Efficient and Privacy-Preserving Compressive Learning

Ph.D. Defense

Author: Antoine Chatalic
Supervisor: Rémi Gribonval
Jury members: Francis Bach (reviewer)
Lorenzo Rosasco (reviewer)
Jamal Atif
Mike Davies
Magalie Fromont Renoir
Marc Tommasi

Université de Rennes 1, IRISA/Inria, Panama research group

November 19th, 2020

Efficient and Privacy-Preserving Compressive Learning

Efficient and Privacy-Preserving Compressive Learning

An introduction to machine learning

Example: email classification for phishing detection.

An introduction to machine learning

Example: email classification for phishing detection.

From: `trusted.colleague@irisa.fr`
Subject: [panama] Next working group
Content: Dear Panama, the reading group is coming back! [...]

✓ OK

From: `inegon06@netscape.com`
Subject: Avoid Suspension!!
Content: Dear subscriber, Your Microsoft account has been compromised. You must update it immediately or your account will be closed. [...]

✗ Phishing

An introduction to machine learning

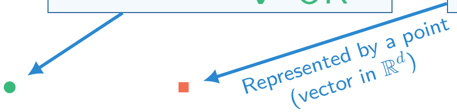
Example: email classification for phishing detection.

From: `trusted.colleague@irisa.fr`
Subject: [panama] Next working group
Content: Dear Panama, the reading group is coming back! [...]

✓ OK

From: `inegon06@netscape.com`
Subject: Avoid Suspension!!
Content: Dear subscriber, Your Microsoft account has been compromised. You must update it immediately or your account will be closed. [...]

✗ Phishing



An introduction to machine learning

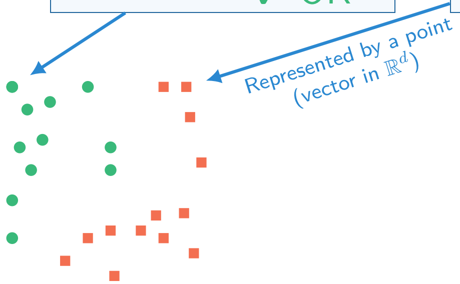
Example: email classification for phishing detection.

From: `trusted.colleague@irisa.fr`
Subject: [panama] Next working group
Content: Dear Panama, the reading group is coming back! [...]

✓ OK

From: `inegon06@netscape.com`
Subject: Avoid Suspension!!
Content: Dear subscriber, Your Microsoft account has been compromised. You must update it immediately or your account will be closed. [...]

✗ Phishing



An introduction to machine learning

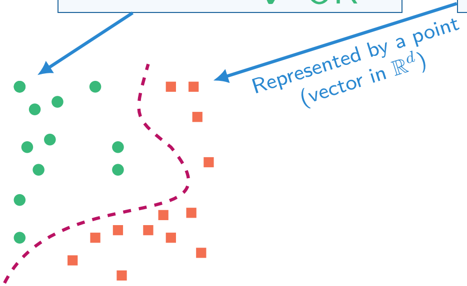
Example: email classification for phishing detection.

From: `trusted.colleague@irisa.fr`
Subject: [panama] Next working group
Content: Dear Panama, the reading group is coming back! [...]

✓ OK

From: `inegon06@netscape.com`
Subject: Avoid Suspension!!
Content: Dear subscriber, Your Microsoft account has been compromised. You must update it immediately or your account will be closed. [...]

✗ Phishing



An introduction to machine learning

Example: email classification for phishing detection.

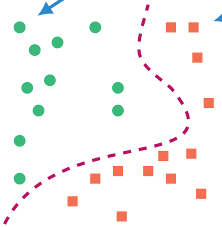
From: `trusted.colleague@irisa.fr`
Subject: [panama] Next working group
Content: Dear Panama, the reading group is coming back! [...]

✓ OK

From: `inegon06@netscape.com`
Subject: Avoid Suspension!!
Content: Dear subscriber, Your Microsoft account has been compromised. You must update it immediately or your account will be closed. [...]

✗ Phishing

Represented by a point (vector in $\mathbb{R}^d$)

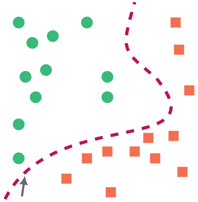


This is a **supervised binary classification problem.**

General idea: fit a **mathematical model** using the measured data.

Different types of learning tasks

Binary classification
(supervised learning)

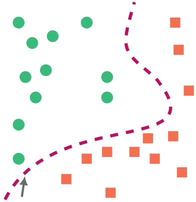


Model: smooth separator.

Applications: phising detection, image classification...

Different types of learning tasks

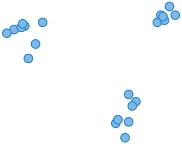
Binary classification
(supervised learning)



Model: smooth separator.

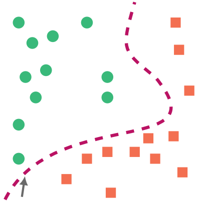
Applications: phising detection, image classification...

Clustering
(unsupervised learning)



Different types of learning tasks

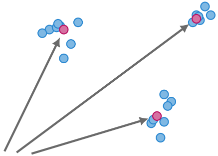
Binary classification
(supervised learning)



Model: smooth separator.

Applications: phising detection, image classification…

Clustering
(unsupervised learning)

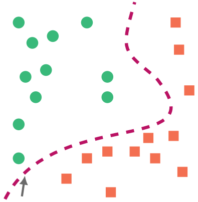


Model: set of k points.

Applications: community detection, anomaly detection…

Different types of learning tasks

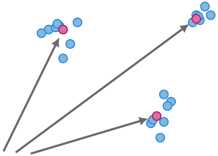
Binary classification
(supervised learning)



Model: smooth separator.

Applications: phising detection, image classification…

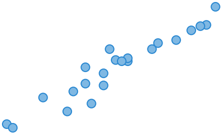
Clustering
(unsupervised learning)



Model: set of k points.

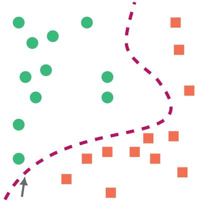
Applications: community detection, anomaly detection…

Principal component analysis (PCA)
(unsupervised learning)



Different types of learning tasks

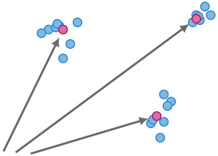
Binary classification
(supervised learning)



Model: smooth separator.

Applications: phising detection, image classification...

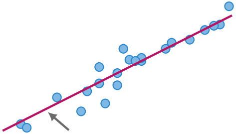
Clustering
(unsupervised learning)



Model: set of k points.

Applications: community detection, anomaly detection...

Principal component analysis (PCA)
(unsupervised learning)

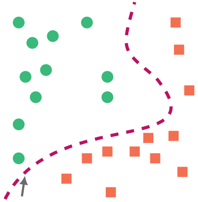


Model: k-dimensional linear subspace.

Applications: compression, data visualization...

Different types of learning tasks

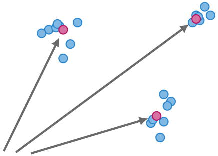
Binary classification
(supervised learning)



Model: smooth separator.

Applications: phising detection, image classification...

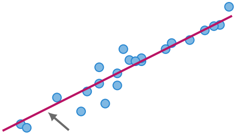
Clustering
(unsupervised learning)



Model: set of k points.

Applications: community detection, anomaly detection...

Principal component analysis (PCA)
(unsupervised learning)



Model: k-dimensional linear subspace.

Applications: compression, data visualization...

Goal: find the parameters θ of our model $\mathcal{M}_\theta$ which best "fit" the data:

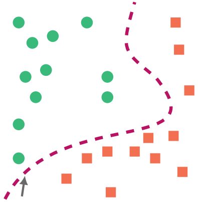
$$h^* = \arg\min_{\theta \in \Theta} \mathcal{R}(\mathcal{M}_\theta, \text{data}).$$

Set of parameters defining a low dimension model

Risk function measuring how a model "fits" the data

Different types of learning tasks

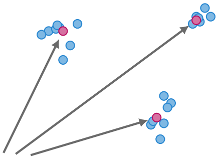
Binary classification
(supervised learning)



Model: smooth separator.

Applications: phising detection, image classification...

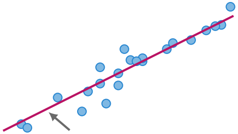
Clustering
(unsupervised learning)



Model: set of k points.

Applications: community detection, anomaly detection...

Principal component analysis (PCA)
(unsupervised learning)



Model: k-dimensional linear subspace.

Applications: compression, data visualization...

Goal: find the parameters θ of our model $\mathcal{M}_\theta$ which best "fit" the data:

$$h^* = \arg\min_{\theta \in \Theta} \mathcal{R}(\mathcal{M}_\theta, \text{data}).$$

Set of parameters defining a low dimension model

Risk function measuring how a model "fits" the data

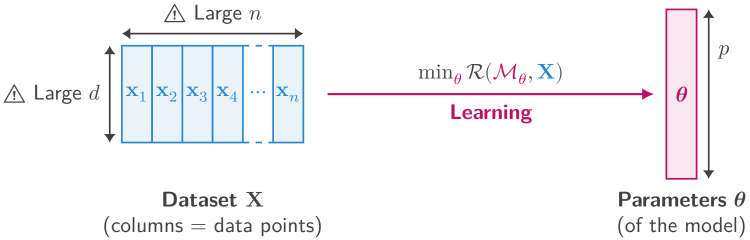
Main challenges

Dataset $\mathbf{X}$
(columns = data points)

Parameters θ
(of the model)

Challenges:

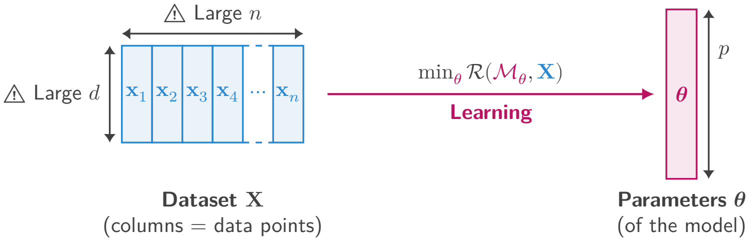
Main challenges



Challenges:

- ↔ Large data collections.
- ↕ High-dimensional features.

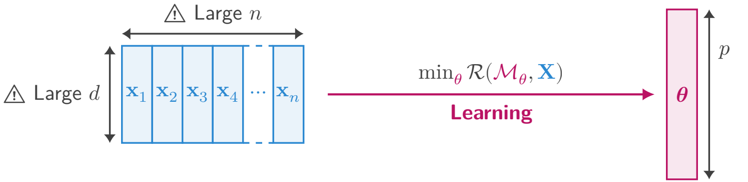
Main challenges



Challenges:

- Large data collections.
- High-dimensional features.
- Distributed datasets.
- Data streams.

Main challenges



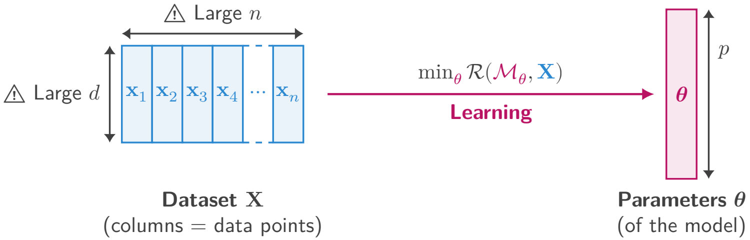
Dataset $\mathbf{X}$
(columns $=$ data points)

Parameters $\boldsymbol{\theta}$
(of the model)

Challenges:

- Large data collections.
- High-dimensional features.
- Distributed datasets.
- Data streams.
- Sensitive data
  (e.g. emails, medical data).

Main challenges



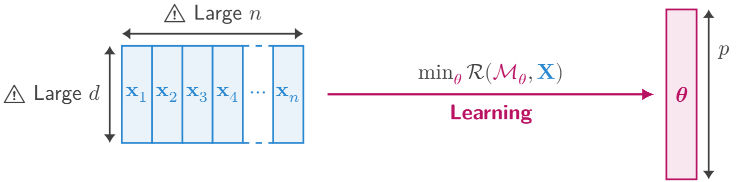
Challenges:

- Large data collections.
- High-dimensional features.
- Distributed datasets.
- Data streams.
- Sensitive data
(e.g. emails, medical data).

Limitations of "standard" methods:

- Multiple passes on the data.
- Computationally expensive.
- High energy consumption.

Main challenges



Dataset $\mathbf{X}$
(columns = data points)

Parameters $\boldsymbol{\theta}$
(of the model)

Challenges:

- Large data collections.
- High-dimensional features.
- Distributed datasets.
- Data streams.
- Sensitive data (e.g. emails, medical data).

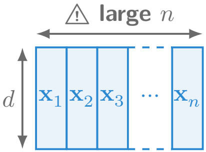
Limitations of “standard” methods:

- Multiple passes on the data.
- Computationally expensive.
- High energy consumption.

Can we do better?

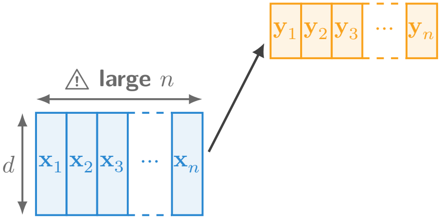
Existing approaches for large-scale learning

Reduce/compress the data!



Existing approaches for large-scale learning

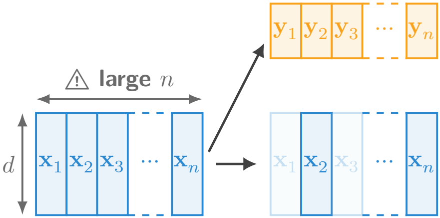
Reduce/compress the data!



- **Dimensionality Reduction**
  Data-dependent or independent.

Existing approaches for large-scale learning

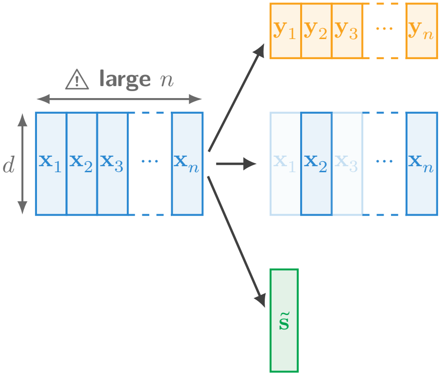
Reduce/compress the data!



- **Dimensionality Reduction**
  Data-dependent or independent.
- **Coresets**
  Subsampling, geometric decompositions.

Existing approaches for large-scale learning

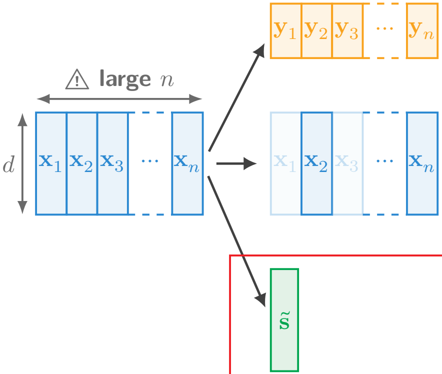
Reduce/compress the data!



- **Dimensionality Reduction**
  Data-dependent or independent.
- **Coresets**
  Subsampling, geometric decompositions.
- **Sketching**
  Sketches of moments, "linear" sketches.

Existing approaches for large-scale learning

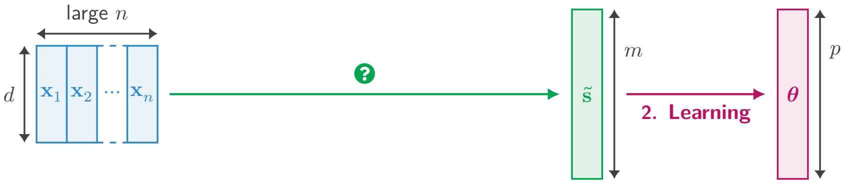
Reduce/compress the data!



- **Dimensionality Reduction**
  Data-dependent or independent.
- **Coresets**
  Subsampling, geometric decompositions.
- **Sketching**
  Sketches of moments, "linear" sketches.

The Compressive Learning Framework

Compressive learning

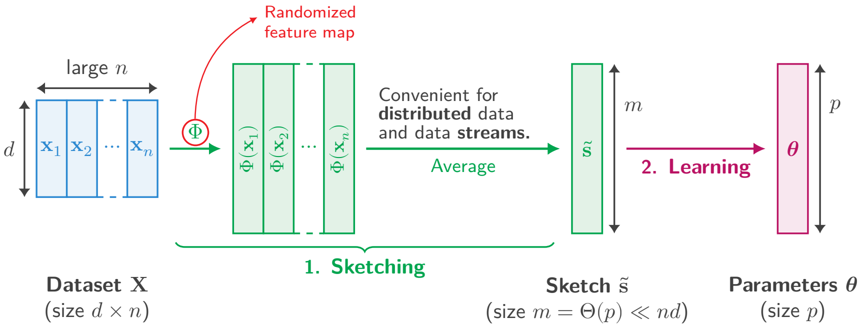


Dataset X
(size $d \times n$)

Sketch $\tilde{\mathbf{s}}$
(size $m = \Theta(p) \ll nd$)

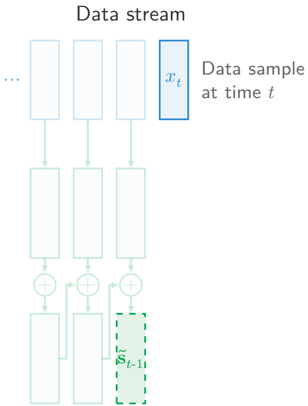
Parameters $\boldsymbol{\theta}$
(size p)

Compressive learning



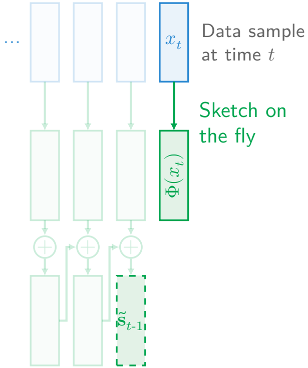
The sketch is just a vector of “generalized” moments!

Benefits of sketching

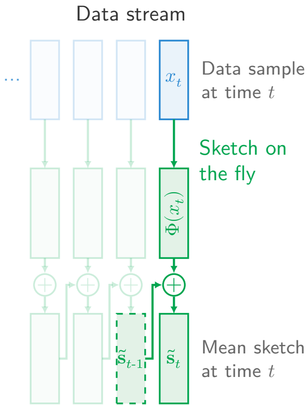


Benefits of sketching

Data stream

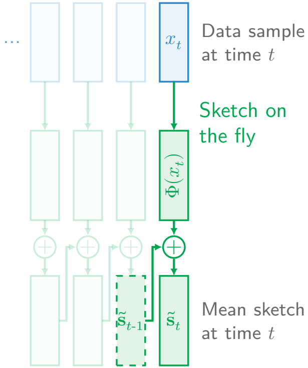


Benefits of sketching

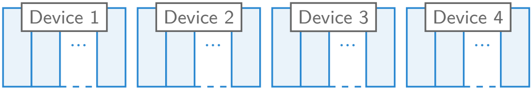


Benefits of sketching

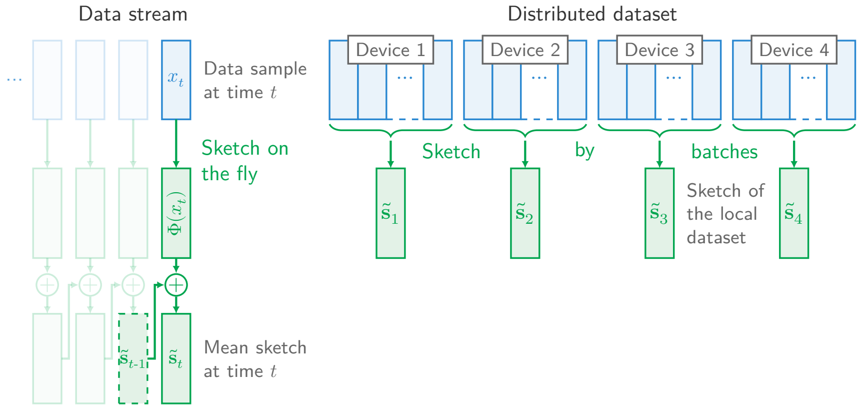
Data stream



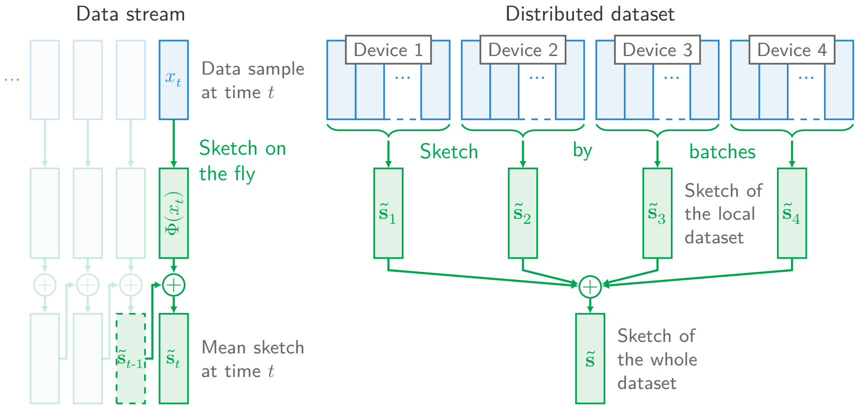
Distributed dataset



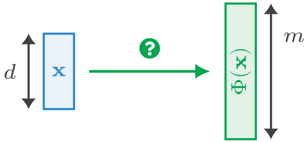
Benefits of sketching



Benefits of sketching



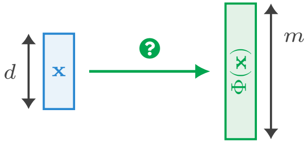
Which feature map Φ can we use?



We consider $\Phi(\mathbf{x}) \triangleq \rho(\mathbf{\Omega}^T\mathbf{x})$ where

- $\mathbf{\Omega} = [\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_m] \in \mathbb{R}^{d\times m}$ is a **random** matrix (e.g., i.i.d. normal entries);
- ρ is a **deterministic non-linear** function applied pointwise.

Which feature map Φ can we use?



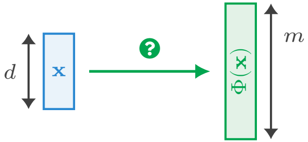
We consider $\Phi(\mathbf{x}) \triangleq \rho(\mathbf{\Omega}^T\mathbf{x})$ where

- $\mathbf{\Omega} = [\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_m] \in \mathbb{R}^{d\times m}$ is a **random** matrix (e.g., i.i.d. normal entries);
- ρ is a **deterministic non-linear** function applied pointwise.

Examples:

- For k-means: $\rho(t) \triangleq \exp(-\iota t)$ **(random Fourier features)**
  [Rahimi and Recht, 2008. **"Random Features for Large-Scale Kernel Machines"**]
  (Sketch = random samples of the empirical **characteristic function**.)

Which feature map Φ can we use?



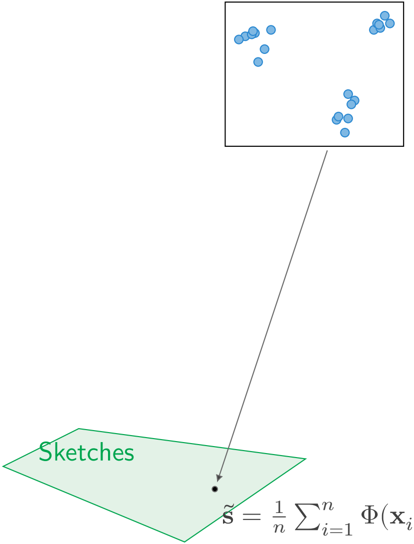
We consider $\Phi(\mathbf{x}) \triangleq \rho(\mathbf{\Omega}^T\mathbf{x})$ where

- $\mathbf{\Omega} = [\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_m] \in \mathbb{R}^{d\times m}$ is a **random** matrix (e.g., i.i.d. normal entries);
- ρ is a **deterministic non-linear** function applied pointwise.

Examples:

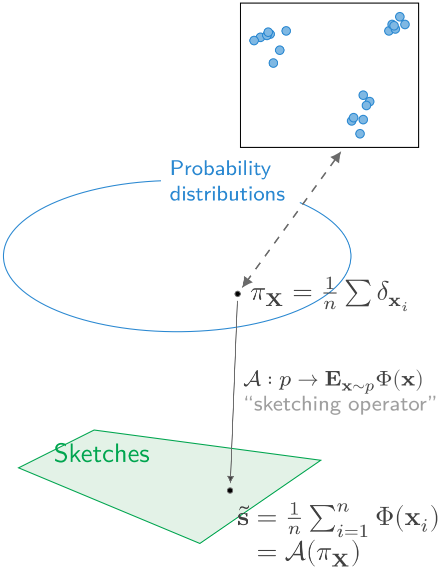
- For k-means: $\rho(t) \triangleq \exp(-\iota t)$ **(random Fourier features)**
  [Rahimi and Recht, 2008. **"Random Features for Large-Scale Kernel Machines"**]
  (Sketch = random samples of the empirical **characteristic function**.)
- For PCA: $\rho(t) \triangleq t^2$ **(random quadratic features)**
  (Sketch = rank-one linear measurements of the covariance matrix for centered data.)

Learning as an inverse problem

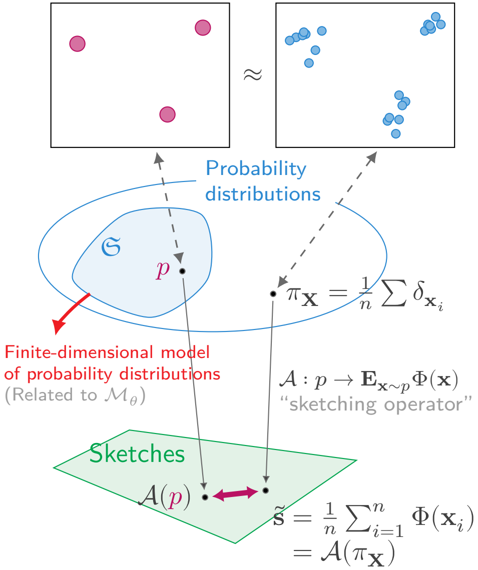


$\tilde{\mathbf{s}} = \frac{1}{n}\sum_{i=1}^{n}\Phi(\mathbf{x}_i)$

Learning as an inverse problem



Learning as an inverse problem

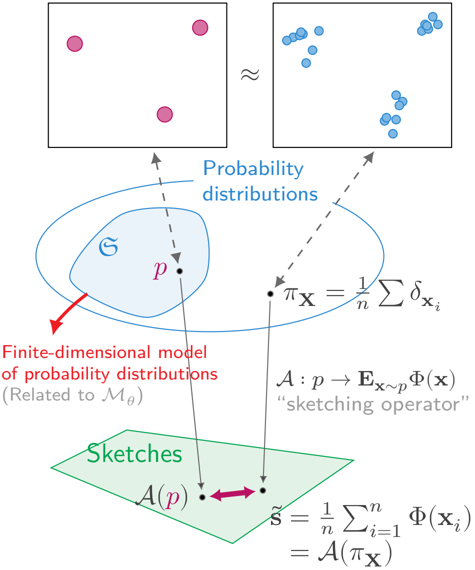


Moment-matching problem:

$$\arg\min_{p\in\mathfrak{S}} \Big\| \underbrace{\mathcal{A}(p)}_{\text{sketch of } p} - \underbrace{\tilde{\mathbf{s}}}_{\text{empirical sketch}} \Big\|_2$$

Cf. generalized method of moments [Hall, 2005] .

Learning as an inverse problem



Moment-matching problem:

$$\arg\min_{p\in\mathfrak{S}} \Big\| \underbrace{\mathcal{A}(p)}_{\text{sketch of } p} - \underbrace{\tilde{\mathbf{s}}}_{\substack{\text{empirical}\\ \text{sketch}}} \Big\|_2$$

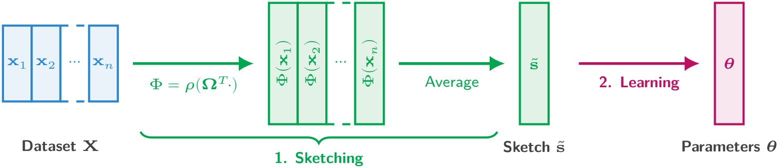
Cf. generalized method of moments [Hall, 2005] .

⚠ Difficult/non-convex problem!

Heuristics can be used, e.g. "continuous" matching pursuit.

[Keriven et al., 2017]

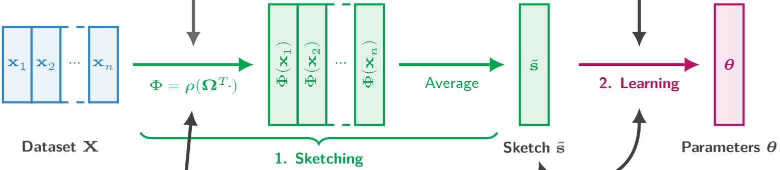
Contributions



Contributions

Sketching operator design
Kernel scale tuning for clustering, considerations for extension to broader tasks.

Algorithmic recovery
A message-passing algorithm for compressive clustering.



Efficiency
Speed-up both sketching and learning phases.

Privacy Preservation
Make the sketch "private", so that it can be released without damaging the privacy of users.

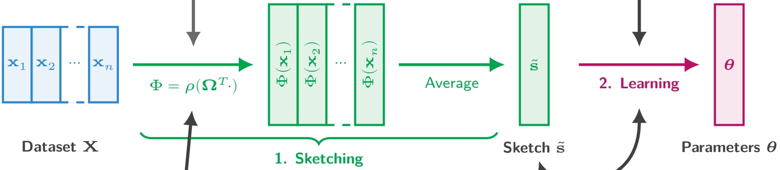
Contributions

Not today

Sketching operator design
Kernel scale tuning for clustering, considerations for extension to broader tasks.

Not today

Algorithmic recovery
A message-passing algorithm for compressive clustering.



2

Efficiency
Speed-up both sketching and learning phases.

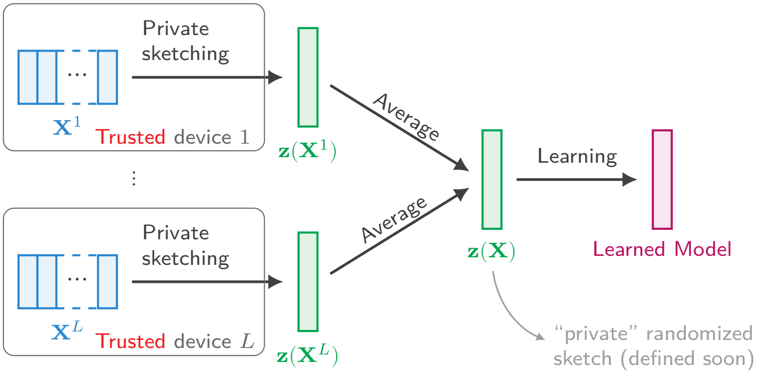
1

Privacy Preservation
Make the sketch “private”, so that it can be released without damaging the privacy of users.

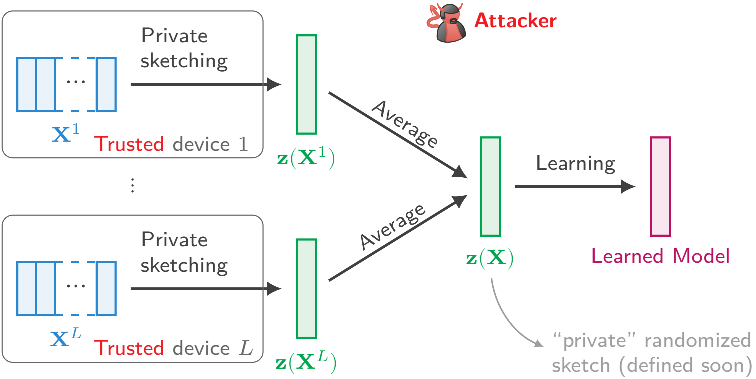
Privacy-Preserving Compressive Learning

(Work in collaboration with V. Schellekens, F. Houssiau, L. Jacques and Y.-A. de Montjoye)

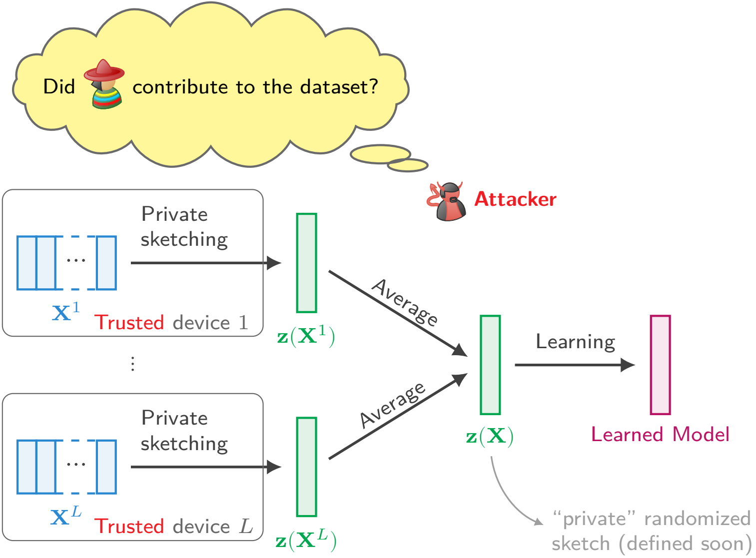
Privacy preservation: what are we talking about?



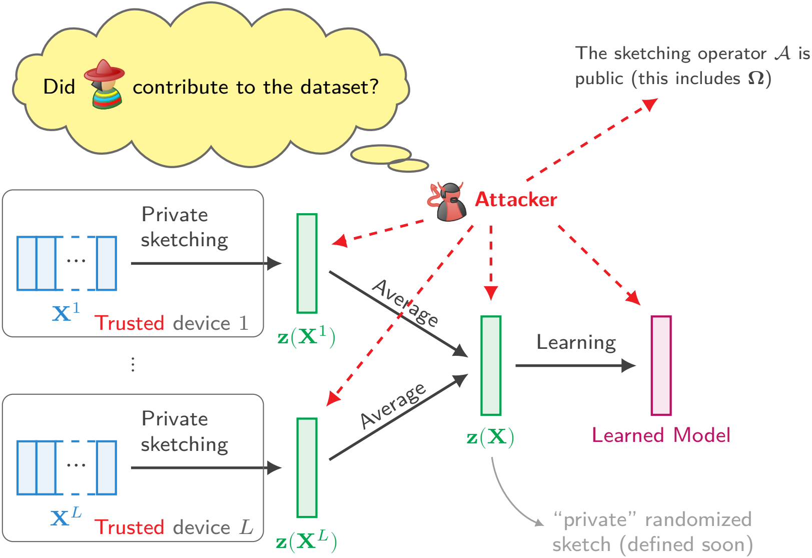
Privacy preservation: what are we talking about?



Privacy preservation: what are we talking about?



Privacy preservation: what are we talking about?



Defining and quantifying privacy

[Dwork et al., 2006. **"Calibrating Noise to Sensitivity in Private Data Analysis"**]

Definition: The randomized mechanism $\mathbf{z}(\cdot)$ achieves (ε, δ)-differential privacy (DP) iff for any (input) neighbor datasets $\mathbf{X}_1 \sim \mathbf{X}_2$ and set S:

$$\mathbb{P}[\mathbf{z}(\mathbf{X}_1) \in S] \leq \exp(\varepsilon)\, \mathbb{P}[\mathbf{z}(\mathbf{X}_2) \in S] + \delta$$

Defining and quantifying privacy

[Dwork et al., 2006. **"Calibrating Noise to Sensitivity in Private Data Analysis"**]

Definition: The randomized mechanism $\mathbf{z}(\cdot)$ achieves (ε, δ)-differential privacy (DP) iff for any (input) neighbor datasets $\mathbf{X}_1 \sim \mathbf{X}_2$ and set S:

$$\mathbb{P}[\mathbf{z}(\mathbf{X}_1) \in S] \leq \exp(\varepsilon)\, \mathbb{P}[\mathbf{z}(\mathbf{X}_2) \in S] + \delta$$

Examples of neighboring relations:

- replacement of one element:

- add/removal of one element:

$\mathbf{X}_1$ $\qquad$ $\mathbf{X}_2 = \mathbf{X}_1 +$ 

Defining and quantifying privacy

[Dwork et al., 2006. **"Calibrating Noise to Sensitivity in Private Data Analysis"**]

Definition: The randomized mechanism $\mathbf{z}(\cdot)$ achieves (ε, δ)-differential privacy (DP) iff for any (input) neighbor datasets $\mathbf{X}_1 \sim \mathbf{X}_2$ and set S:

$$\mathbb{P}[\mathbf{z}(\mathbf{X}_1) \in S] \leq \exp(\varepsilon)\, \mathbb{P}[\mathbf{z}(\mathbf{X}_2) \in S] + \delta$$

Examples of neighboring relations:

- replacement of one element:

- add/removal of one element:


$\mathbf{X}_1$ $\quad$ $\mathbf{X}_2 = \mathbf{X}_1 +$ 

Defining and quantifying privacy

[Dwork et al., 2006. **"Calibrating Noise to Sensitivity in Private Data Analysis"**]

Definition: The randomized mechanism $\mathbf{z}(\cdot)$ achieves (ε, δ)-differential privacy (DP) iff for any (input) neighbor datasets $\mathbf{X}_1 \sim \mathbf{X}_2$ and set S:

$$\mathbb{P}[\mathbf{z}(\mathbf{X}_1) \in S] \leq \exp(\varepsilon)\, \mathbb{P}[\mathbf{z}(\mathbf{X}_2) \in S] + \delta$$

relaxation ("approximate DP" when $\delta > 0$)

privacy "budget" (smaller ε = more privacy)

Examples of neighboring relations:

- replacement of one element:
   $\sim$  .
- add/removal of one element:
   $\sim$ .

$\mathbf{X}_1$ $\quad$ $\mathbf{X}_2 = \mathbf{X}_1 +$ 

Defining and quantifying privacy

[Dwork et al., 2006. **"Calibrating Noise to Sensitivity in Private Data Analysis"**]

Definition: The randomized mechanism $\mathbf{z}(\cdot)$ achieves (ε, δ)-differential privacy (DP) iff for any (input) neighbor datasets $\mathbf{X}_1 \sim \mathbf{X}_2$ and set S:

$$\mathbb{P}[\mathbf{z}(\mathbf{X}_1) \in S] \leq \exp(\varepsilon)\, \mathbb{P}[\mathbf{z}(\mathbf{X}_2) \in S] + \delta$$

relaxation ("approximate DP" when $\delta > 0$)

privacy "budget" (smaller ε = more privacy)

Examples of neighboring relations:

- replacement of one element:

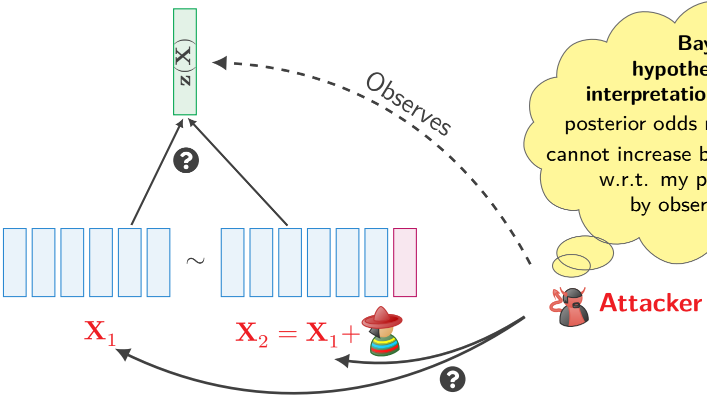
- add/removal of one element:

$\mathbf{X}_1$ $\quad$ $\mathbf{X}_2 = \mathbf{X}_1 +$ 

Notation:

- (ε, δ)-DP in general;
- ε-DP when $\delta = 0$.

Interpretation of ε-DP

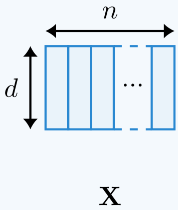


Bayesian hypothesis testing interpretation of ε-DP: The posterior odds ratio $\frac{\mathbb{P}[\mathbf{X}=\mathbf{X}_1|\mathbf{z}(\mathbf{X})]}{\mathbb{P}[\mathbf{X}=\mathbf{X}_2|\mathbf{z}(\mathbf{X})]}$ cannot increase by more than $\exp(\varepsilon)$ w.r.t. my prior odds ratio by observing $\mathbf{z}(\mathbf{X})$.

Differential privacy by additive perturbation

Simple way to satisfy DP: add noise to the output.

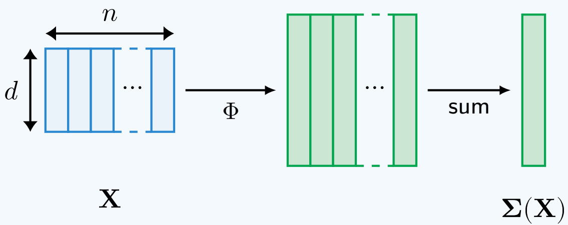
Proposed mechanism



Differential privacy by additive perturbation

Simple way to satisfy DP: add noise to the output.

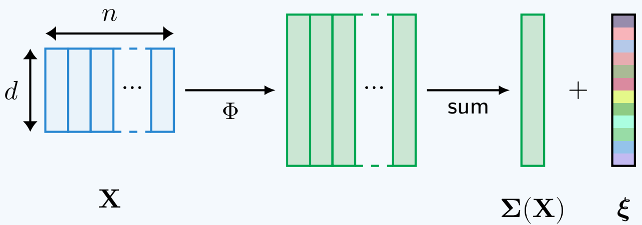
Proposed mechanism



Differential privacy by additive perturbation

Simple way to satisfy DP: add noise to the output.

Proposed mechanism

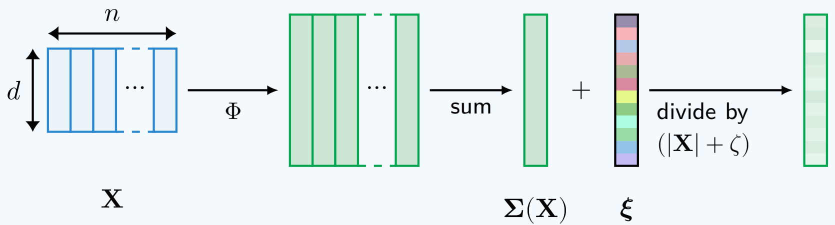


- Add noise ξ on the sum of features.

Differential privacy by additive perturbation

Simple way to satisfy DP: add noise to the output.

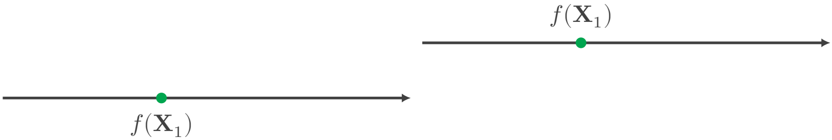
Proposed mechanism



- Add noise $\boldsymbol{\xi}$ on the sum of features.
- Add noise ζ on $|\mathbf{X}|$.

Which noise to ensure privacy? (Common knowledge)

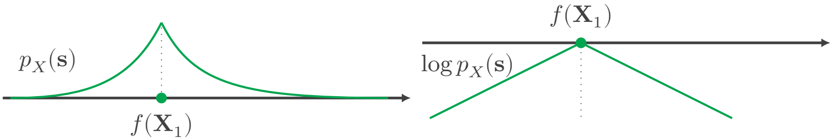
- Laplacian noise for pure ε-DP.



[Dwork et al., 2006. **"Calibrating Noise to Sensitivity in Private Data Analysis"**]

Which noise to ensure privacy? (Common knowledge)

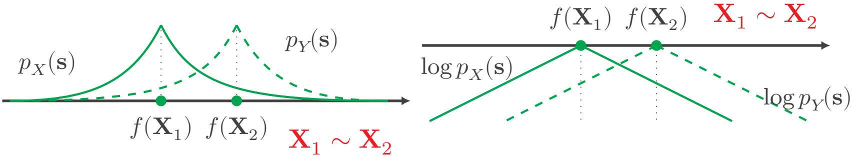
- Laplacian noise for pure ε-DP.



[Dwork et al., 2006. **"Calibrating Noise to Sensitivity in Private Data Analysis"**]

Which noise to ensure privacy? (Common knowledge)

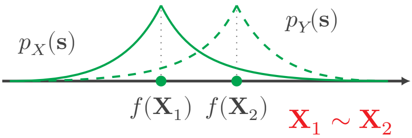
- Laplacian noise for pure ε-DP.

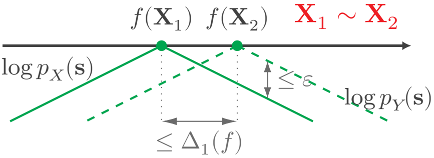


[Dwork et al., 2006. **“Calibrating Noise to Sensitivity in Private Data Analysis”**]

Which noise to ensure privacy? (Common knowledge)

- Laplacian noise for pure ε-DP.



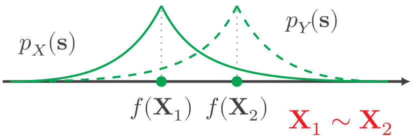


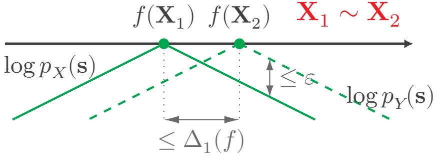
Noise level: $b^* = \frac{\Delta_1(f)}{\varepsilon}$ with $\Delta_1(f) \triangleq \sup_{\mathbf{X}_1 \sim \mathbf{X}_2} \|f(\mathbf{X}_1) - f(\mathbf{X}_2)\|_1$.

[Dwork et al., 2006. **"Calibrating Noise to Sensitivity in Private Data Analysis"**]

Which noise to ensure privacy? (Common knowledge)

- Laplacian noise for pure ε-DP.





Noise level: $b^* = \frac{\Delta_1(f)}{\varepsilon}$ with $\Delta_1(f) \triangleq \sup_{\mathbf{X}_1 \sim \mathbf{X}_2} \|f(\mathbf{X}_1) - f(\mathbf{X}_2)\|_1$.

[Dwork et al., 2006. **"Calibrating Noise to Sensitivity in Private Data Analysis"**]

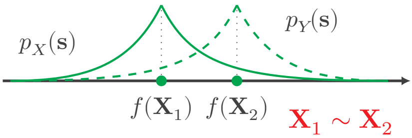
- Gaussian noise for approximate (ε, δ)-DP.

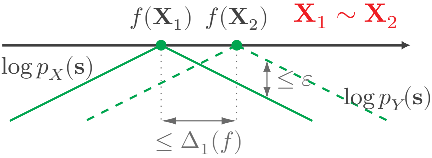
The noise scales with $\Delta_2(f) \triangleq \sup_{\mathbf{X}_1 \sim \mathbf{X}_2} \|f(\mathbf{X}_1) - f(\mathbf{X}_2)\|_2$.

[Balle and Wang, 2018. **"Improving the Gaussian Mechanism for Differential Privacy"**]

Which noise to ensure privacy? (Common knowledge)

- Laplacian noise for pure ε-DP.





Noise level: $b^* = \frac{\Delta_1(f)}{\varepsilon}$ with $\boxed{\Delta_1(f) \triangleq \sup_{\mathbf{X}_1 \sim \mathbf{X}_2} \|f(\mathbf{X}_1) - f(\mathbf{X}_2)\|_1.}$

[Dwork et al., 2006. **"Calibrating Noise to Sensitivity in Private Data Analysis"**]

l_1/l_2 "sensitivities"

- Gaussian noise for approximate (ε, δ)-DP.

  The noise scales with $\boxed{\Delta_2(f) \triangleq \sup_{\mathbf{X}_1 \sim \mathbf{X}_2} \|f(\mathbf{X}_1) - f(\mathbf{X}_2)\|_2.}$

  [Balle and Wang, 2018. **"Improving the Gaussian Mechanism for Differential Privacy"**]

Privacy results

	Pure ε-DP	Approximate (ε, δ)-DP
	$\Delta_1(\mathbf{\Sigma})$	$\Delta_2(\mathbf{\Sigma})$
Fourier features	$\leq \sqrt{2}m$	$= \sqrt{m}$
$+\ \mathbf{\Omega}$ nonresonant	$= \sqrt{2}m$	$= \sqrt{m}$
Quadratic features	$= \|\mathbf{\Omega}\|_2^2$	$= \|\mathbf{\Omega}^T\|_{2\to 4}^2$
$+\ \mathbf{\Omega}$ union of orthogonal bases.	$= m/d$	No particular closed form.

(Results for the “replacement” neighboring relation can be found in the manuscript.)

Privacy results

Holds almost surely!

	Pure ε-DP	Approximate (ε, δ)-DP
	$\Delta_1(\mathbf{\Sigma})$	$\Delta_2(\mathbf{\Sigma})$
Fourier features	$\leq \sqrt{2}m$	$= \sqrt{m}$
$+ \mathbf{\Omega}$ nonresonant	$= \sqrt{2}m$	$= \sqrt{m}$
Quadratic features	$= \Vert\mathbf{\Omega}\Vert_2^2$	$= \Vert\mathbf{\Omega}^T\Vert_{2\to 4}^2$
$+ \mathbf{\Omega}$ union of orthogonal bases.	$= m/d$	No particular closed form.

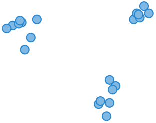
Order-4 tensor approximation problem (NP-hard)

(Results for the "replacement" neighboring relation can be found in the manuscript.)

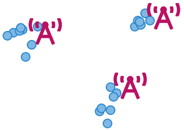
Different problems:

- obtaining upper bounds (easy);
- obtaining sharp bounds (🧩);
- computing numerically the bound (🧩 in some settings).

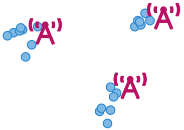
Experimental results



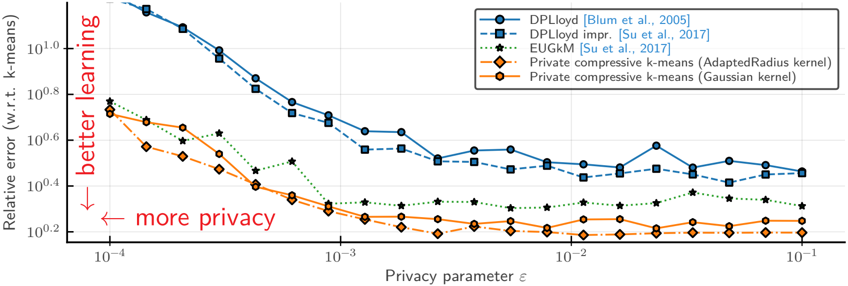
Experimental results



Experimental results

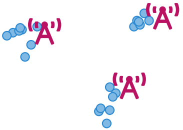


Example for a **clustering task:**

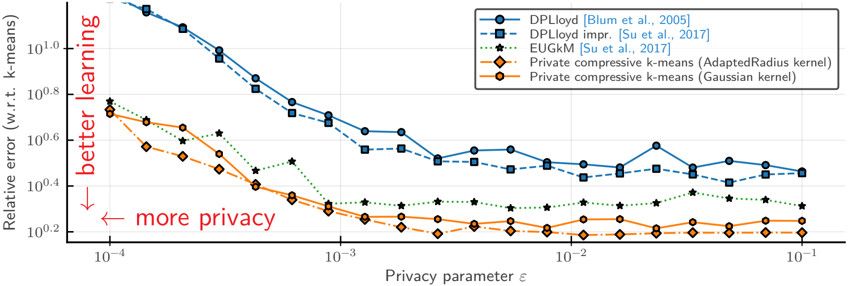


Gowalla dataset, $d = 2, n \approx 10^6$ (real GPS data) — medians over 100 trials.

Experimental results



Example for a **clustering task:**



Gowalla dataset, $d = 2, n \approx 10^6$ (real GPS data) — medians over 100 trials.

Observations:

- Competitive results with other methods from the literature.
- DPLloyd suffers from its "iterative" nature.

Role of the noise-to-signal ratio

Noise-to-signal ratio:

private sketch

$$\mathrm{NSR} \triangleq \frac{\mathbf{E}\|\mathbf{z}(\mathbf{X}) - \tilde{\mathbf{s}}\|_2^2}{\|\tilde{\mathbf{s}}\|^2}.$$

empirical sketch

Role of the noise-to-signal ratio

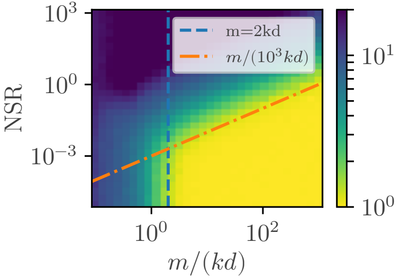
Noise-to-signal ratio:

$$\mathrm{NSR} \triangleq \frac{\mathbf{E}\|\mathbf{z}(\mathbf{X}) - \tilde{\mathbf{s}}\|_2^2}{\|\tilde{\mathbf{s}}\|^2}.$$

private sketch

empirical sketch

Empirical correlation (clustering task):



Color = relative error.

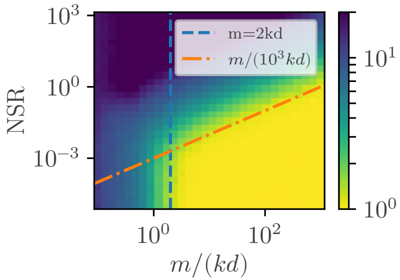
Recall: m = sketch size
$kd \approx$ number of parameters to learn

Role of the noise-to-signal ratio

Noise-to-signal ratio:

$$\mathrm{NSR} \triangleq \frac{\overbrace{\mathbf{E}\|\mathbf{z}(\mathbf{X}) - \tilde{\mathbf{s}}\|_2^2}^{\text{private sketch}}}{\underbrace{\|\tilde{\mathbf{s}}\|^2}_{\text{empirical sketch}}}.$$

Empirical correlation (clustering task):



Color = relative error.

For m large enough and fixed, the NSR is **good indicator of the error.**

In the manuscript:

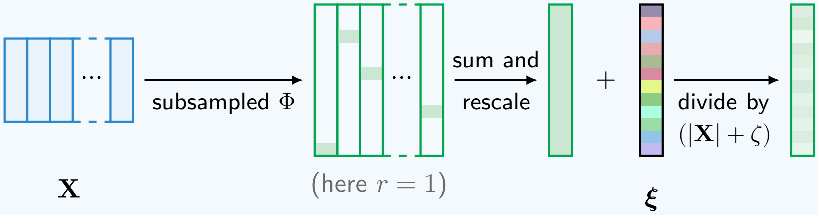
- ☑ **analytical expression** of the NSR;
- ☑ estimation of useful **regimes**;
- ☑ hyperparameter **tuning**.

Recall: m = sketch size
$kd \approx$ number of parameters to learn

Subsampling

Compute only $r < m$ features of Φ when sketching.

Proposed mechanism (with subsampling)



Goal 1: Reduce the computational complexity.
Goal 2: Reduce the amount of released information.

Other approach in the literature: subsampling the **data records**.
[Balle et al., 2018. **"Privacy Amplification by Subsampling"**]

Record subsampling vs feature supsampling

Lemma (Cf. manuscript)

Both types of subsampling "**do not improve** privacy" when properly rescaling the sketch. In most settings, they also "**do not reduce** privacy" (i.e. previous bounds remain valid).

Note: In spite of that, subsampling improves the complexity-privacy tradeoff.

Record subsampling vs feature supsampling

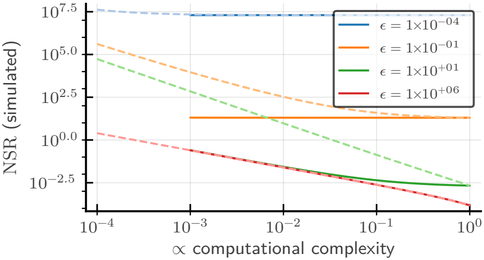
Lemma (Cf. manuscript)

Both types of subsampling "**do not improve** privacy" when properly rescaling the sketch. In most settings, they also "**do not reduce** privacy" (i.e. previous bounds remain valid).

Note: In spite of that, subsampling improves the complexity-privacy tradeoff.

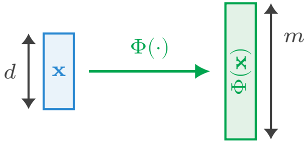
Legend:
— feature subsampling
- - record subsampling

Observation: feature subsampling might in some regimes yield a better utility!



Efficient Sketching using Structured Matrices

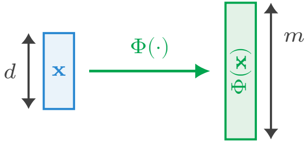
About computational efficiency



Recall: $\Phi(\mathbf{x}) = \rho(\mathbf{\Omega}^T\mathbf{x})$.

$\mathbf{\Omega}$ is a $d \times m$ random matrix.

About computational efficiency

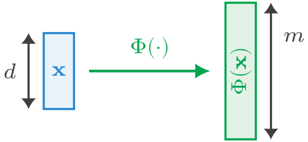


Recall: $\Phi(\mathbf{x}) = \rho(\mathbf{\Omega}^T\mathbf{x})$.

$\mathbf{\Omega}$ is a $d \times m$ random matrix.

$\rightarrow \mathcal{O}(nmd)$ computational complexity when $\mathbf{\Omega}$ is dense.

About computational efficiency



Recall: $\Phi(\mathbf{x}) = \rho(\mathbf{\Omega}^T\mathbf{x})$.

$\mathbf{\Omega}$ is a $d \times m$ random matrix.

$\rightarrow \mathcal{O}(nmd)$ computational complexity when $\mathbf{\Omega}$ is dense.

Use a structured matrix $\mathbf{\Omega}$.

Goals:

- Reduce the sketching/learning complexity (and runtimes).
- Reduce the storage cost.

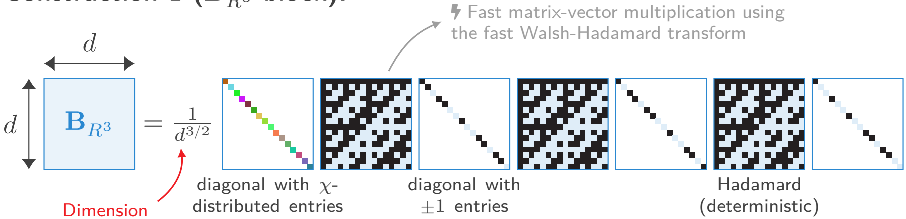
Building a square structured block

Standard examples of square structured matrices: Vandermonde, circulant, **Walsh-Hadamard**...

Building a square structured block

Standard examples of square structured matrices: Vandermonde, circulant, **Walsh-Hadamard**...

Construction 1 (B_{R^3} block):

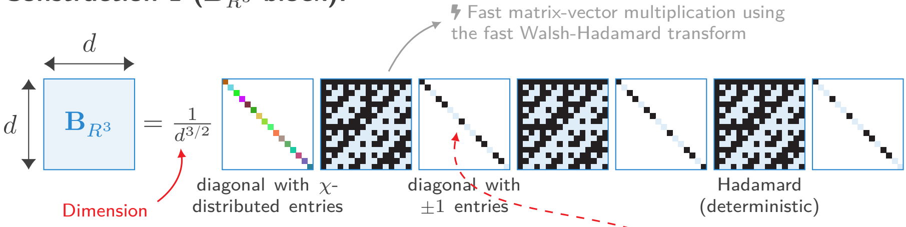


[Yu et al., 2016. **"Orthogonal Random Features"**]

Building a square structured block

Standard examples of square structured matrices: Vandermonde, circulant, **Walsh-Hadamard**...

Construction 1 (B_{R^3} block):



[Yu et al., 2016. **"Orthogonal Random Features"**]

Construction 2 (B_{GR^2} block): use Gaussian values in the third matrix.

(The normalization term must also be adapted.)

The matrix $\mathbf{\Omega}$ can be built by stacking such i.i.d. square blocks!

What do we gain?

"Compressive k-means"

In theory:

	CKM	Fast CKM	k-means
Time	$kd^2(n+k^2)$	$kd\ln(d)(n+k\ln(k))$	$ndkI$
Sketching	nkd^2	$nkd\ln(d)$	—
Learning (CL-OMP(R))	k^3d^2	$k^3d\ln(d)$	—
Space	$kd(d+n_b)$	kdn_b	nd
$\mathbf{\Omega}$	kd^2	kd	—
$\mathbf{\Omega}^T\mathbf{X}$ (by batch)	kdn_b	kdn_b	—

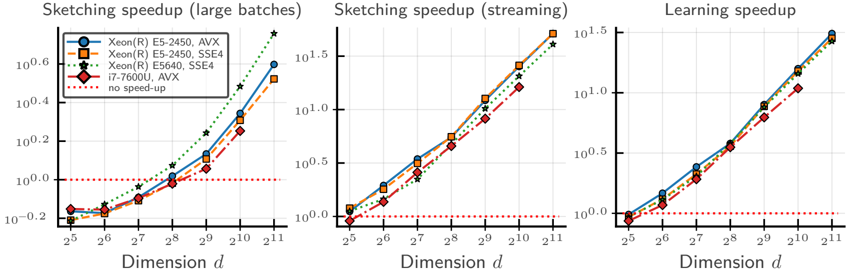
What do we gain?

"Compressive k-means"

In theory:

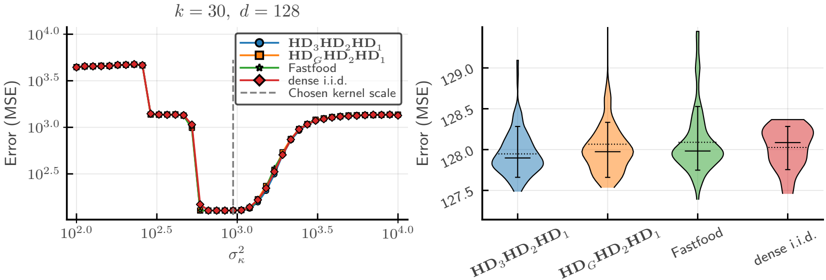
	CKM	Fast CKM	k-means
Time	$kd^2(n+k^2)$	$kd\ln(d)(n+k\ln(k))$	$ndkI$
Sketching	nkd^2	$nkd\ln(d)$	—
Learning (CL-OMP(R))	k^3d^2	$k^3d\ln(d)$	—
Space	$kd(d+n_b)$	kdn_b	nd
$\mathbf{\Omega}$	kd^2	kd	—
$\mathbf{\Omega}^T\mathbf{X}$ (by batch)	kdn_b	kdn_b	—

Speedup factors in practice:



Significative speed-ups, particularly for **small batches (e.g. for learning)**.

Impact on learning quality



(The 4 curves are all superimposed on the left).

[Le et al., 2013. **"Fastfood: Approximating Kernel Expansions in Loglinear Time"**]

Observations:

- No degradation of the overall performance.
- The median error is even slightly reduced when using structured operators.

Some Theoretical Elements

A Key Ingredient for Theory: the Mean Kernel

Any distribution on $\mathbf{\Omega}$ induces a mean **kernel** $\kappa(\mathbf{x}, \mathbf{y}) = \frac{1}{m}\mathbf{E}_{\Omega}\langle \Phi(\mathbf{x}), \Phi(\mathbf{y})\rangle$.

"Similarity measure"

A Key Ingredient for Theory: the Mean Kernel

Any distribution on $\mathbf{\Omega}$ induces a mean **kernel** $\kappa(\mathbf{x}, \mathbf{y}) = \frac{1}{m}\mathbf{E}_{\Omega}\langle \Phi(\mathbf{x}), \Phi(\mathbf{y})\rangle$.

"Similarity measure"

Dense matrix $\mathbf{\Omega}$ with i.i.d. standard normal entries

known

Gaussian mean kernel
$\kappa(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|_2^2}{2\sigma_\kappa^2}\right)$

A Key Ingredient for Theory: the Mean Kernel

Any distribution on $\boldsymbol{\Omega}$ induces a mean **kernel** $\kappa(\mathbf{x}, \mathbf{y}) = \frac{1}{m}\mathbf{E}_{\Omega}\langle\Phi(\mathbf{x}), \Phi(\mathbf{y})\rangle$.

"Similarity measure"

Dense matrix $\boldsymbol{\Omega}$ with i.i.d. standard normal entries

known

Structured matrix $\boldsymbol{\Omega}$ with $\mathbf{B}_{GR^2}$ blocks

See manuscript

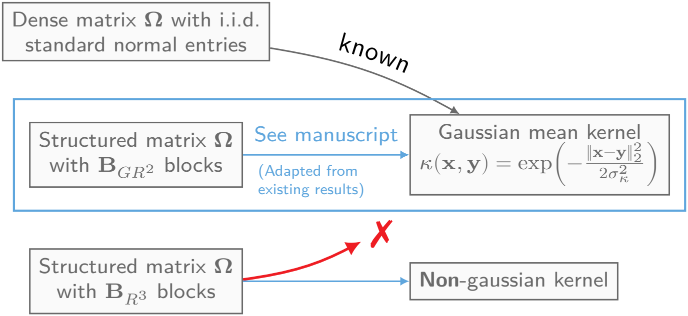
(Adapted from existing results)

Gaussian mean kernel $\kappa(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|_2^2}{2\sigma_\kappa^2}\right)$

A Key Ingredient for Theory: the Mean Kernel

Any distribution on $\mathbf{\Omega}$ induces a mean **kernel** $\kappa(\mathbf{x}, \mathbf{y}) = \frac{1}{m}\mathbf{E}_{\Omega}\langle\Phi(\mathbf{x}), \Phi(\mathbf{y})\rangle$.

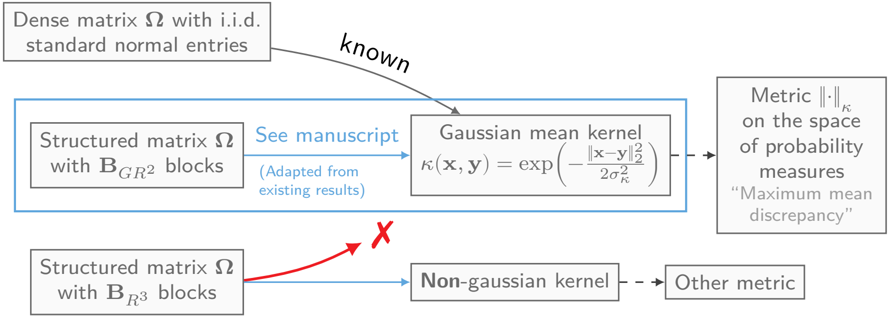
"Similarity measure"



A Key Ingredient for Theory: the Mean Kernel

Any distribution on $\boldsymbol{\Omega}$ induces a mean **kernel** $\kappa(\mathbf{x}, \mathbf{y}) = \frac{1}{m}\mathbf{E}_{\Omega}\langle \Phi(\mathbf{x}), \Phi(\mathbf{y})\rangle$.

"Similarity measure"



Towards Theoretical Guarantees

Goal: establish a lower restricted isometry property (LRIP) of the form

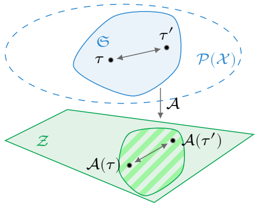
$$\forall \tau, \tau' \in \mathfrak{S}, \; \|\tau - \tau'\|_{\mathcal{L}(\mathcal{H})} \lesssim \|\mathcal{A}(\tau) - \mathcal{A}(\tau')\|_2$$

Low-dimensional model

Metric related to the learning task

[Gribonval et al., 2020. ***Compressive Statistical Learning with Random Feature Moments***]

LRIP



Towards Theoretical Guarantees

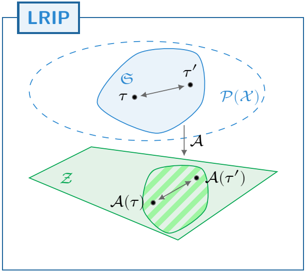
Goal: establish a lower restricted isometry property (LRIP) of the form

$$\forall \tau, \tau' \in \mathfrak{S},\ \|\tau - \tau'\|_{\mathcal{L}(\mathcal{H})} \lesssim \|\mathcal{A}(\tau) - \mathcal{A}(\tau')\|_2$$

Low-dimensional model

Metric related to the learning task

[Gribonval et al., 2020. ***Compressive Statistical Learning with Random Feature Moments***]



Strategy:

- Establish a LRIP for the metric $\|\cdot\|_\kappa$ associated to the **mean** kernel κ:

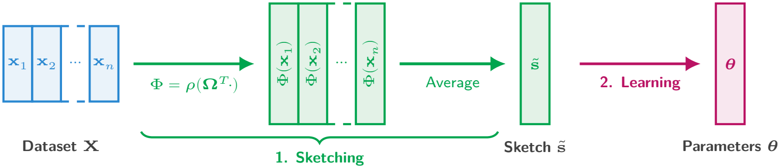
$$\forall \tau, \tau' \in \mathfrak{S},\ \|\tau - \tau'\|_{\mathcal{L}(\mathcal{H})} \lesssim \|\tau - \tau'\|_\kappa .$$

✓ for $\mathbf{B}_{GR^2}$
(more technical for $\mathbf{B}_{R^3}$)

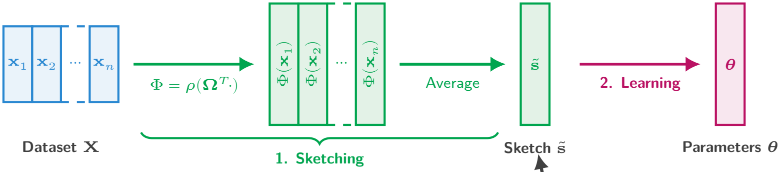
- Study the **concentration** of $\|\mathcal{A}(\tau) - \mathcal{A}(\tau')\|_2$ w.r.t. $\|\tau - \tau'\|_\kappa$.
  - Establish a pointwise concentration result. Work in progress...
  - Use covering arguments to get a uniform result.

Summary and Perspectives

Summary of contributions



Summary of contributions

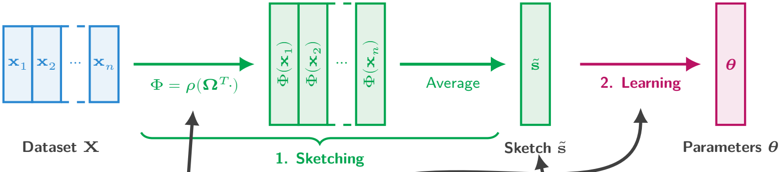


1. Privacy Preservation

- ☑ a noisy sketching mechanism;
- ☑ **sharp** DP guarantees
  (for Fourier and quadratic sketches);
- ☑ a new **subsampling** mechanism;
- ☑ an analysis of the NSR
  (+ application to parameters tuning).

Perspectives: Investigate other privacy definitions and attack models.

Summary of contributions



2. Reduction of the Complexity

- ☑ a **structured design** for the random sketching matrix;
- ☑ significant **empirical speedups** (for both sketching/learning);
- ☑ some preliminary guarantees.

Perspectives: control the concentration of the kernel; investigate further the orthogonal setting.

1. Privacy Preservation

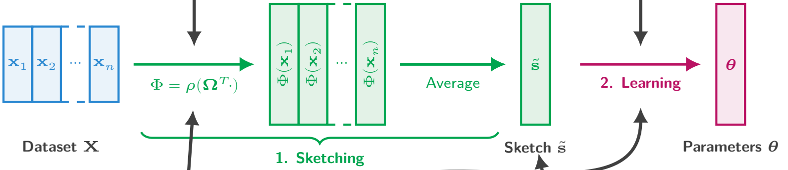
- ☑ a noisy sketching mechanism;
- ☑ **sharp** DP guarantees (for Fourier and quadratic sketches);
- ☑ a new **subsampling** mechanism;
- ☑ an analysis of the NSR (+ application to parameters tuning).

Perspectives: Investigate other privacy definitions and attack models.

Summary of contributions

Sketching operator design
Kernel scale tuning for clustering, considerations for extension to broader tasks.

Algorithmic recovery
A message-passing algorithm for compressive clustering.



2. Reduction of the Complexity

- ☑ a **structured design** for the random sketching matrix;
- ☑ significant **empirical speedups** (for both sketching/learning);
- ☑ some preliminary guarantees.

Perspectives: control the concentration of the kernel; investigate further the orthogonal setting.

1. Privacy Preservation

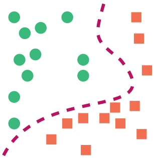
- ☑ a noisy sketching mechanism;
- ☑ **sharp** DP guarantees (for Fourier and quadratic sketches);
- ☑ a new **subsampling** mechanism;
- ☑ an analysis of the NSR (+ application to parameters tuning).

Perspectives: Investigate other privacy definitions and attack models.

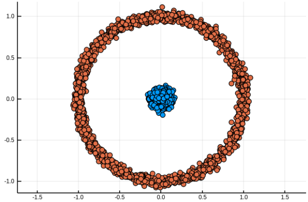
Perspectives

Directions for future work:

- Handling new learning tasks.
  In particular, addressing supervised learning tasks.
  (cf. Chapter 10 of the manuscript.)



- Working with intermediate features
  ... starting with kernel k-means and kernel PCA!
  Akin to two-layers random neural networks.



- Sketching structured data.
  E.g. graphs, images, etc.
- Obtaining guarantees on the heuristics used for the inverse problem.

List of publications

- Privacy-preserving compressive learning
  - Antoine Chatalic et al. "Compressive Learning with Privacy Guarantees". Submitted to Information and Inference (under review)., Mar. 3, 2020
  - Vincent Schellekens et al. "Differentially Private Compressive K-Means". In: 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). May 2019, pp. 7933–7937
  - Vincent Schellekens et al. "Compressive K-Means with Differential Privacy". In: SPARS Workshop. July 1, 2019
- Efficient compressive learning
  - Antoine Chatalic et al. "Large-Scale High-Dimensional Clustering with Fast Sketching". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018
  - Antoine Chatalic and Rémi Gribonval. "Learning to Sketch for Compressive Clustering". In: International Traveling Workshop on Interactions between Low-Complexity Data Models and Sensing Techniques (iTWIST). June 2020
- Compressive clustering with message passing
  - Evan Byrne et al. "Sketched Clustering via Hybrid Approximate Message Passing". In: *IEEE Transactions on Signal Processing* 67.17 (Sept. 2019), pp. 4556–4569
- Overview articles
  - Antoine Chatalic et al. "Projections aléatoires pour l'apprentissage compressif". In: Gretsi. Aug. 26, 2019
  - Rémi Gribonval et al. "Sketching Datasets for Large-Scale Learning (Long Version)". Aug. 4, 2020 (under review)

... and coding ... and teaching.

Thank you!