Sketching for Large-Scale Learning of Mixture Models

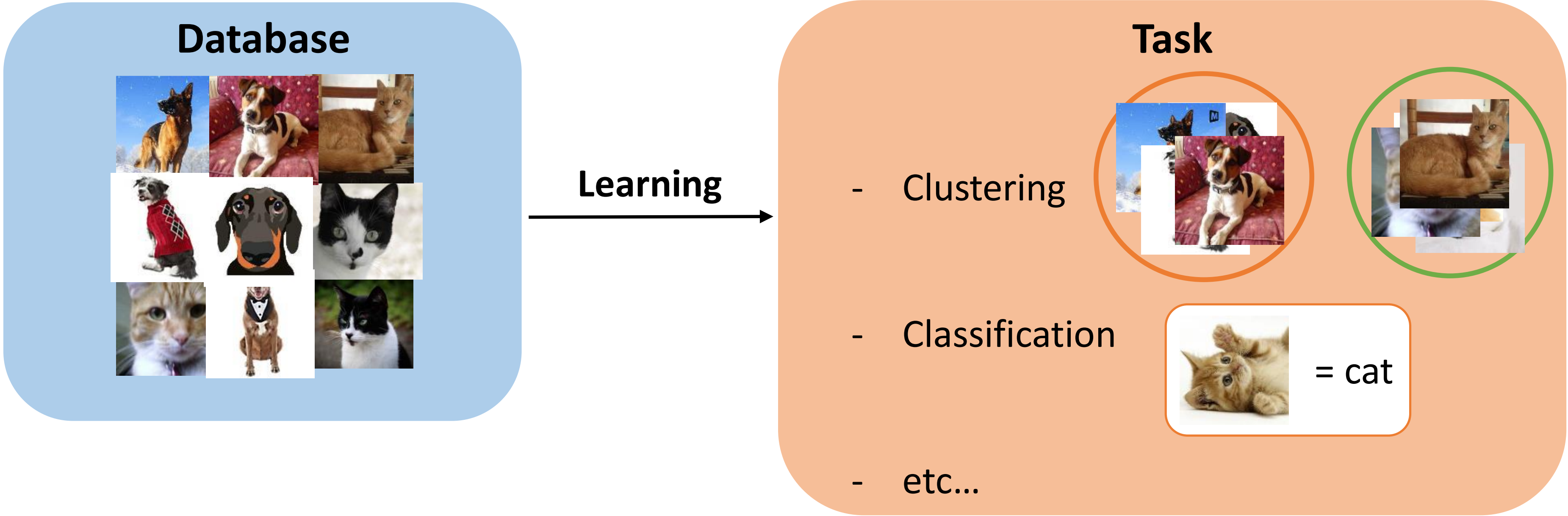
Nicolas Keriven

Ecole Normale Supérieure (Paris)
CFM-ENS chair in Data Science, « Laplace » post-doc

(thesis with Rémi Gribonval at Inria Rennes)

Dec. 8th 2017

Context: machine learning



Database

Learning

Task

- Clustering
- Classification
- etc…

= cat

Context: machine learning

Large database

Large elements

Billions of elements

Learning →

Task

- Clustering

- Classification

= cat

- etc…

Context: machine learning

Large **database**

Large elements
Billions of elements

Learning →

Slow, costly

Task

- Clustering

- Classification

= cat

- etc…

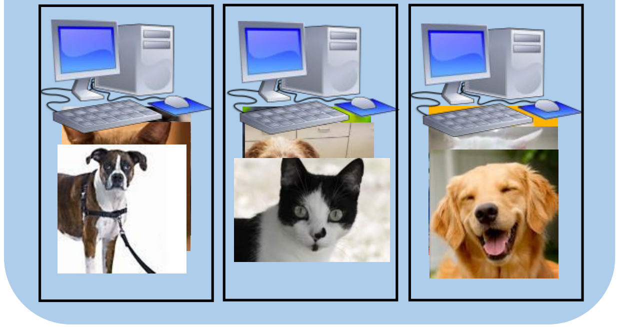
Context: machine learning

Large database

Large elements

Billions of elements

Distributed database



Learning

Slow, costly

Task

- Clustering

- Classification

= cat

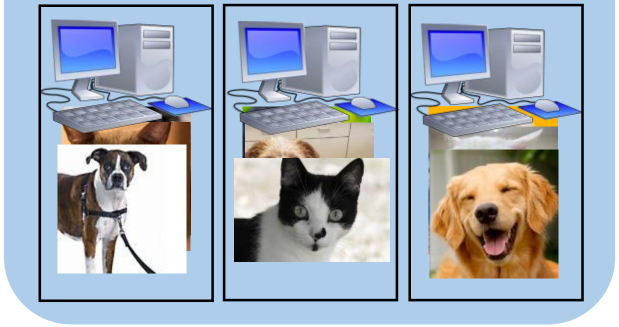
- etc…

Context: machine learning

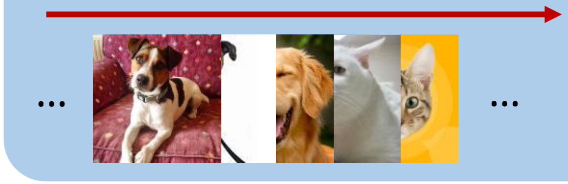
Large **database**

Large elements
Billions of elements

Distributed database



Data *Stream*



Learning

Slow, costly

Task

- Clustering

- Classification

= cat

- etc…

Context: machine learning

Large **database**

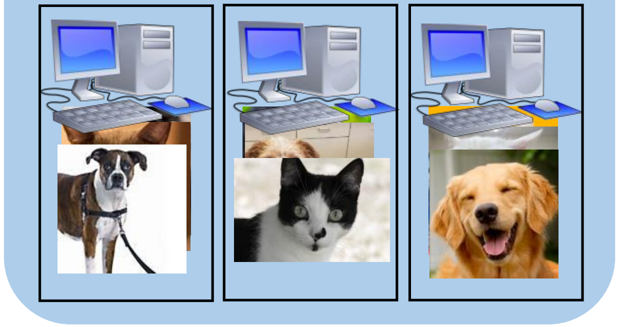
Large elements

Billions of elements

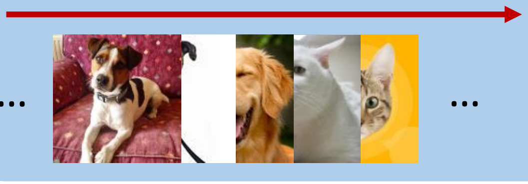
Learning

Slow, costly

Distributed **database**



Data *Stream*



… …

Task

- Clustering

- Classification

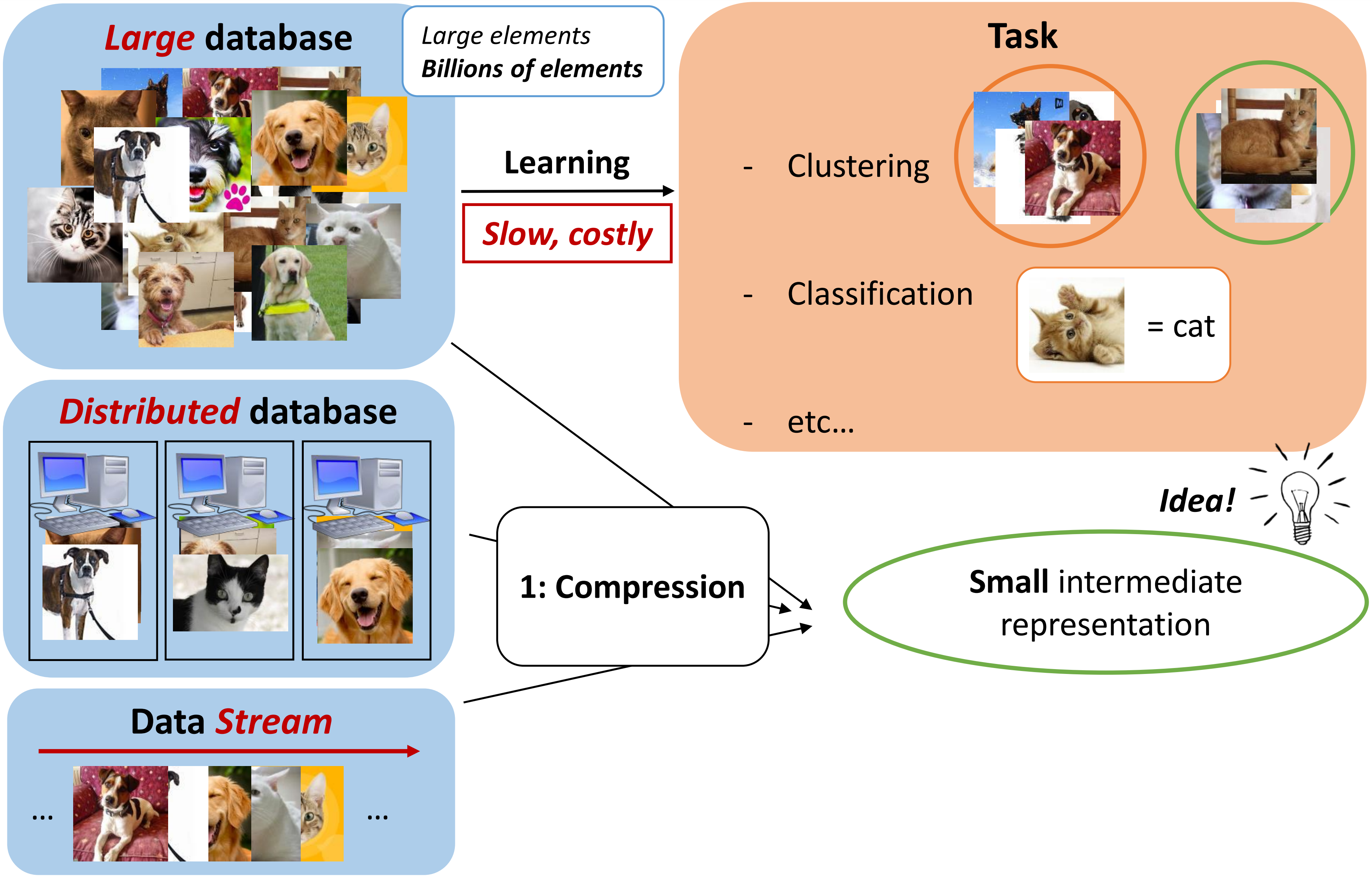
= cat

- etc…

Idea!

Small intermediate representation

Context: machine learning



Large **database**

Large elements
Billions of elements

Learning

Slow, costly

Task

- Clustering
- Classification = cat
- etc…

Distributed **database**

Data *Stream*

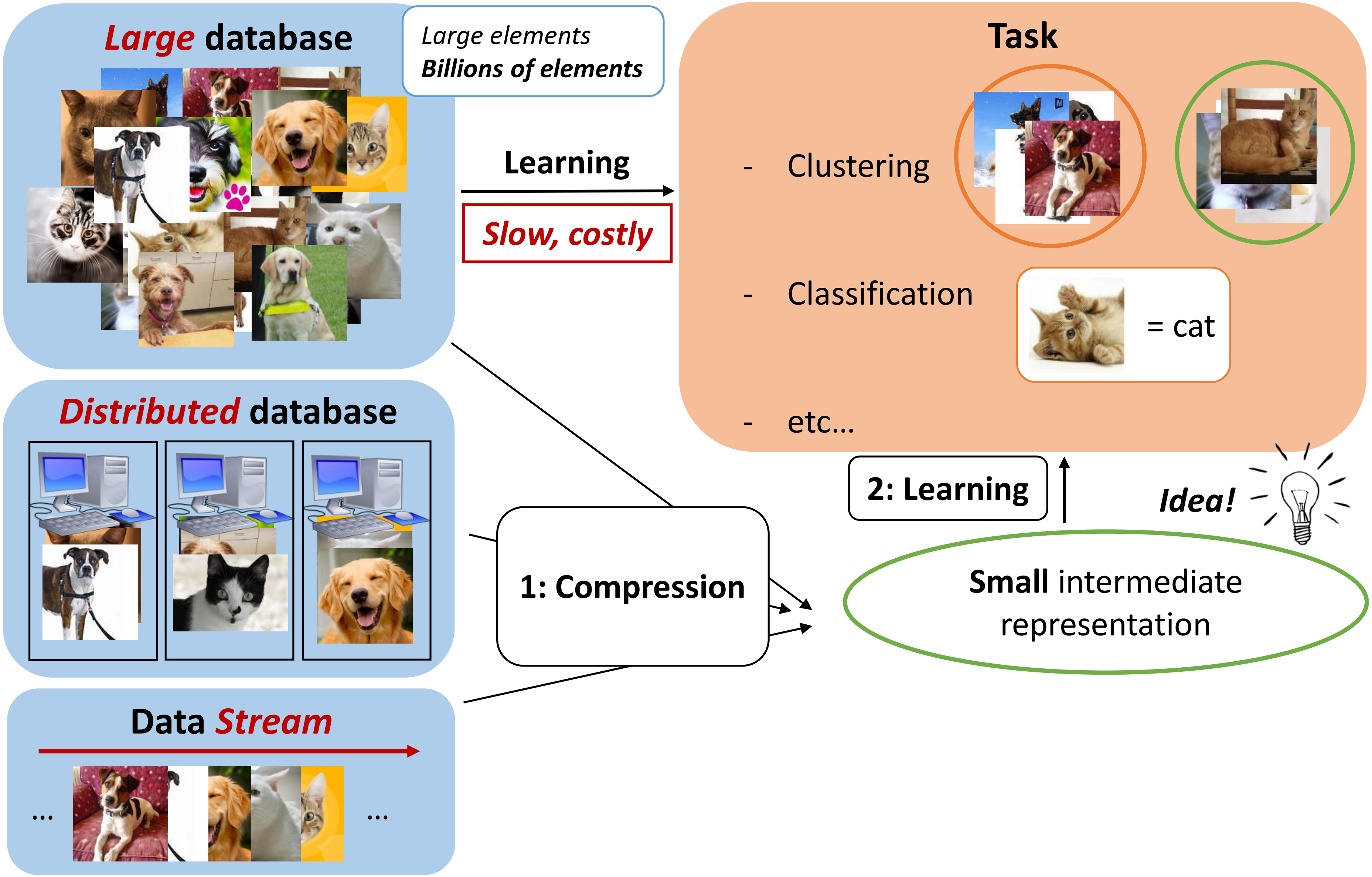
… …

1: Compression

Idea!

Small intermediate representation

Context: machine learning



Large database

Large elements
Billions of elements

Learning

Slow, costly

Task

- Clustering
- Classification = cat
- etc…

Distributed database

Data _Stream_

… …

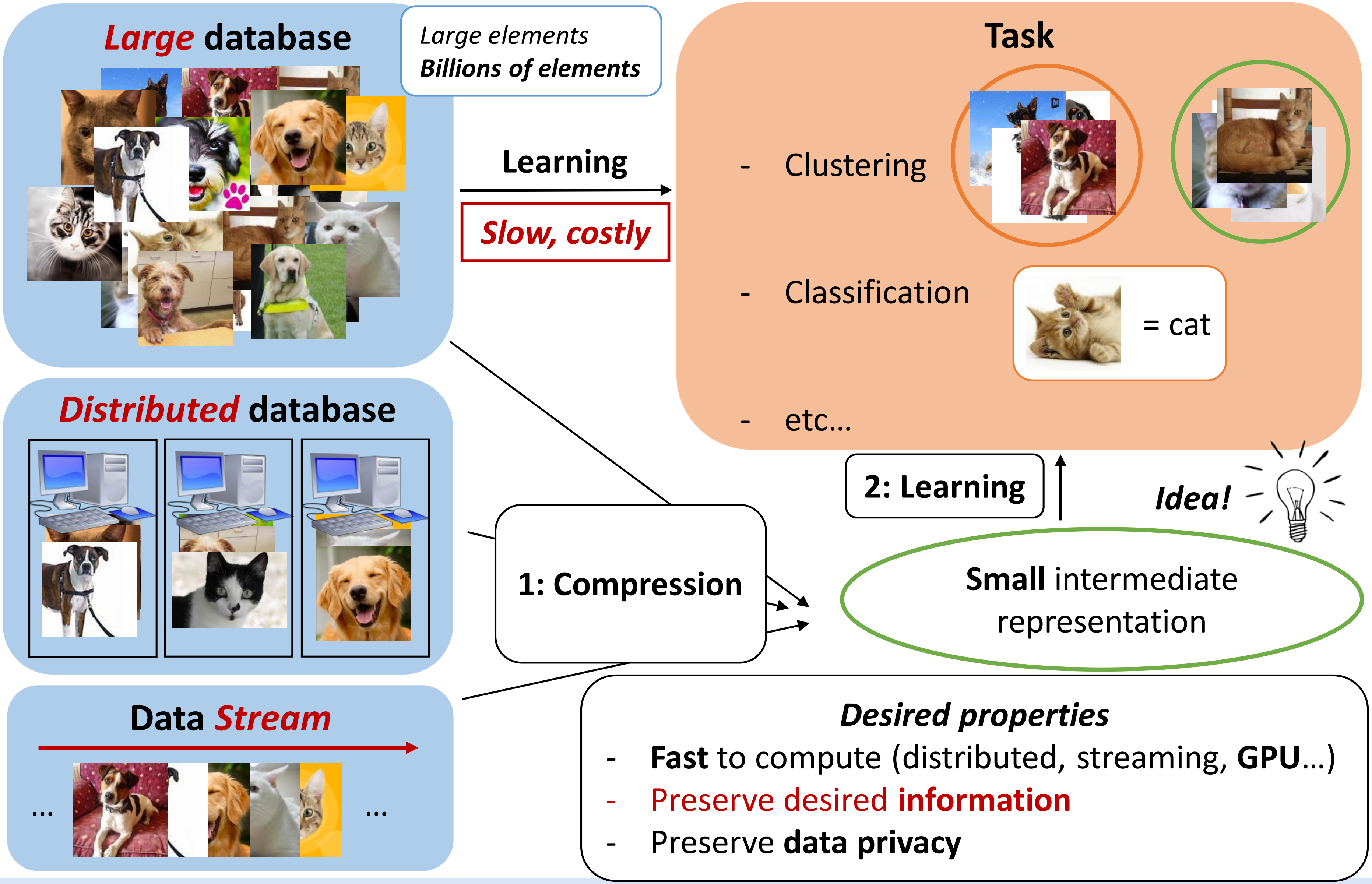
1: Compression

2: Learning

Idea!

Small intermediate representation

Context: machine learning



Large **database**

Large elements

Billions of elements

Learning

Slow, costly

Distributed **database**

Data ***Stream***

… …

Task

- Clustering
- Classification = cat
- etc…

1: Compression

2: Learning

Idea!

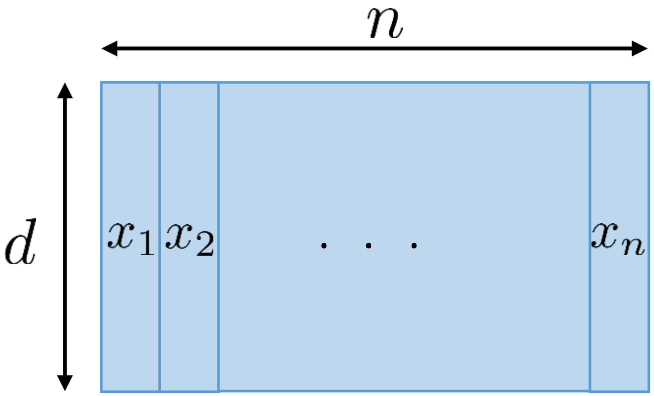
Small intermediate representation

Desired properties

- **Fast** to compute (distributed, streaming, **GPU**…)
- Preserve desired **information**
- Preserve **data privacy**

Three compression schemes

Feature extraction

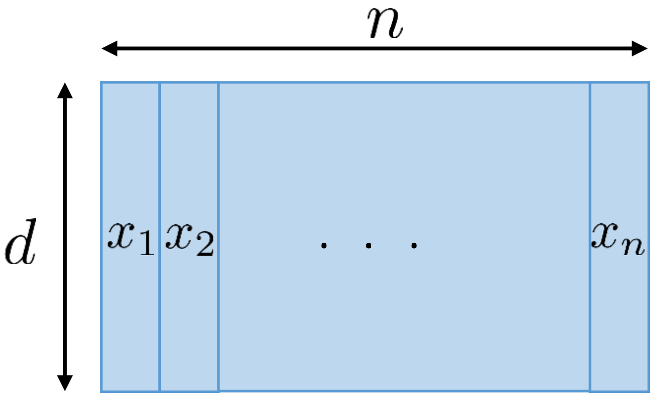


Data = Collection of vectors

Three compression schemes

Database

Feature extraction



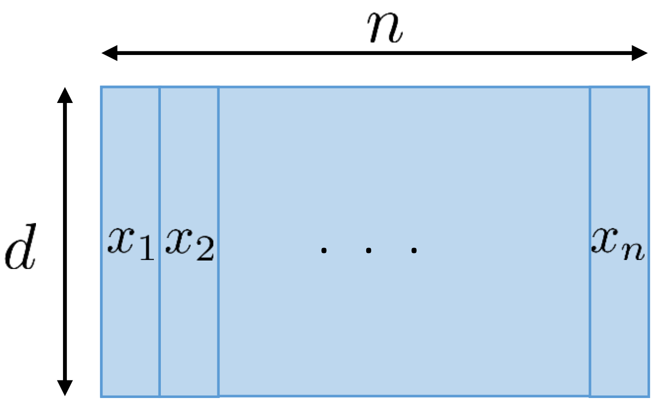
Data = Collection of vectors

Compression ?

Three compression schemes

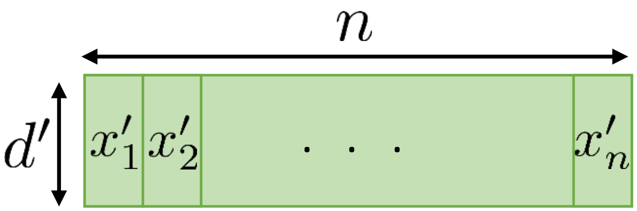
Database

Feature extraction →



Data = Collection of vectors

Compression ?



Dimensionality reduction

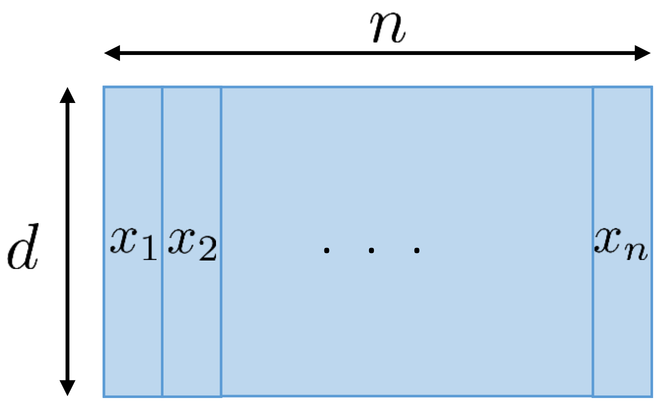
See eg *[Calderbank 2009, Boutsidis 2010]*

- Random Projection
- Feature selection

Three compression schemes

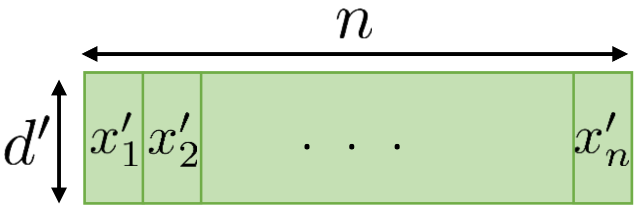
Database

Feature extraction →



Data = Collection of vectors

Compression ?



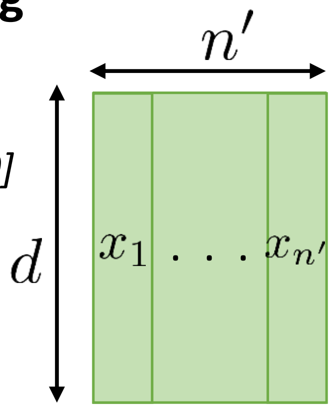
Dimensionality reduction

See eg *[Calderbank 2009, Boutsidis 2010]*

- Random Projection
- Feature selection

Subsampling coresets

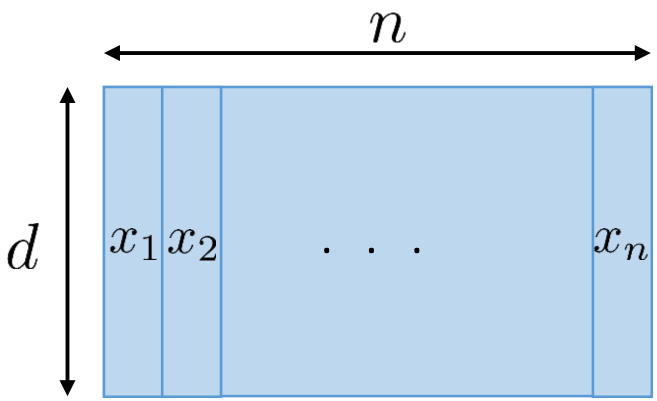
See eg *[Feldman 2010]*



- Uniform sampling (naive)
- Adaptive sampling…

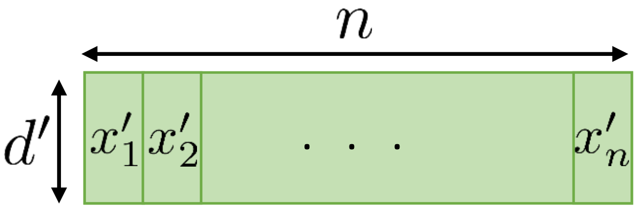
Three compression schemes

Feature extraction



Data = Collection of vectors

Compression ?



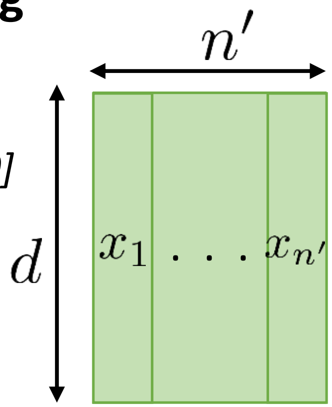
Dimensionality reduction

See eg *[Calderbank 2009, Boutsidis 2010]*

- Random Projection
- Feature selection

Subsampling coresets

See eg *[Feldman 2010]*

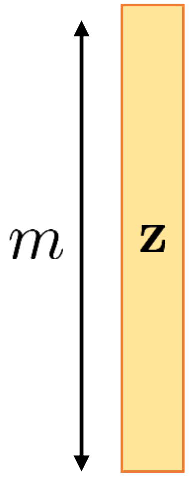


- Uniform sampling (naive)
- Adaptive sampling…

Linear sketch

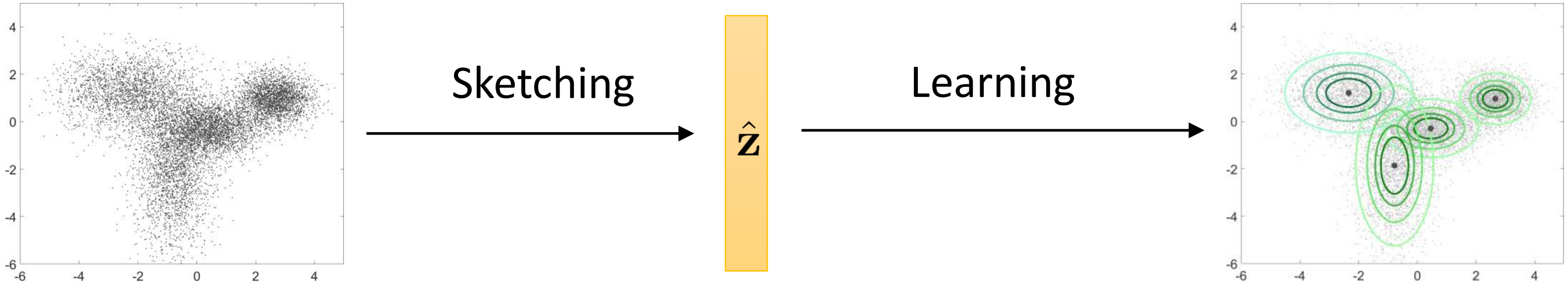
See *[Thaper 2002] [Cormode 2011]*

Distributed, streaming



- Hash tables, histograms
- **Sketching for learning ?**

Sketch learning ?

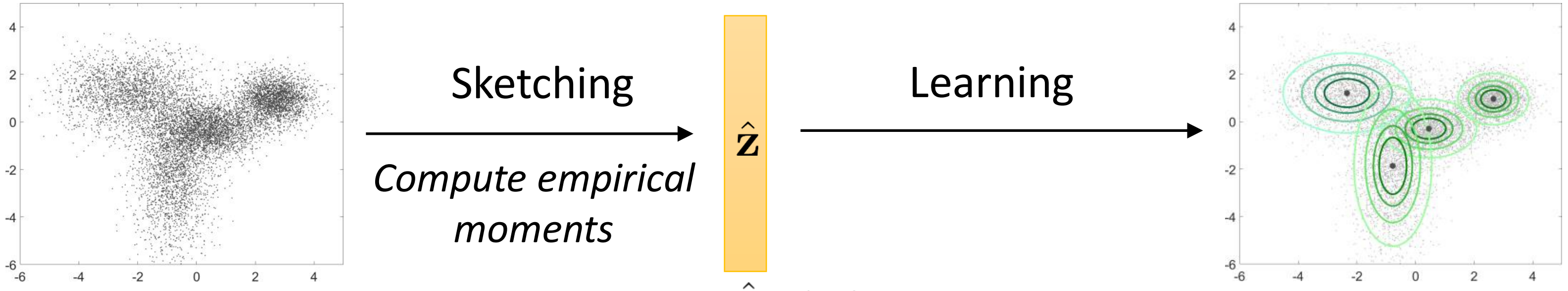


Sketching

$\hat{\mathbf{z}}$

Learning

Sketch learning ?



Sketching

Compute empirical moments

Learning

$$\hat{\mathbf{z}} = \hat{\mathbb{E}}\Phi(X)$$

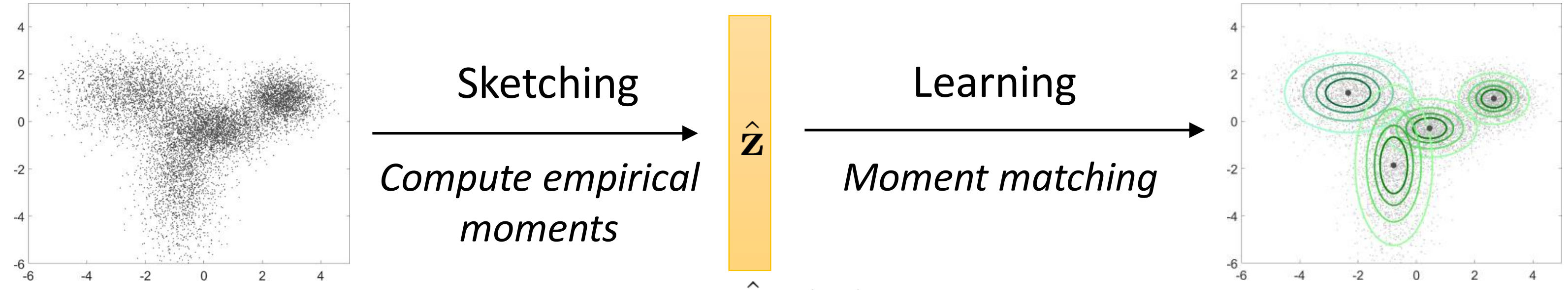
Observation: ***necessarily…***

Any ***linear*** sketch = **empirical moments**

$$\boxed{\hat{\mathbf{z}} = \hat{\mathbb{E}}\Phi(X)} = \frac{1}{n}\sum_i \Phi(x_i)$$

$$\Phi : \mathbb{R}^d \to \mathbb{C}^m$$

Sketch learning ?



Sketching

Compute empirical moments

$\hat{\mathbf{z}}$

Learning

Moment matching

$$\hat{\mathbf{z}} = \hat{\mathbb{E}}\Phi(X)$$

Observation: *necessarily*…

Any *linear* sketch = **empirical moments**

$$\boxed{\hat{\mathbf{z}} = \hat{\mathbb{E}}\Phi(X)} = \frac{1}{n}\sum_i \Phi(x_i)$$

$$\Phi : \mathbb{R}^d \to \mathbb{C}^m$$

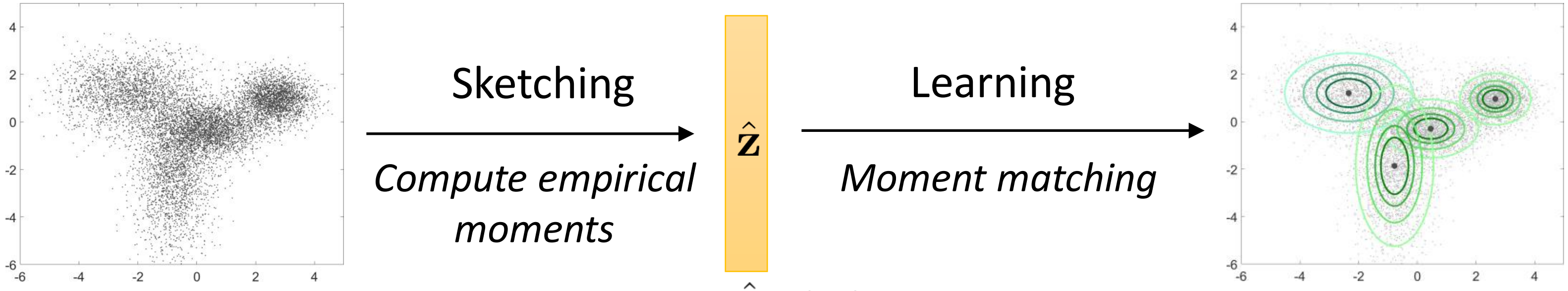
… hence:

Sketch learning = moment matching

$$\boxed{\min_\theta \|\hat{\mathbf{z}} - \mathbb{E}_\theta \Phi(X)\|}$$

True moments (param. θ *)*

Sketch learning ?



Sketching

Compute empirical moments

$\hat{\mathbf{z}}$

Learning

Moment matching

$$\hat{\mathbf{z}} = \hat{\mathbb{E}}\Phi(X)$$

Observation: *necessarily…*

Any *linear* sketch = **empirical moments**

$$\boxed{\hat{\mathbf{z}} = \hat{\mathbb{E}}\Phi(X)} = \frac{1}{n}\sum_i \Phi(x_i)$$

$$\Phi : \mathbb{R}^d \rightarrow \mathbb{C}^m$$

… hence:

Sketch learning = moment matching

$$\boxed{\min_\theta \left\|\hat{\mathbf{z}} - \mathbb{E}_\theta\Phi(X)\right\|}$$

True moments (param. θ *)*

Good empirical properties of the « sketching » function Φ *[Bourrier 2013]*

- **« Sufficient » dimension** m **(size of the sketch)**
- **Randomly designed (convenient, only mild training)**

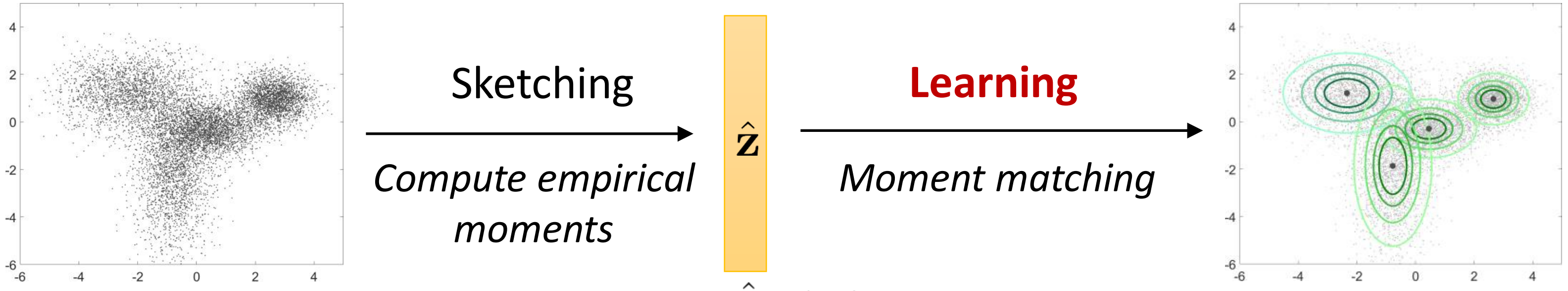
Outline

① Illustration: Sketched Mixture Model Estimation

② A Compressive Sensing analysis

③ Conclusion, outlooks

Algorithm

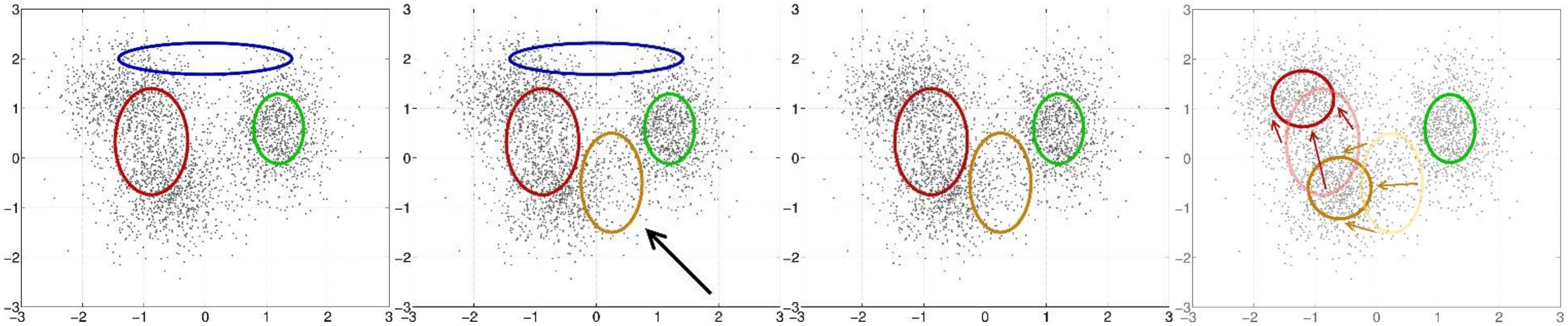


$$\hat{\mathbf{z}} = \hat{\mathbb{E}}\Phi(X)$$

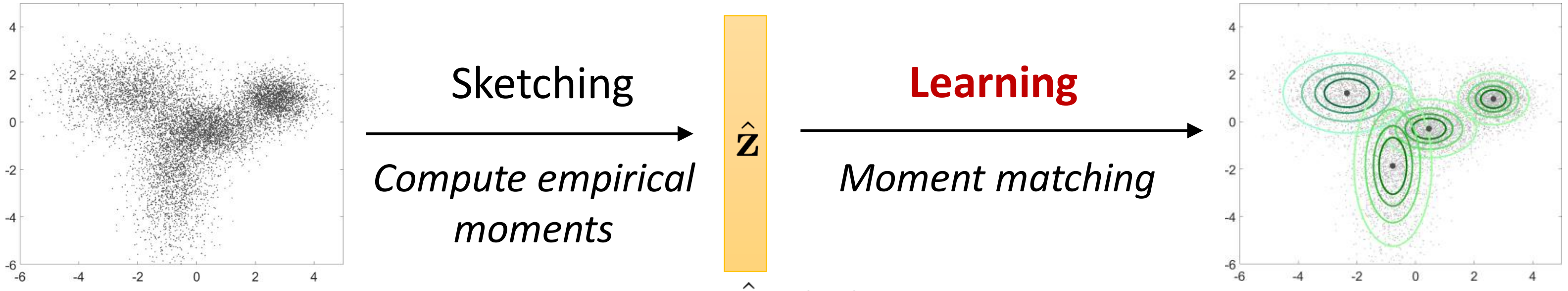
Algorithm for mixture models: **Compressive Learning OMPR (CL-OMPR)**

Continuous (off-the-grid) adaptation of *Orthogonal Matching Pursuit with Replacement*

[Jain 2011]



Algorithm



Sketching

Compute empirical moments

$\hat{\mathbf{z}}$

Learning

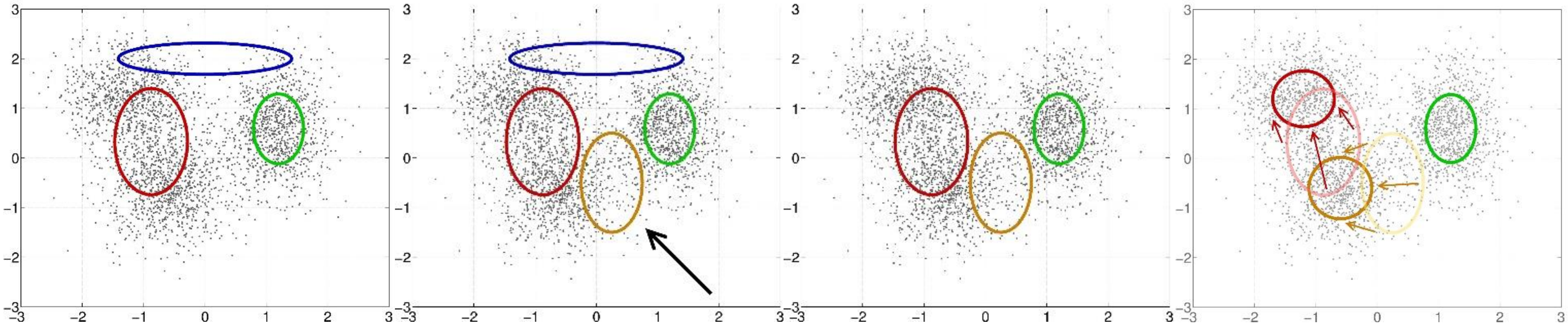
Moment matching

$$\hat{\mathbf{z}} = \hat{\mathbb{E}}\Phi(X)$$

Algorithm for mixture models: Compressive Learning OMPR (CL-OMPR)

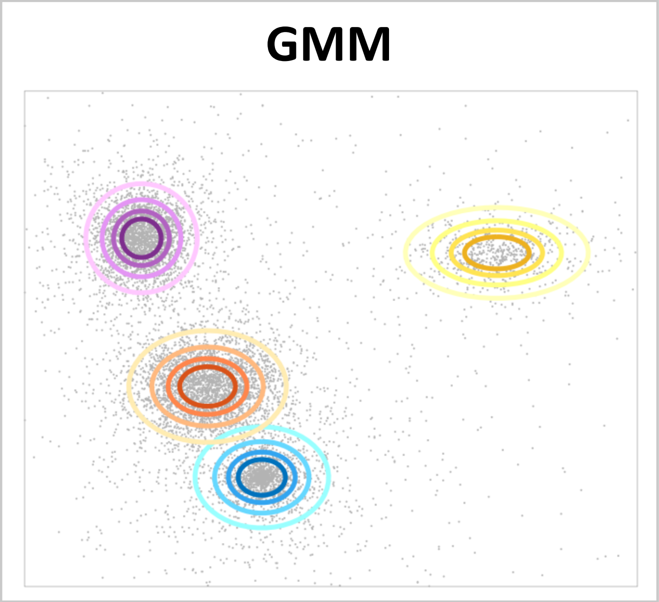
Continuous (off-the-grid) adaptation of *Orthogonal Matching Pursuit with Replacement*

[Jain 2011]

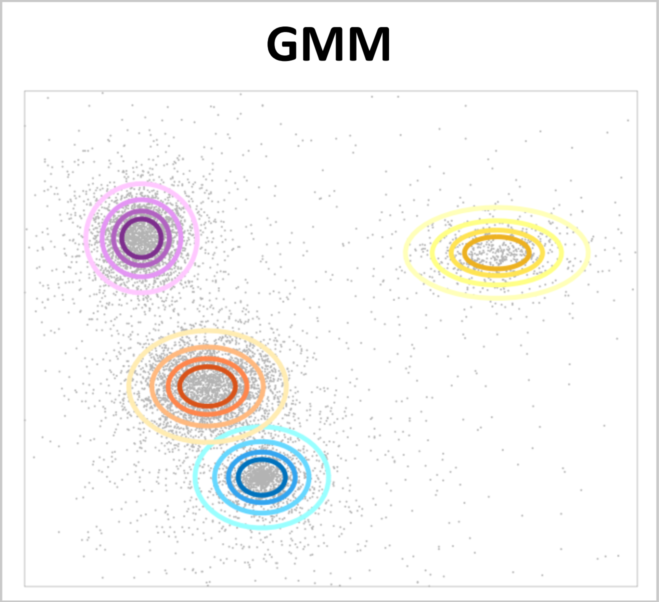


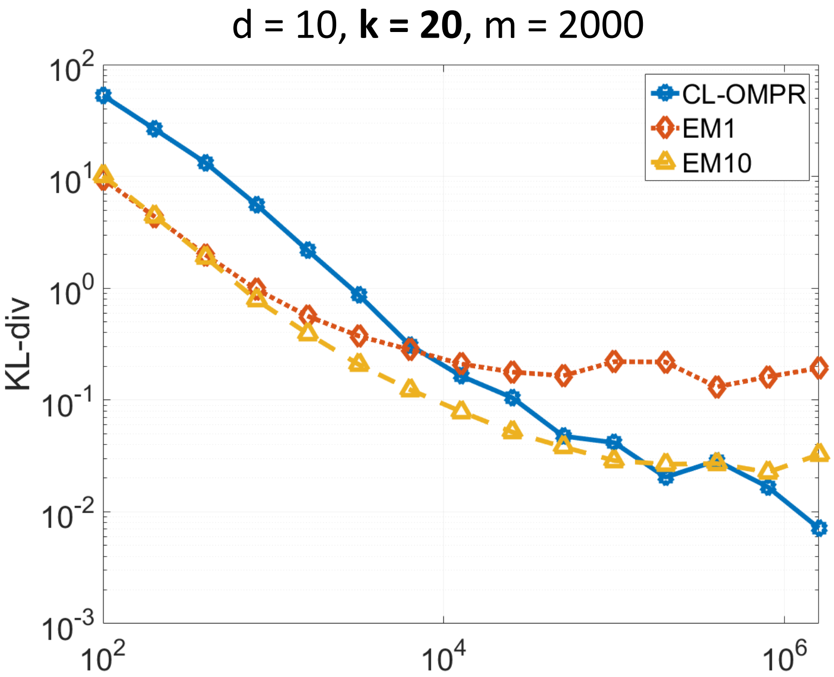
With Φ = (random) fourier sampling, applicable to any mixture model with an analytic expression for the characteristic function

Gaussian mixture models



Gaussian mixture models

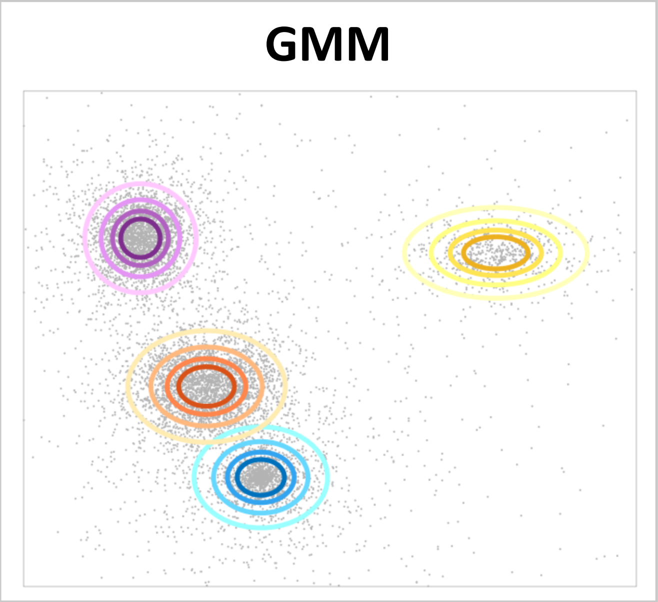


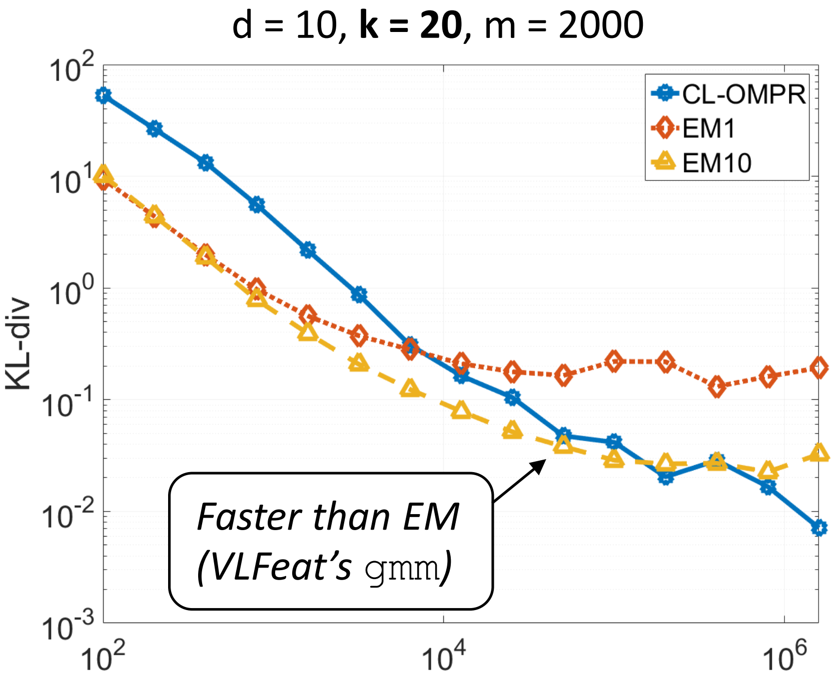


Error

Size of database

Gaussian mixture models



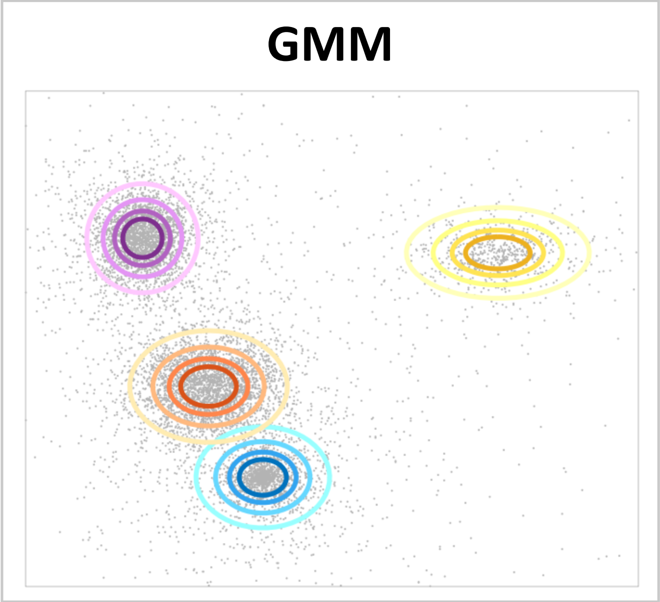


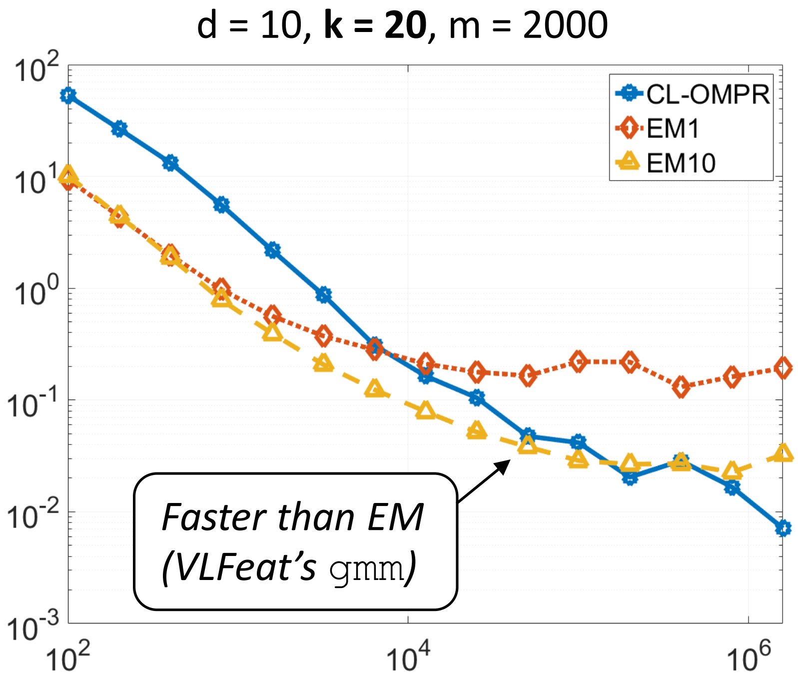
Error

Faster than EM (VLFeat's `gmm`*)*

Size of database

Gaussian mixture models


GMM



Error

Faster than EM (VLFeat's `gmm`*)*

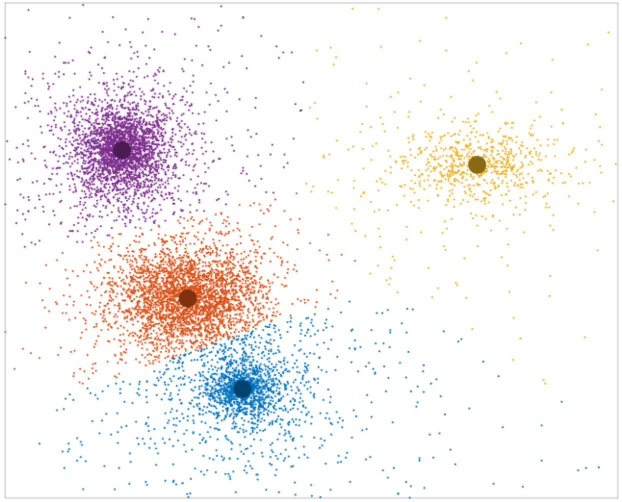
Size of database

Application: ***speaker verification*** *[Reynolds 2000]* (d=12, k=64)

- EM on 300 000 vectors : **29.53**
- **20kB** sketch computed on **50GB** database: **28.96**

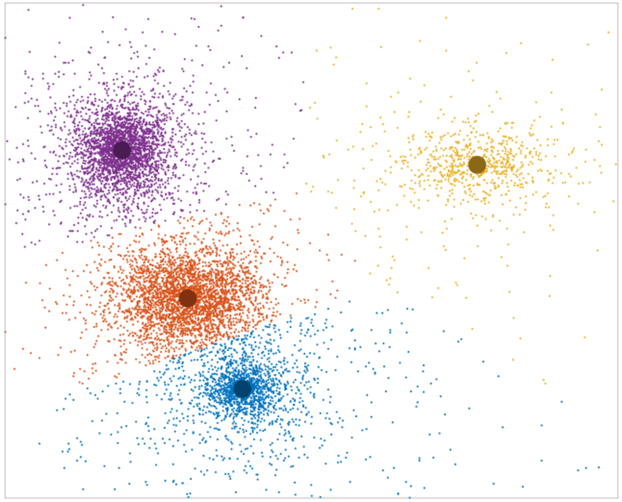
Compressive k-means [Keriven et al 2017]

Mixture of Diracs



Compressive k-means [Keriven et al 2017]

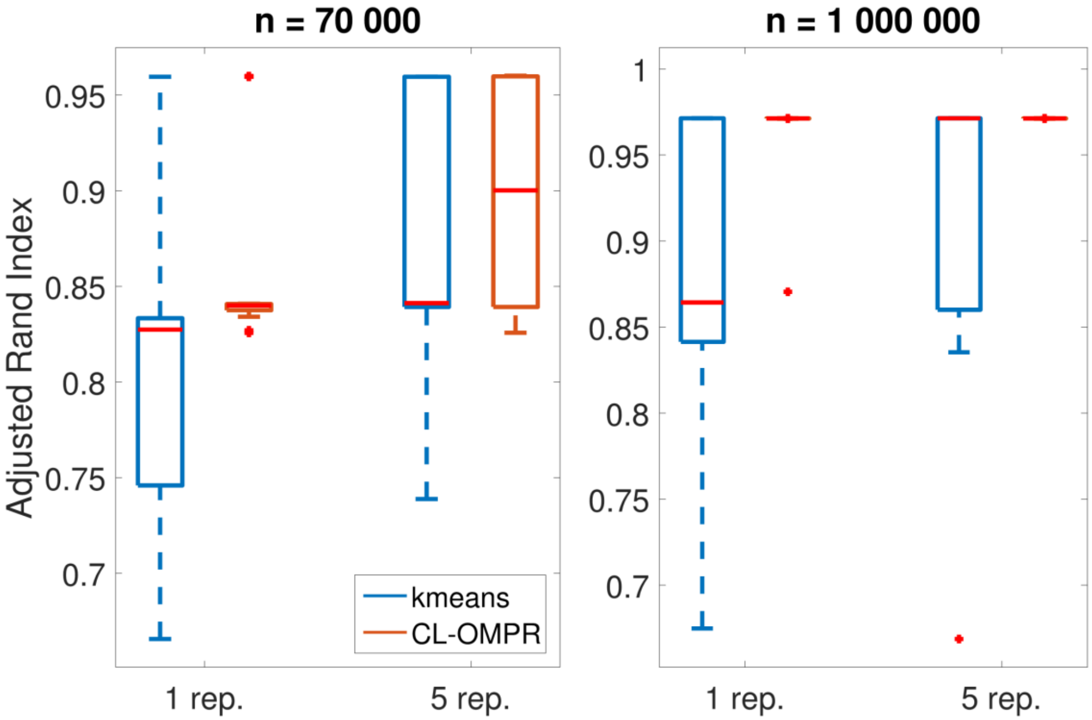
Mixture of Diracs



Application: Spectral clustering
for MNIST classification [Uw 2001]

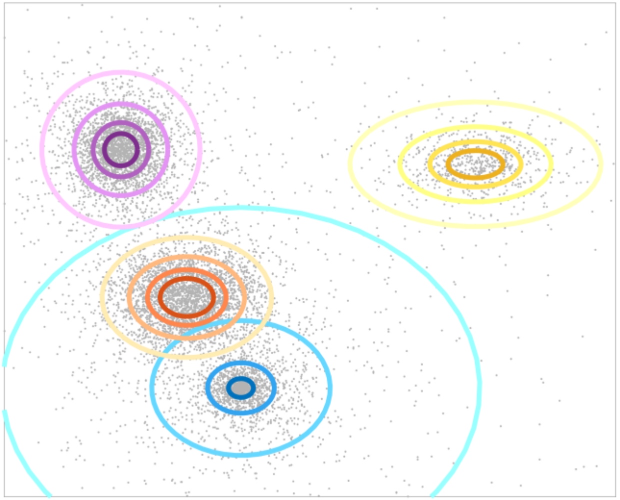
(d=10, k=10, m=1000)

Classif. Perf.



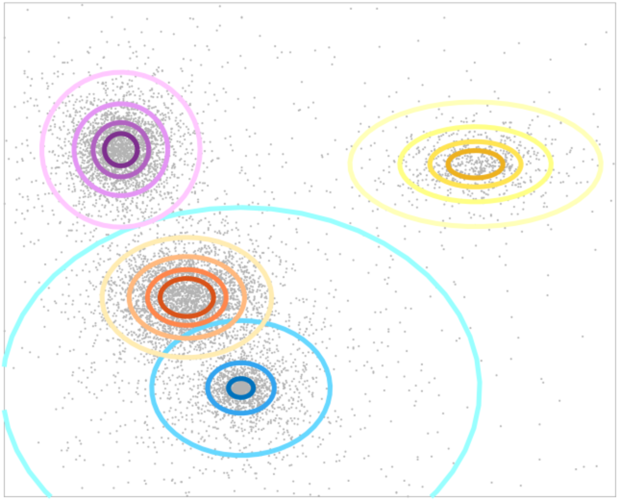
Mixtures of alpha-stable distribution

Mixture of stable dist.



Mixtures of alpha-stable distribution

Mixture of stable dist.



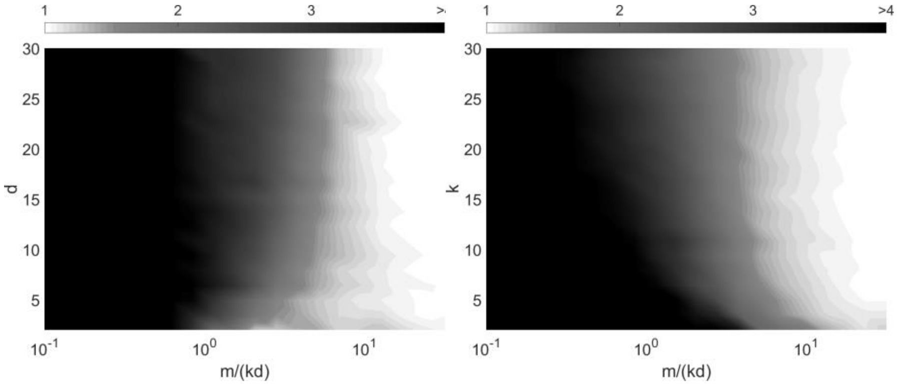
***Application**: audio source separation* *[submitted]*

Model: hybrid between rank-1 alpha-stable and Gaussian noise…

	SDR (dB)	SIR (dB)	MER (dB)
Oracle	8.33 ± 3.16	18.3 ± 4.13	N/A
Gaussian (EM)	3.50 ± 2.87	9.04 ± 4.92	12.3 ± 11.0
CF-α	$\mathbf{4.11 \pm 2.59}$	$\mathbf{9.17 \pm 3.51}$	$\mathbf{12.65 \pm 9.73}$

How big a sketch ?

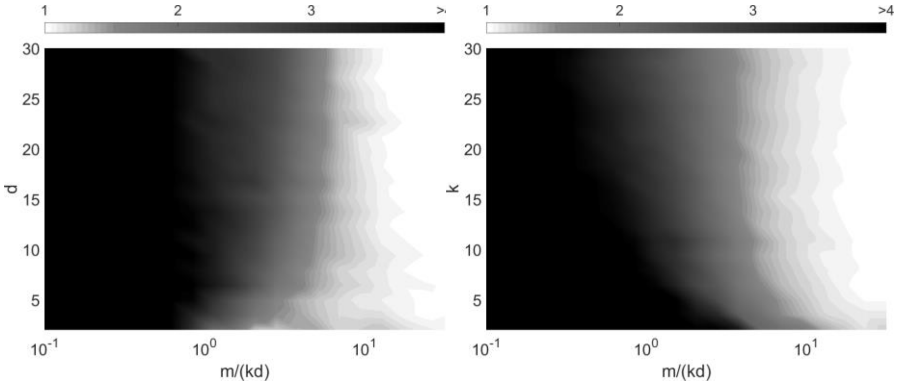
k-means



Relative sketch size m/((kd)

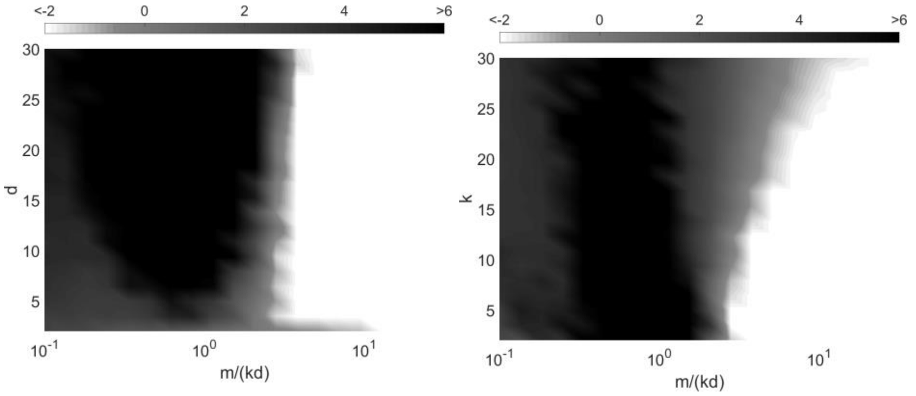
How big a sketch ?

k-means



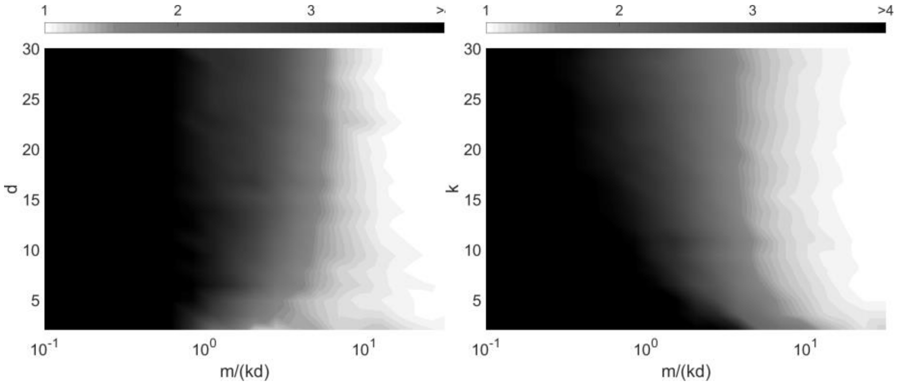
Relative sketch size m/(kd)

GMM



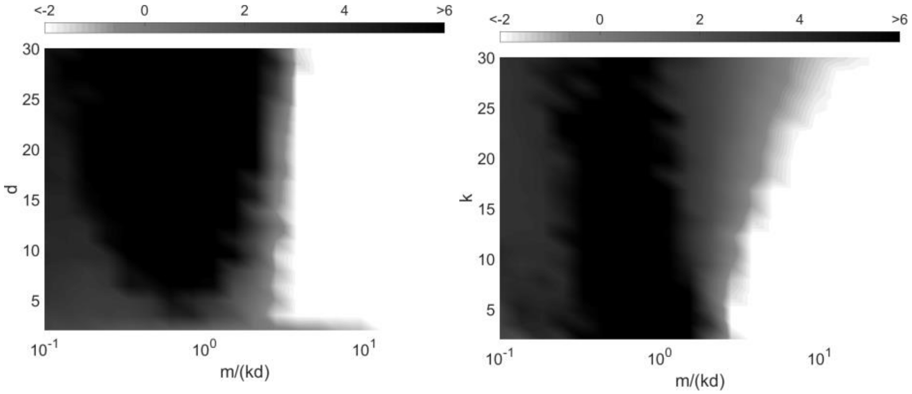
How big a sketch ?

k-means

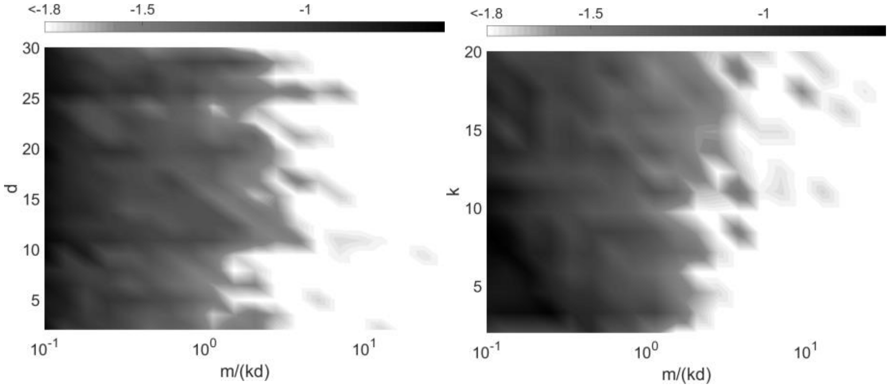


Relative sketch size m/(kd)

GMM

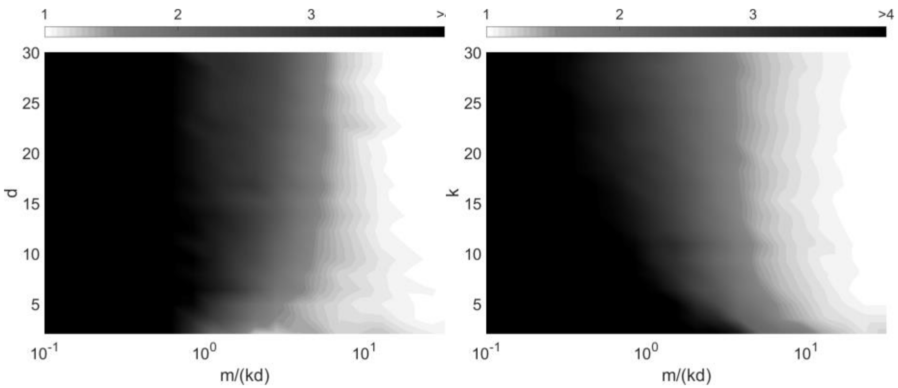


Stable distributions

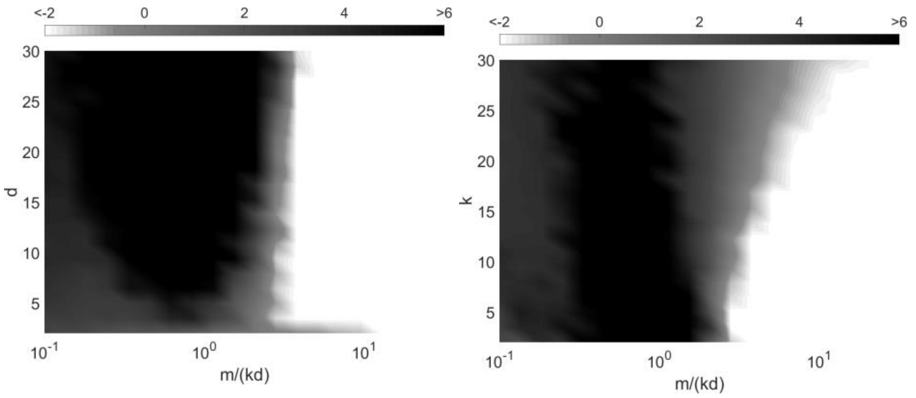


How big a sketch ?

k-means

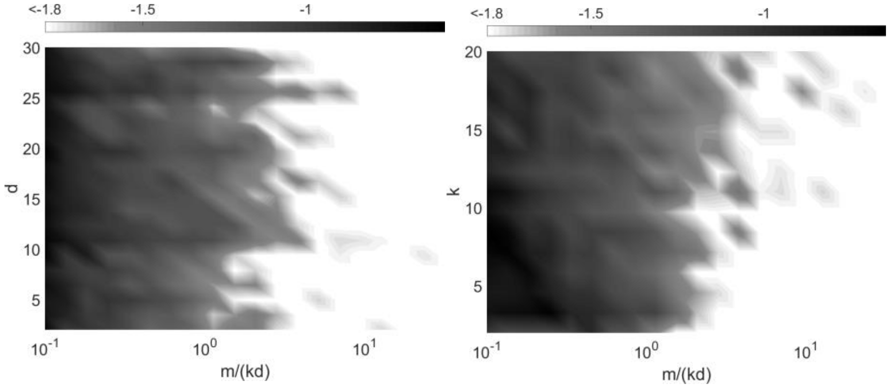


GMM



Relative sketch size m/(kd)

Stable distributions



Sufficient sketch size?

$$m \approx \mathcal{O}(kd)$$

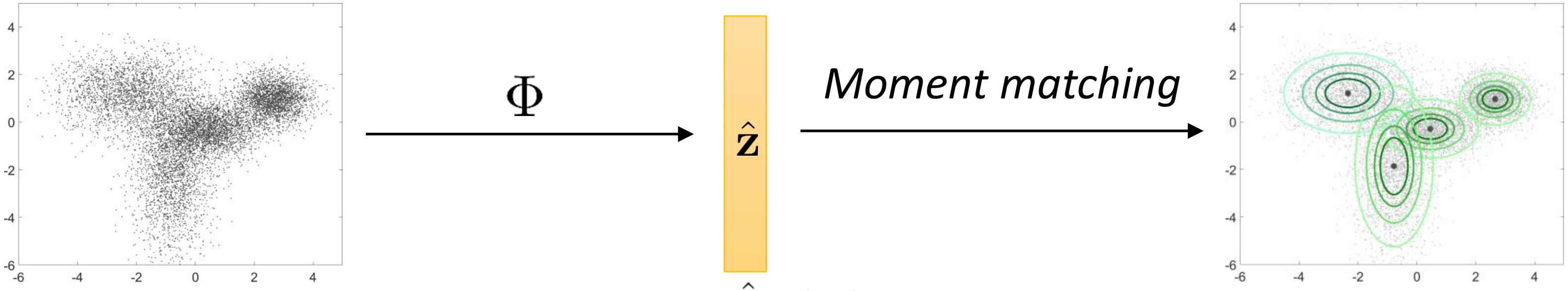
Outline

① Illustration: Sketched Mixture Model Estimation

② A Compressive Sensing analysis

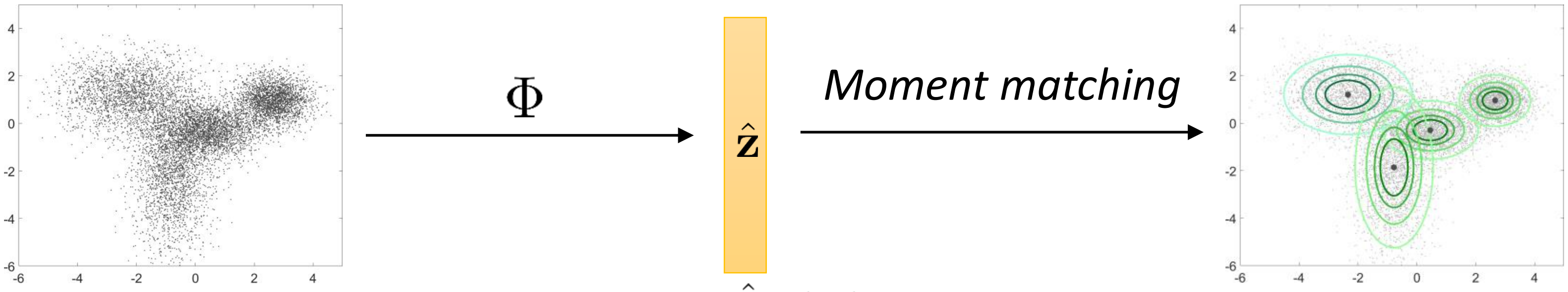
③ Conclusion, outlooks

Linear inverse problem



$$\hat{\mathbf{z}} = \hat{\mathbb{E}}\Phi(X)$$

Linear inverse problem

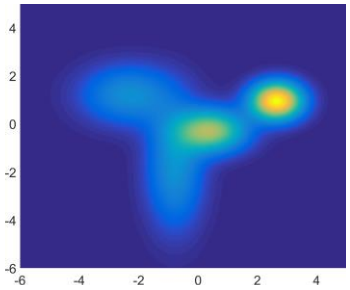


Φ

$\hat{\mathbf{z}}$

Moment matching

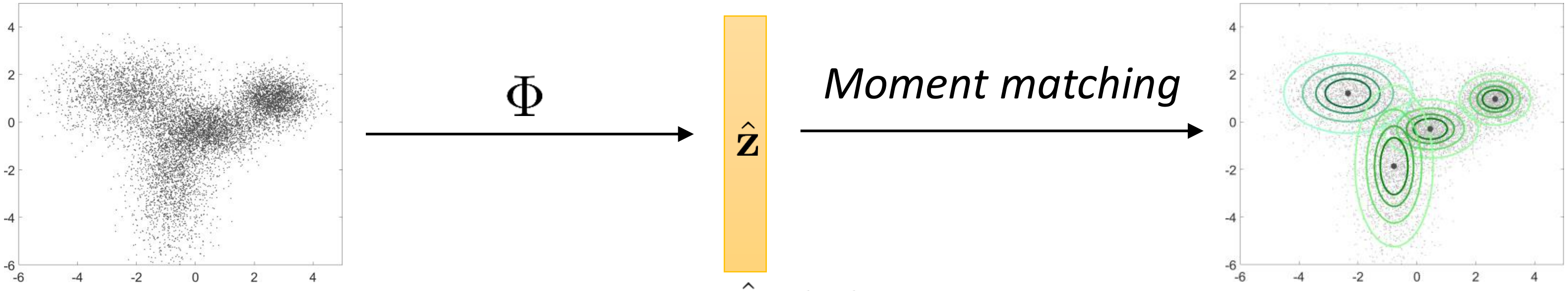
$$\hat{\mathbf{z}} = \hat{\mathbb{E}}\Phi(X)$$



$\pi^\star$

True distribution: $\boxed{x_1, \ldots, x_n \overset{i.i.d.}{\sim} \pi^\star}$

Linear inverse problem

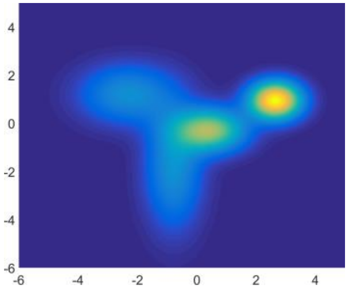


Φ

$\hat{\mathbf{z}}$

Moment matching

$$\hat{\mathbf{z}} = \hat{\mathbb{E}}\Phi(X)$$

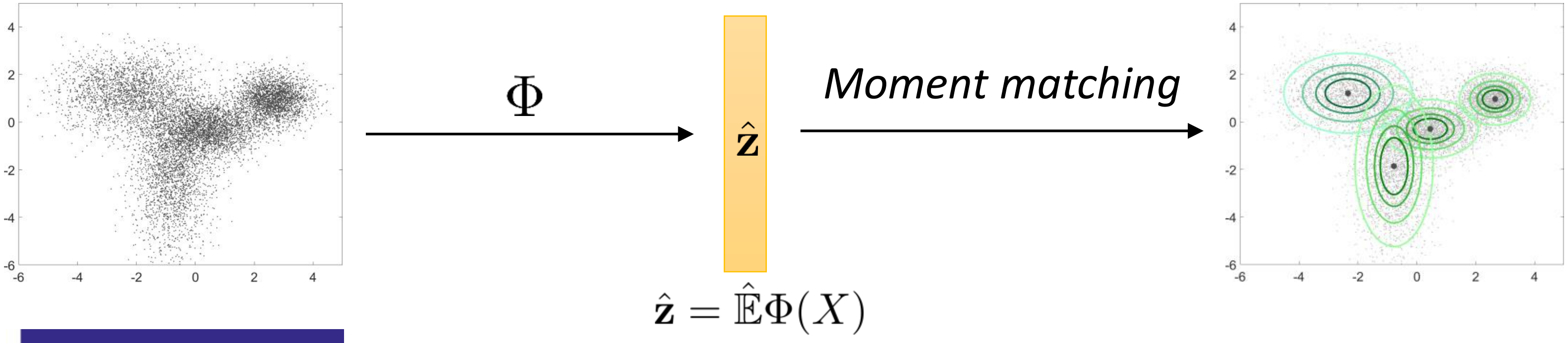


$\pi^\star$

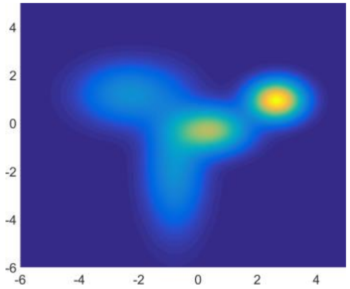
True distribution: $\boxed{x_1, \ldots, x_n \overset{i.i.d.}{\sim} \pi^\star}$

Reformulation of the sketching

Linear inverse problem



$$\hat{\mathbf{z}} = \hat{\mathbb{E}}\Phi(X)$$



$\pi^\star$

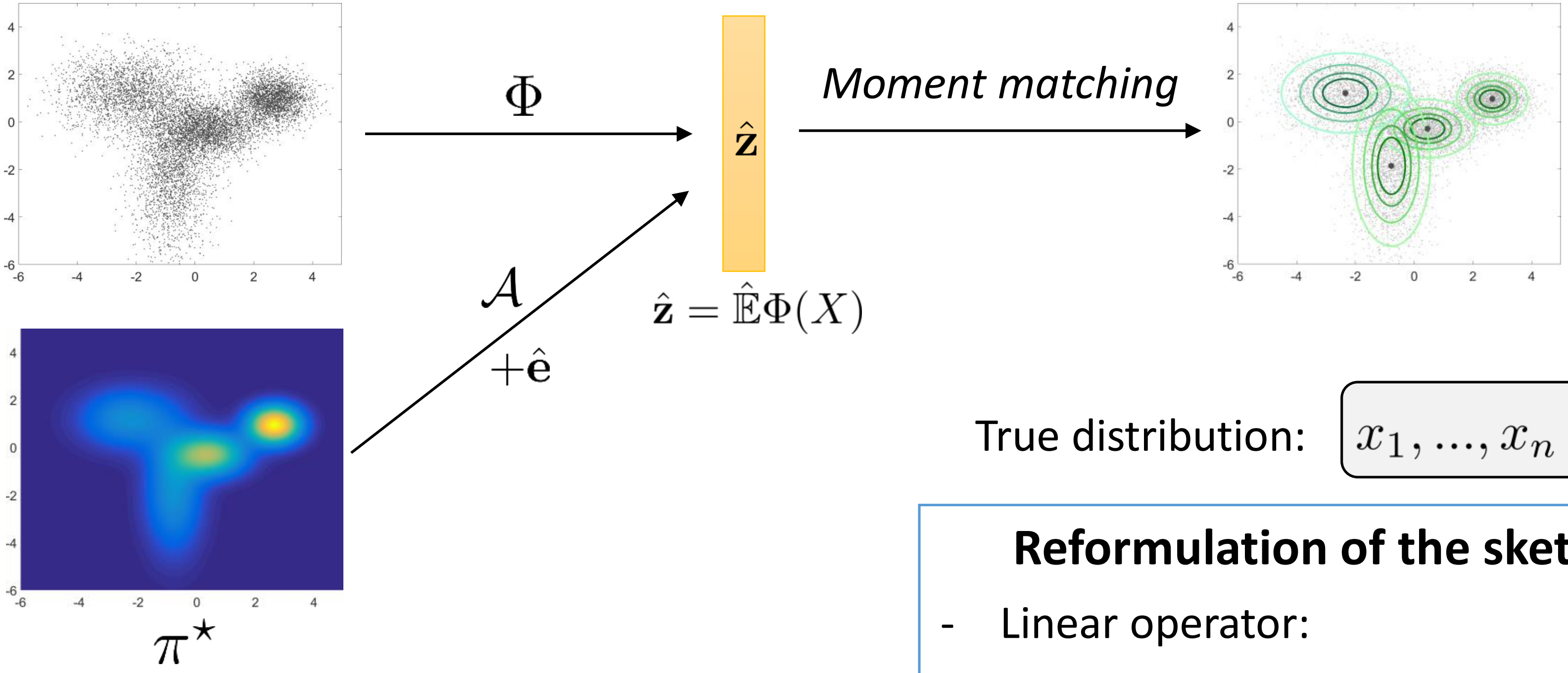
True distribution: $\boxed{x_1, ..., x_n \overset{i.i.d.}{\sim} \pi^\star}$

Reformulation of the sketching

- Linear operator:

$$\boxed{\mathcal{A}\pi = \mathbb{E}_{X\sim\pi}\Phi(X)}$$

Linear inverse problem



True distribution: $\boxed{x_1, \ldots, x_n \overset{i.i.d.}{\sim} \pi^\star}$

- Estimation problem = **linear inverse problem on measures**

Reformulation of the sketching

- Linear operator:

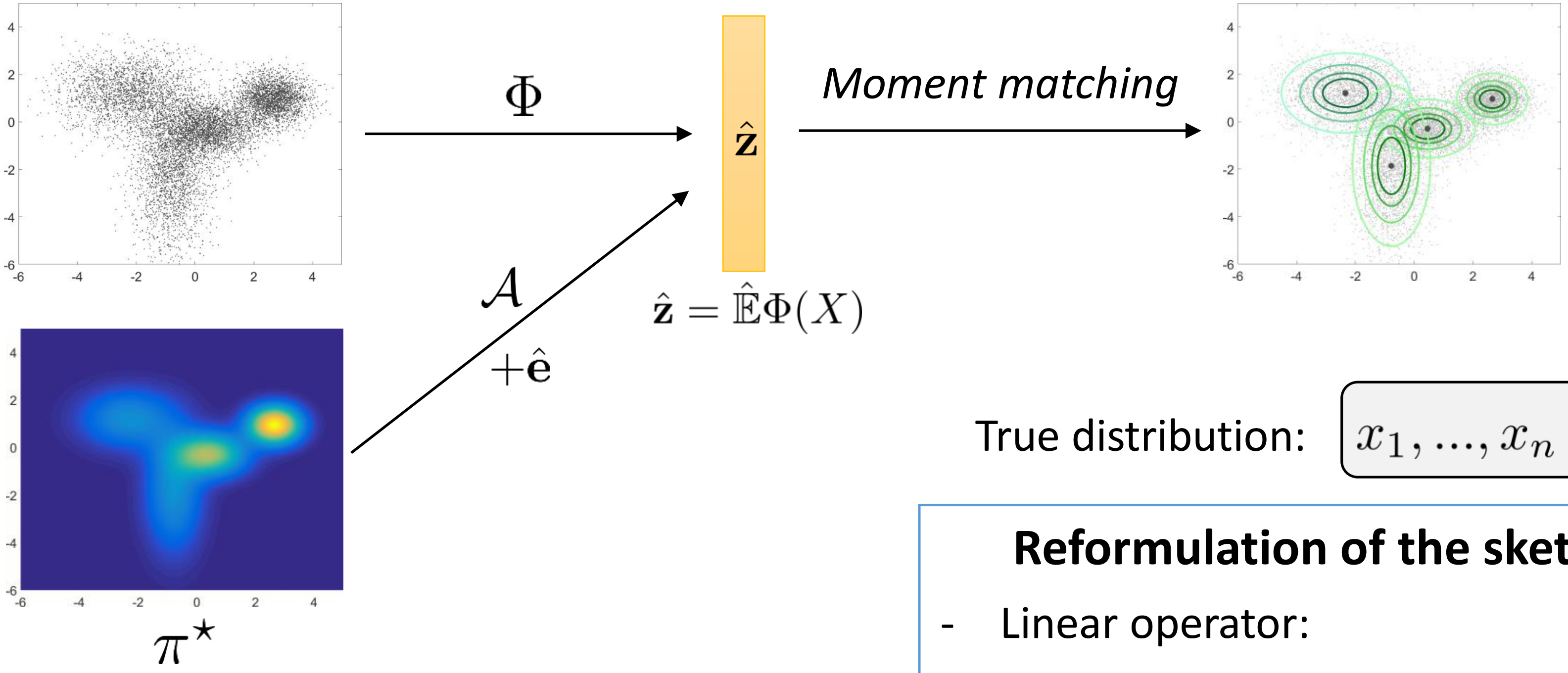
$$\boxed{\mathcal{A}\pi = \mathbb{E}_{X\sim\pi}\Phi(X)}$$

- « Noisy » linear measurement:

$$\boxed{\hat{\mathbf{z}} = \mathcal{A}\pi^\star + \hat{\mathbf{e}}}$$

Noise $\hat{\mathbf{e}} = \hat{\mathbb{E}}\Phi(X) - \mathbb{E}_{\pi^\star}\Phi(X)$ ***small***

Linear inverse problem



True distribution: $x_1, \ldots, x_n \overset{i.i.d.}{\sim} \pi^\star$

Reformulation of the sketching

- Linear operator:

$$\mathcal{A}\pi = \mathbb{E}_{X\sim\pi}\Phi(X)$$

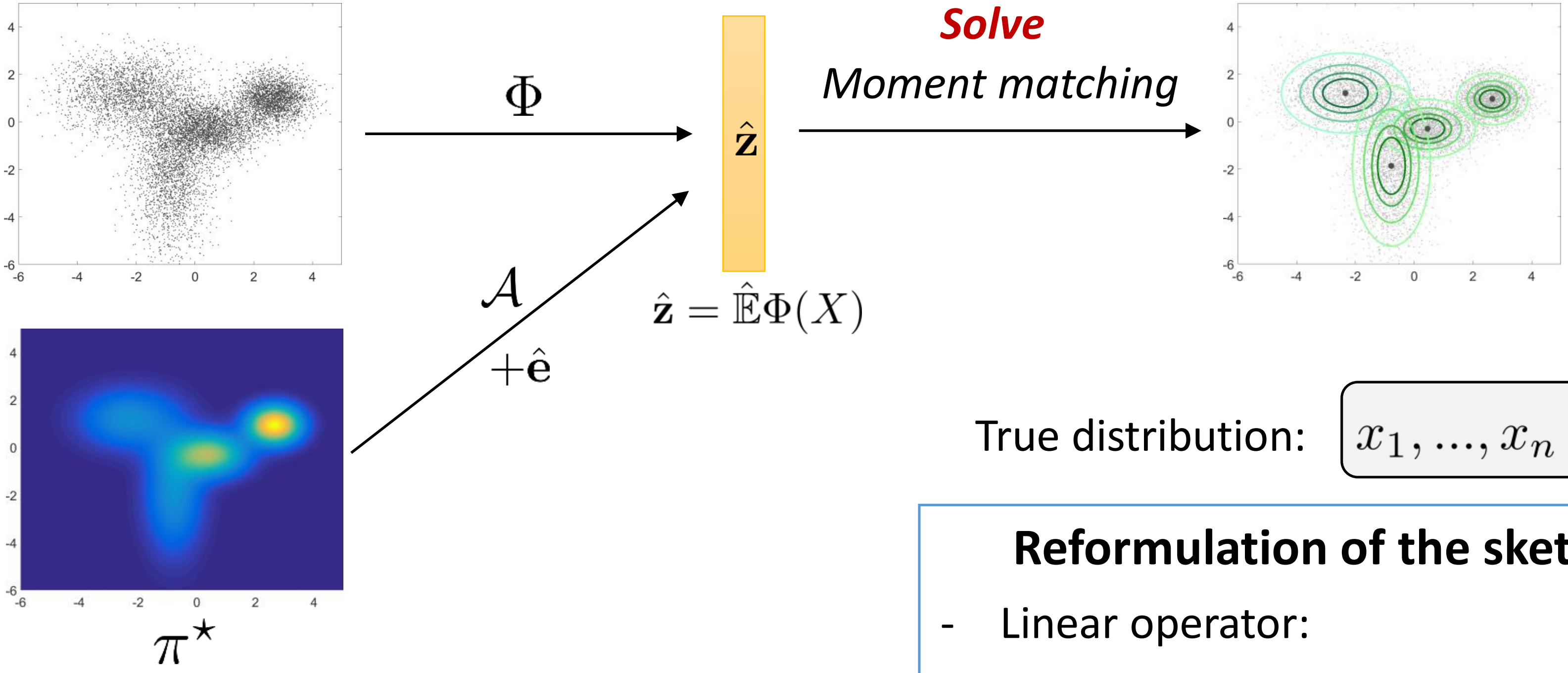
- « Noisy » linear measurement:

$$\boxed{\hat{\mathbf{z}} = \mathcal{A}\pi^\star + \hat{\mathbf{e}}}$$

Noise $\hat{\mathbf{e}} = \hat{\mathbb{E}}\Phi(X) - \mathbb{E}_{\pi^\star}\Phi(X)$ ***small***

- Estimation problem = **linear inverse problem on measures**
- **Extremely ill-posed !**

Linear inverse problem



True distribution: $\boxed{x_1, \ldots, x_n \overset{i.i.d.}{\sim} \pi^\star}$

- Estimation problem = **linear inverse problem on measures**
- **Extremely ill-posed !**
- ***Feasibility?*** *(information-preservation)*

Reformulation of the sketching

- Linear operator:

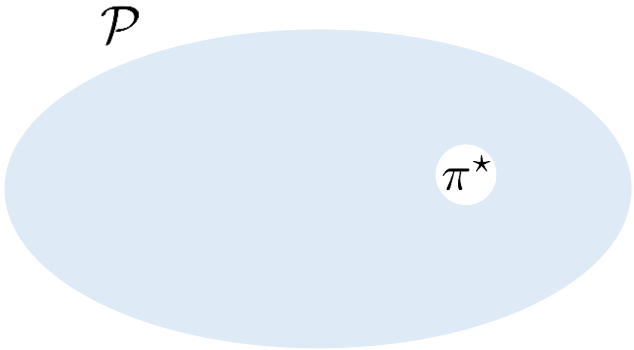
$$\boxed{\mathcal{A}\pi = \mathbb{E}_{X\sim\pi}\Phi(X)}$$

- « Noisy » linear measurement:

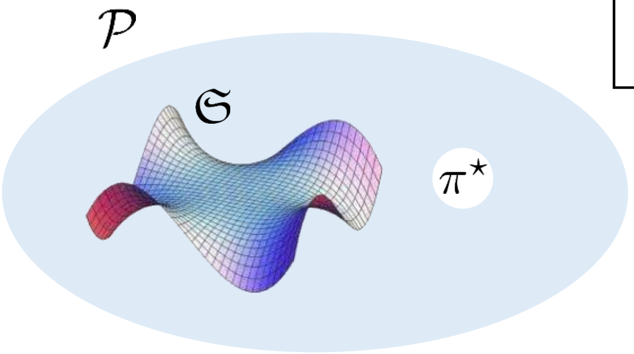
$$\boxed{\hat{\mathbf{z}} = \mathcal{A}\pi^\star + \hat{\mathbf{e}}}$$

Noise $\hat{\mathbf{e}} = \hat{\mathbb{E}}\Phi(X) - \mathbb{E}_{\pi^\star}\Phi(X)$ ***small***

Information preservation guarantees

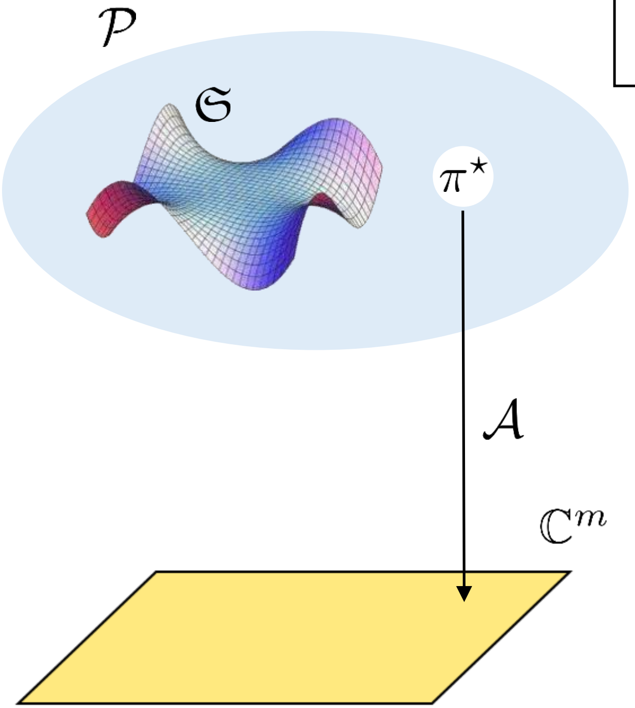


Information preservation guarantees



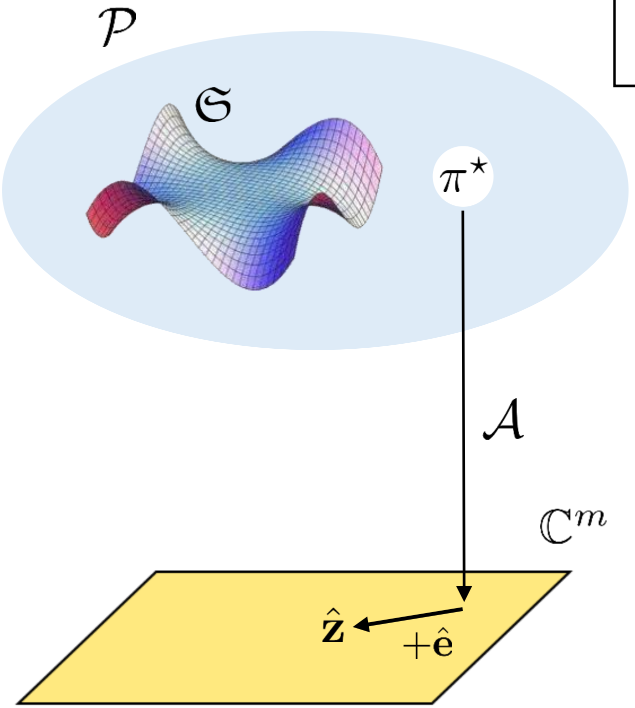
$\mathfrak{S}$ *: Model set of « simple » distributions (eg. GMMs)*

Information preservation guarantees



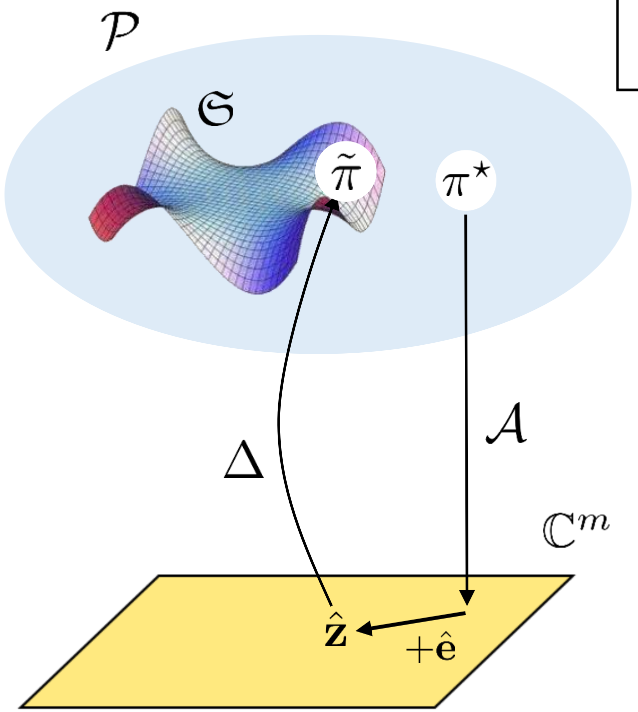
$\mathfrak{S}$ *: Model set of « simple » distributions (eg. GMMs)*

Information preservation guarantees



$\mathfrak{S}$ *: Model set of « simple » distributions (eg. GMMs)*

Information preservation guarantees



$\mathfrak{S}$ *: Model set of « simple » distributions (eg. GMMs)*

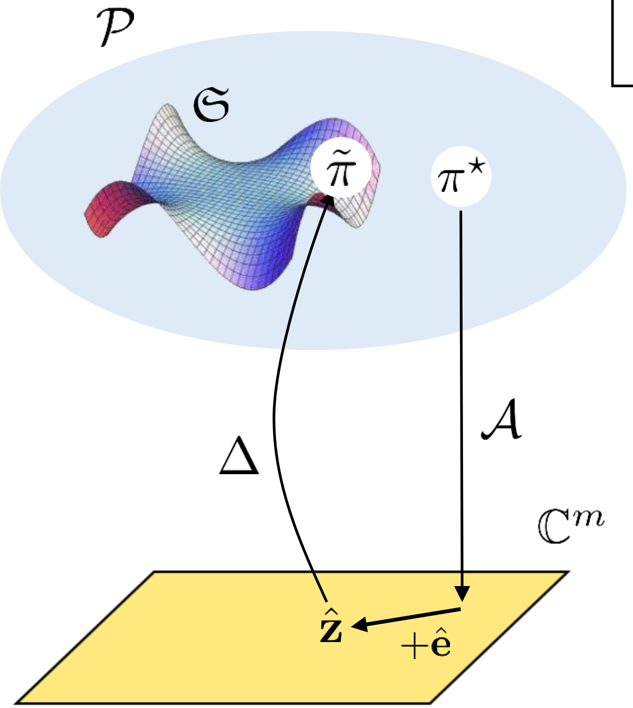
Goal

Prove the existence of a *decoder* Δ robust to noise and stable to modeling error.

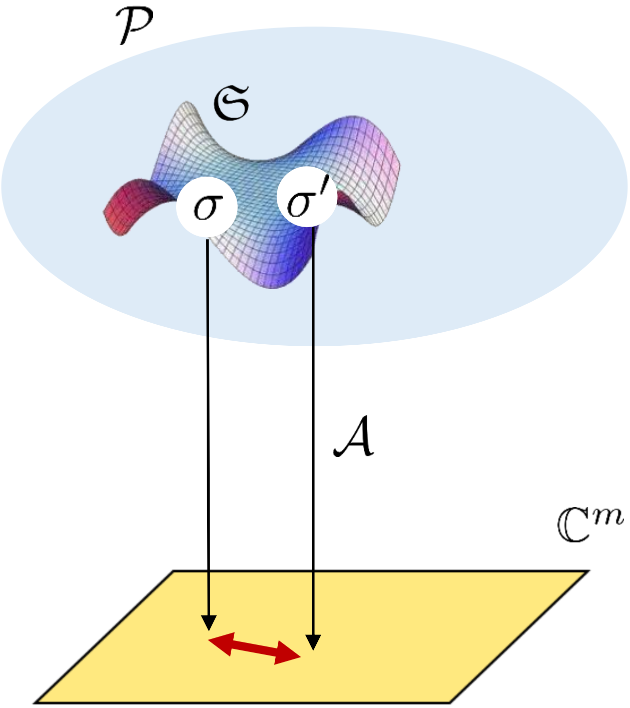
« Instance-optimal » decoder

Information preservation guarantees

$\mathfrak{S}$ *: Model set of « simple » distributions (eg. GMMs)*



$\Longrightarrow$



Goal

Prove the existence of a *decoder* Δ robust to noise and stable to modeling error.

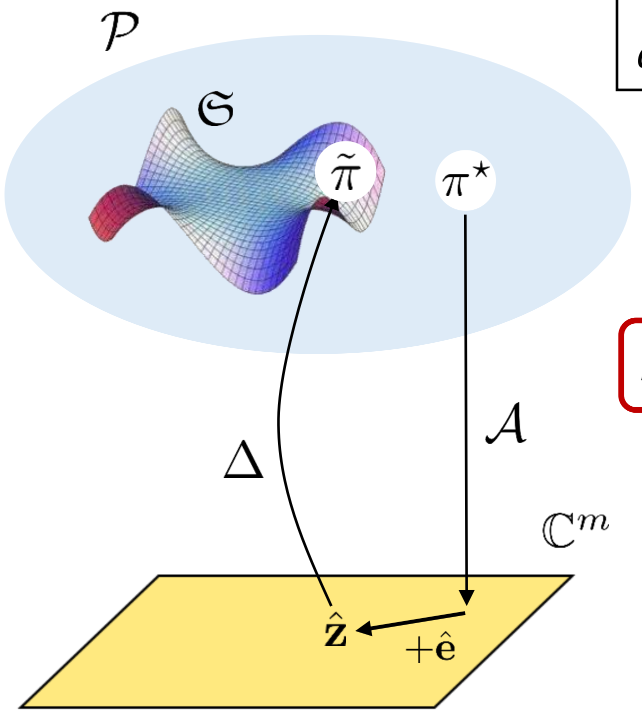
« Instance-optimal » decoder

Lower Restricted Isometry Property

$$\|\sigma - \sigma'\| \lesssim \|\mathcal{A}\sigma - \mathcal{A}\sigma'\|_2$$

Information preservation guarantees

$\mathfrak{S}$: Model set of « simple » distributions (eg. GMMs)

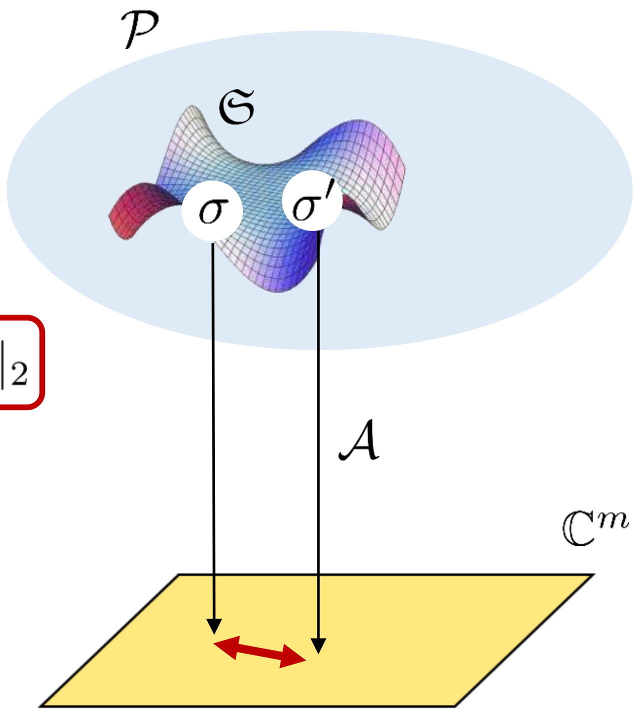


$$\Longrightarrow$$

$$\Longleftarrow$$

$$\boxed{\Delta(\hat{\mathbf{z}}) \in \arg\min_{\sigma \in \mathfrak{S}} \|\hat{\mathbf{z}} - \mathcal{A}\sigma\|_2}$$

Moment matching



Goal

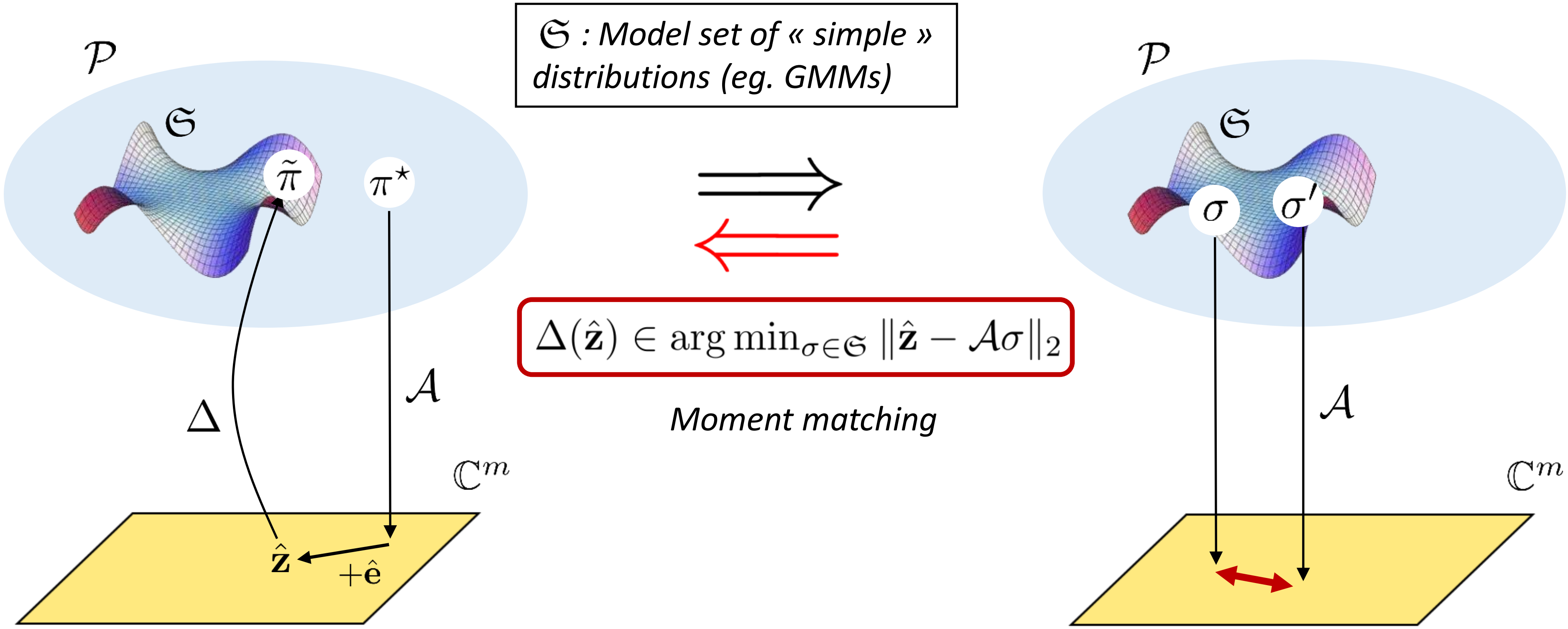
Prove the existence of a *decoder* Δ robust to noise and stable to modeling error.

« Instance-optimal » decoder

Lower Restricted Isometry Property

$$\|\sigma - \sigma'\| \lesssim \|\mathcal{A}\sigma - \mathcal{A}\sigma'\|_2$$

Information preservation guarantees



$\mathfrak{S}$: Model set of « simple » distributions (eg. GMMs)

$$\Delta(\hat{\mathbf{z}}) \in \arg\min_{\sigma\in\mathfrak{S}} \|\hat{\mathbf{z}} - \mathcal{A}\sigma\|_2$$

Moment matching

Goal

Prove the existence of a *decoder* Δ robust to noise and stable to modeling error.

« Instance-optimal » decoder

Lower Restricted Isometry Property

$$\|\sigma - \sigma'\| \lesssim \|\mathcal{A}\sigma - \mathcal{A}\sigma'\|_2$$

New goal: find/construct models $\mathfrak{S}$ and operators $\mathcal{A}$ that satisfy the LRIP (w.h.p.)

Proving the LRIP

Goal: LRIP $\text{w.h.p. on } \mathcal{A}, \forall \sigma, \sigma' \in \mathfrak{S}, \|\sigma - \sigma'\| \lesssim \|\mathcal{A}\sigma - \mathcal{A}\sigma'\|_2.$

Proving the LRIP

Goal: LRIP $\text{w.h.p. on } \mathcal{A}, \forall \sigma, \sigma' \in \mathfrak{S}, \|\sigma - \sigma'\| \lesssim \|\mathcal{A}\sigma - \mathcal{A}\sigma'\|_2.$

① **Pointwise LRIP**

Construction of $\mathcal{A}$:

Kernel mean *[Gretton 2006, Borgwardt 2006]*

Random features *[Rahimi 2007]*

Proving the LRIP

Goal: LRIP $\text{w.h.p. on } \mathcal{A}, \forall \sigma, \sigma' \in \mathfrak{S}, \ \|\sigma - \sigma'\| \lesssim \|\mathcal{A}\sigma - \mathcal{A}\sigma'\|_2.$

① **Pointwise LRIP**

Construction of $\mathcal{A}$:

Kernel mean *[Gretton 2006, Borgwardt 2006]*

Random features *[Rahimi 2007]*

$\forall \sigma, \sigma', \ \text{w.h.p. on } \mathcal{A}, \ \text{LRIP.}$

Proving the LRIP

Goal: LRIP $\text{w.h.p. on } \mathcal{A}, \forall \sigma, \sigma' \in \mathfrak{S}, \|\sigma - \sigma'\| \lesssim \|\mathcal{A}\sigma - \mathcal{A}\sigma'\|_2.$

① **Pointwise LRIP**

Construction of $\mathcal{A}$:

Kernel mean *[Gretton 2006, Borgwardt 2006]*

Random features *[Rahimi 2007]*

$\forall \sigma, \sigma', \text{ w.h.p. on } \mathcal{A}, \text{ LRIP.}$

② **Extension to LRIP**

Covering numbers (compacity) of the **normalized secant set** $\mathcal{S}(\mathfrak{S})$

Proving the LRIP

Goal: LRIP $\text{w.h.p. on } \mathcal{A}, \forall \sigma, \sigma' \in \mathfrak{S}, \|\sigma - \sigma'\| \lesssim \|\mathcal{A}\sigma - \mathcal{A}\sigma'\|_2.$

① **Pointwise LRIP**

Construction of $\mathcal{A}$:

Kernel mean *[Gretton 2006, Borgwardt 2006]*

Random features *[Rahimi 2007]*

$\forall \sigma, \sigma', \text{ w.h.p. on } \mathcal{A}, \text{ LRIP.}$

② **Extension to LRIP**

Covering numbers (compacity) of the **normalized secant set** $\mathcal{S}(\mathfrak{S})$

Subset of a unit ball (infinite dimension) that only depends on $\mathfrak{S}$

Proving the LRIP

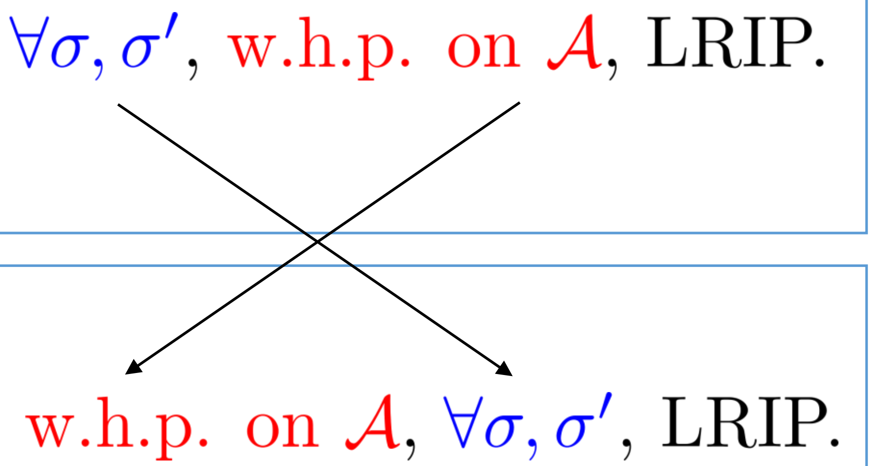
Goal: LRIP $\text{w.h.p. on } \mathcal{A},\ \forall \sigma, \sigma' \in \mathfrak{S},\ \|\sigma - \sigma'\| \lesssim \|\mathcal{A}\sigma - \mathcal{A}\sigma'\|_2.$

① Pointwise LRIP

Construction of $\mathcal{A}$:

Kernel mean *[Gretton 2006, Borgwardt 2006]*

Random features *[Rahimi 2007]*



$\forall \sigma, \sigma',\ \text{w.h.p. on } \mathcal{A},\ \text{LRIP}.$

② Extension to LRIP

Covering numbers (compacity) of the **normalized secant set** $\mathcal{S}(\mathfrak{S})$

$\text{w.h.p. on } \mathcal{A},\ \forall \sigma, \sigma',\ \text{LRIP}.$

Subset of a unit ball (infinite dimension) that only depends on $\mathfrak{S}$

Main result [Keriven 2016]

Main hypothesis

The *normalized secant set* $\mathcal{S}(\mathfrak{S})$ has finite covering numbers.

Main result [Keriven 2016]

Main hypothesis

The *normalized secant set* $\mathcal{S}(\mathfrak{S})$ has finite covering numbers.

Result

For $\boxed{m \geq C \ \times \ \log(\text{cov. num.})}$,

Main result [Keriven 2016]

Main hypothesis

The *normalized secant set* $\mathcal{S}(\mathfrak{S})$ has finite covering numbers.

Result

For $$\boxed{m \geq \underbrace{C}_{\text{Pointwise concentration}} \times \log(\underbrace{\text{cov. num.}}_{\text{Dimensionality of the model}})}\,,$$

Main result [Keriven 2016]

Main hypothesis

The *normalized secant set* $\mathcal{S}(\mathfrak{S})$ has finite covering numbers.

Result

For $\boxed{m \geq C \times \log(\text{cov. num.})}$,

Pointwise concentration

Dimensionality of the model

W.h.p.

$$\|\pi^\star - \Delta(\hat{\mathbf{z}})\| \leq d(\pi^\star, \mathfrak{S}) + \mathcal{O}(1/\sqrt{n})$$

Main result [Keriven 2016]

Main hypothesis

The *normalized secant set* $\mathcal{S}(\mathfrak{S})$ has finite covering numbers.

Result

For $\boxed{m \geq C \times \log(\text{cov. num.})}$,

Pointwise concentration

Dimensionality of the model

W.h.p.

Modeling error

Empirical noise

$$\|\pi^\star - \Delta(\hat{\mathbf{z}})\| \leq d(\pi^\star, \mathfrak{S}) + \mathcal{O}(1/\sqrt{n})$$

Main result [Keriven 2016]

Main hypothesis

The *normalized secant set* $\mathcal{S}(\mathfrak{S})$ has finite covering numbers.

Result

For $\boxed{m \geq C \times \log(\text{cov. num.})}$,

Pointwise concentration

Dimensionality of the model

W.h.p.

Modeling error

Empirical noise

$$\|\pi^\star - \Delta(\hat{\mathbf{z}})\| \leq d(\pi^\star, \mathfrak{S}) + \mathcal{O}(1/\sqrt{n})$$

- **Classic Compressive Sensing**: finite dimension: **Known**
- **Here**: infinite dimension: **Technical**

Compressive statistical learning

[Gribonval, Blanchard, Keriven, Traonmilin 2017]

k-means/k-medians with mixtures of Diracs

Compressive statistical learning

[Gribonval, Blanchard, Keriven, Traonmilin 2017]

k-means/k-medians with mixtures of Diracs

Hypotheses

- ε - separated centroids
- M- bounded domain for centroids

Compressive statistical learning

[Gribonval, Blanchard, Keriven, Traonmilin 2017]

k-means/k-medians with mixtures of Diracs

*(no assumption on the **data**)*

Hypotheses

- ε - separated centroids
- M- bounded domain for centroids

Compressive statistical learning

[Gribonval, Blanchard, Keriven, Traonmilin 2017]

k-means/k-medians with mixtures of Diracs

*(no assumption on the **data**)*

Hypotheses

- ε - separated centroids
- M- bounded domain for centroids

Sketch

- *Weighted* Fourier sampling *(for technical reasons)*

Compressive statistical learning

[Gribonval, Blanchard, Keriven, Traonmilin 2017]

k-means/k-medians with mixtures of Diracs

(no assumption on the ***data****)*

Hypotheses

- ε - separated centroids
- M- bounded domain for centroids

Sketch

- *Weighted* Fourier sampling *(for technical reasons)*

Result

- W.r.t. **k-means usual cost (SSE)**

Compressive statistical learning

[Gribonval, Blanchard, Keriven, Traonmilin 2017]

k-means/k-medians with mixtures of Diracs

(no assumption on the ***data****)*

Hypotheses

- ε - separated centroids
- M- bounded domain for centroids

Sketch

- *Weighted* Fourier sampling *(for technical reasons)*

Result

- W.r.t. **k-means usual cost (SSE)**

Sketch size

$$m \geq \mathcal{O}\left(k^2 d \cdot \mathtt{polylog}(k,d) \log(M/\varepsilon)\right)$$

Compressive statistical learning

[Gribonval, Blanchard, Keriven, Traonmilin 2017]

k-means/k-medians with mixtures of Diracs

*(no assumption on the **data**)*

Hypotheses

- ε - separated centroids
- M- bounded domain for centroids

Sketch

- *Weighted* Fourier sampling *(for technical reasons)*

Result

- W.r.t. **k-means usual cost (SSE)**

Sketch size

$$m \geq \mathcal{O}\left(k^2 d \cdot \mathtt{polylog}(k,d) \log(M/\varepsilon)\right)$$

GMM with known covariance

Compressive statistical learning

[Gribonval, Blanchard, Keriven, Traonmilin 2017]

k-means/k-medians with mixtures of Diracs

(no assumption on the ***data****)*

Hypotheses

- ε - separated centroids
- M- bounded domain for centroids

Sketch

- *Weighted* Fourier sampling *(for technical reasons)*

Result

- W.r.t. **k-means usual cost (SSE)**

Sketch size

$$m \geq \mathcal{O}\left(k^2 d \cdot \texttt{polylog}(k,d) \log(M/\varepsilon)\right)$$

GMM with known covariance

Hypotheses

- Sufficiently separated means
- Bounded domain for means

Compressive statistical learning

[Gribonval, Blanchard, Keriven, Traonmilin 2017]

k-means/k-medians with mixtures of Diracs

(no assumption on the ***data****)*

Hypotheses

- ε - separated centroids
- M- bounded domain for centroids

Sketch

- *Weighted* Fourier sampling *(for technical reasons)*

Result

- W.r.t. **k-means usual cost (SSE)**

Sketch size

$$m \geq \mathcal{O}\left(k^2 d \cdot \texttt{polylog}(k,d) \log(M/\varepsilon)\right)$$

GMM with known covariance

Hypotheses

- Sufficiently separated means
- Bounded domain for means

Sketch

- Fourier sampling

Compressive statistical learning

[Gribonval, Blanchard, Keriven, Traonmilin 2017]

k-means/k-medians with mixtures of Diracs

(no assumption on the ***data****)*

Hypotheses

- ε - separated centroids
- M- bounded domain for centroids

Sketch

- *Weighted* Fourier sampling *(for technical reasons)*

Result

- W.r.t. **k-means usual cost (SSE)**

Sketch size

$$m \geq \mathcal{O}\left(k^2 d \cdot \texttt{polylog}(k,d) \log(M/\varepsilon)\right)$$

GMM with known covariance

Hypotheses

- Sufficiently separated means
- Bounded domain for means

Sketch

- Fourier sampling

Result

- With respect to **log-likelihood**

Compressive statistical learning

[Gribonval, Blanchard, Keriven, Traonmilin 2017]

k-means/k-medians with mixtures of Diracs

(no assumption on the ***data****)*

Hypotheses

- ε - separated centroids
- M- bounded domain for centroids

Sketch

- *Weighted* Fourier sampling *(for technical reasons)*

Result

- W.r.t. **k-means usual cost (SSE)**

Sketch size

$$m \geq \mathcal{O}\left(k^2 d \cdot \texttt{polylog}(k,d) \log(M/\varepsilon)\right)$$

GMM with known covariance

Hypotheses

- Sufficiently separated means
- Bounded domain for means

Sketch

- Fourier sampling

Result

- With respect to **log-likelihood**

Sketch size

$$m \geq \mathcal{O}(k^2 d \cdot \texttt{polylog}(k,d) \varphi(\text{sep.}))$$

Compressive statistical learning

[Gribonval, Blanchard, Keriven, Traonmilin 2017]

k-means/k-medians with mixtures of Diracs

*(no assumption on the **data**)*

Hypotheses

- ε - separated centroids
- M- bounded domain for centroids

Sketch

- *Weighted* Fourier sampling *(for technical reasons)*

Result

- W.r.t. **k-means usual cost (SSE)**

Sketch size

$$m \geq \mathcal{O}\left(k^2 d \cdot \texttt{polylog}(k,d) \log(M/\varepsilon)\right)$$

GMM with known covariance

Hypotheses

- Sufficiently separated means
- Bounded domain for means

Sketch

- Fourier sampling

Result

- With respect to **log-likelihood**

Sketch size

$$m \geq \mathcal{O}(k^2 d \cdot \texttt{polylog}(k,d) \varphi(\text{sep.}))$$

$$\varphi(\sqrt{d \log k}) = 1 \qquad \varphi(\sqrt{\log k}) = e^d$$

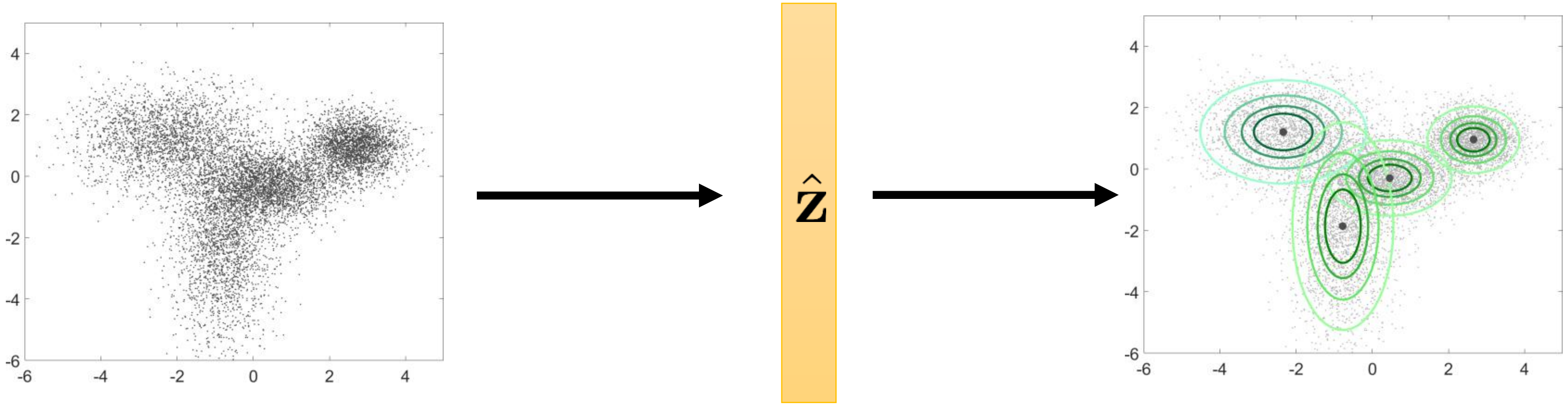
Outline

① Illustration: Sketched Mixture Model Estimation

② A Compressive Sensing analysis

③ Conclusion, outlooks

Sketch learning



- Sketching method for **large-scale density estimation**
  - Well-adapted to **distributed** or **streaming** context
  - Focus on **mixture model estimation**

Summary of contributions

- Practical illustration: **flexible heuristic algorithm for sketched mixture model estimation**
  - GMM with diagonal covariance
  - k-means (mixture of Diracs)
  - *Mixture of multivariate elliptic stable distributions*

Summary of contributions

- Practical illustration: **flexible heuristic algorithm for sketched mixture model estimation**
    - GMM with diagonal covariance
    - k-means (mixture of Diracs)
    - *Mixture of multivariate elliptic stable distributions*

- Information-preservation guarantees
    - Infinite dimensional **Compressive Sensing** (Restricted isometry property)
    - **Kernel methods** on distributions (Kernel mean, Random features)
- Generic assumptions of *low-dimensionality* of the model set

Summary of contributions

- Practical illustration: **flexible heuristic algorithm for sketched mixture model estimation**
  - GMM with diagonal covariance
  - k-means (mixture of Diracs)
  - *Mixture of multivariate elliptic stable distributions*

- Information-preservation guarantees
  - Infinite dimensional **Compressive Sensing** (Restricted isometry property)
  - **Kernel methods** on distributions (Kernel mean, Random features)
- Generic assumptions of *low-dimensionality* of the model set

- Outlooks
  - Convex relaxation (super-resolution)
  - Reduction of the dimension d
  - Hierarchical sketch (neural networks…)

Thank you !

- Keriven, Bourrier, Gribonval, Pérez. **Sketching for Large-Scale Learning of Mixture Models** *Information & Inference: a Journal of the IMA,* 2017. <arXiv:1606.02838>
- Keriven, Tremblay, Traonmilin, Gribonval. **Compressive k-means** *ICASSP,* 2017.
- Gribonval, Blanchard, Keriven, Traonmilin. **Compressive Statistical Learning with Random Feature Moments**. *Preprint* 2017. <arXiv:1706.07180>
- Keriven. **Sketching for Large-Scale Learning of Mixture Models**. *PhD Thesis.* <tel-01620815>
- **Code**: sketchml.gforge.inria.fr