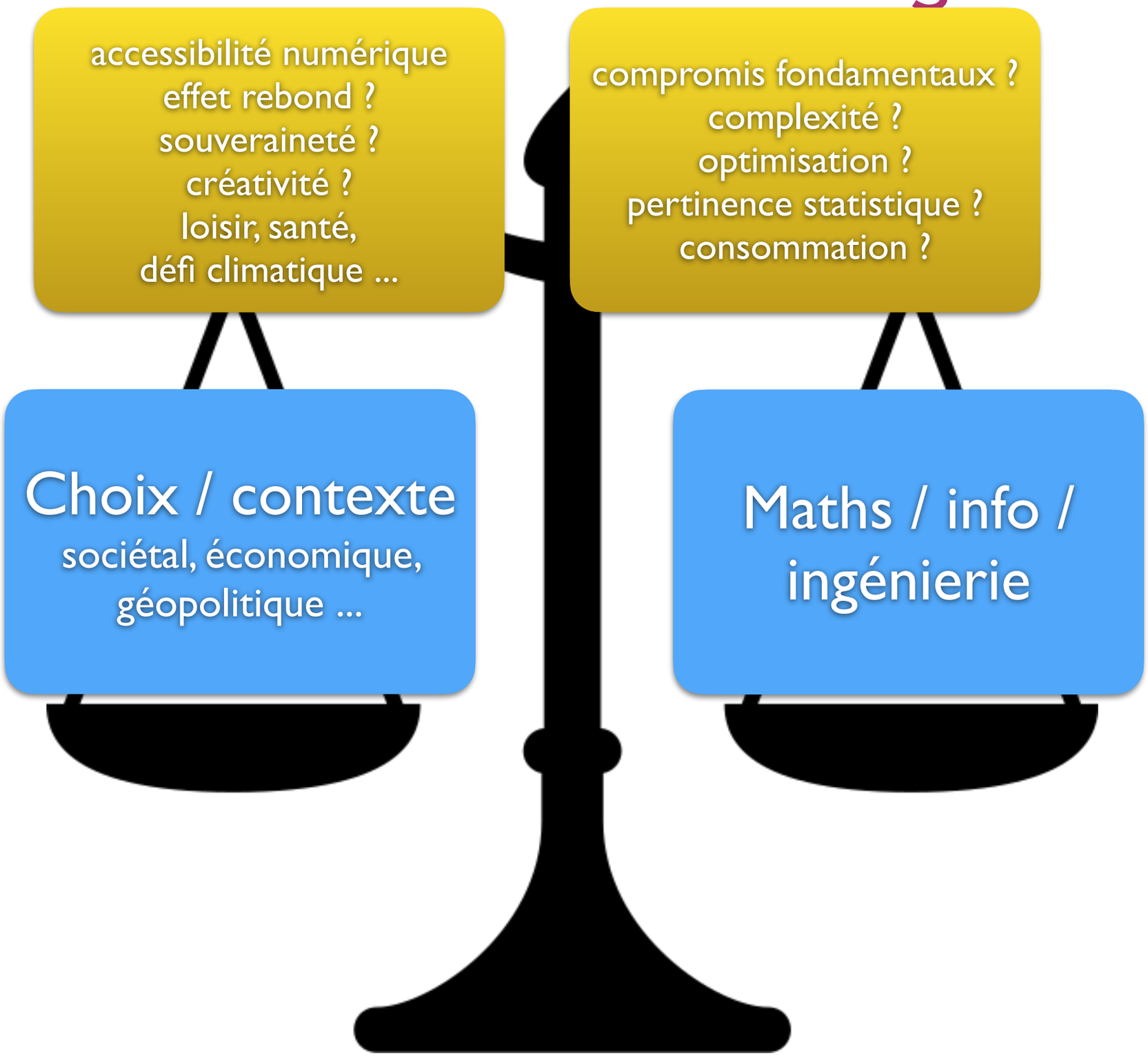
source image: https://fr.m.wikipedia.org/wiki/Fichier:Balance,_by_David.svg, CC Attribution 3.0

To be or not to be frugal ?



source image: https://fr.m.wikipedia.org/wiki/Fichier:Balance,_by_David.svg, CC Attribution 3.0

To be or not to be frugal ?



source image: https://fr.m.wikipedia.org/wiki/Fichier:Balance,_by_David.svg, CC Attribution 3.0

This work was supported in part by the AllegroAssai ANR project ANR-19-CHIA-0009, and GdR ISIS project MOMIGS

The End