Sketching for Large-Scale Learning of Mixture Models

Nicolas Keriven

Université Rennes 1 Ecole doctorale MATISSE

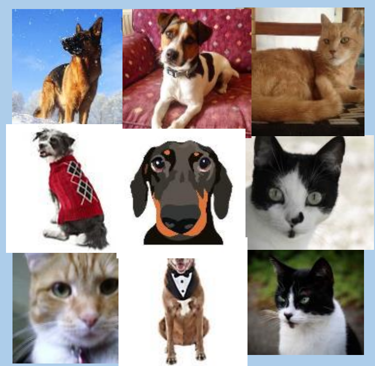
IRISA (CNRS/UMR 6074), Team PANAMA

Advisor: Rémi Gribonval

Thesis defense – 2017 October 12th

Context: machine learning

Database



Learning

Automatic task

- Clustering

- Classification

= cat

- etc…

Context: machine learning

Large **database**

Learning

Automatic task

- Clustering

- Classification

= cat

- etc…

Context: machine learning

Large **database**

Large elements
Billions of elements

Learning →

Automatic task

- Clustering

- Classification

= cat

- etc…

Context: machine learning

Large **database**

Large elements

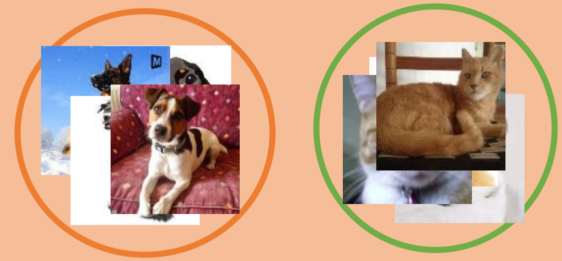
Billions of elements

Learning

Slow, costly

Automatic task

- Clustering



- Classification

= cat

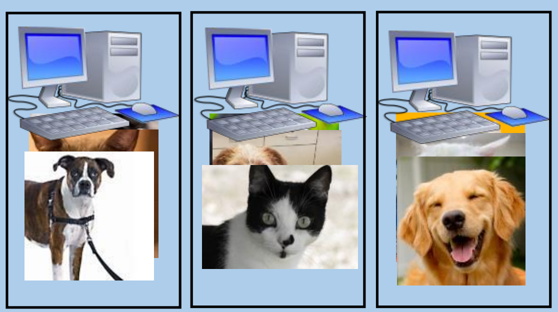
- etc…

Context: machine learning

Large **database**

Large elements
Billions of elements

Distributed **database**

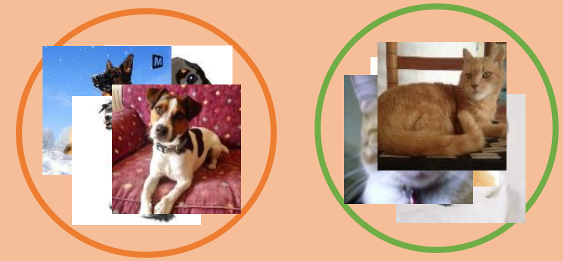


Learning

Slow, costly

Automatic task

- Clustering



- Classification

 = cat

- etc…

Context: machine learning

Large **database**

Large elements

Billions of elements

Learning

Slow, costly

Automatic task

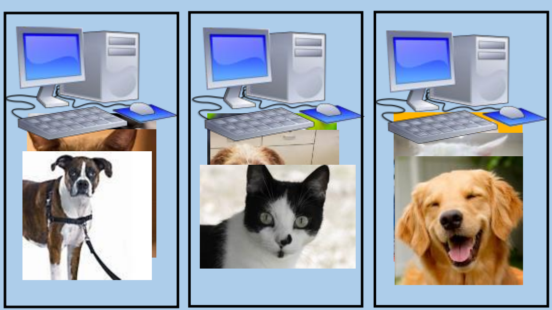
- Clustering

- Classification

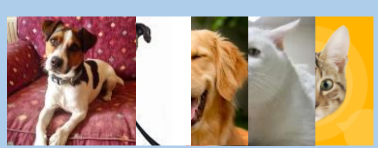
= cat

- etc…

Distributed **database**



Data *Stream*

…  …

Context: machine learning

Large **database**

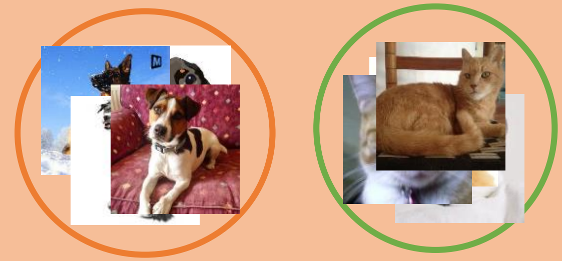
Large elements
Billions of elements

Learning

Slow, costly

Automatic task

- Clustering



- Classification

= cat

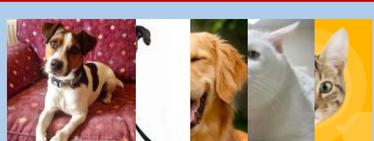
- etc...

Idea!

Small intermediate representation

Distributed **database**

Data ***Stream***

...  ...

Context: machine learning

Large **database**

Large elements
Billions of elements

Learning

Slow, costly

Automatic task

- Clustering

- Classification

= cat

- etc…

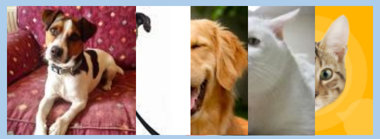
Distributed **database**

1: Compression

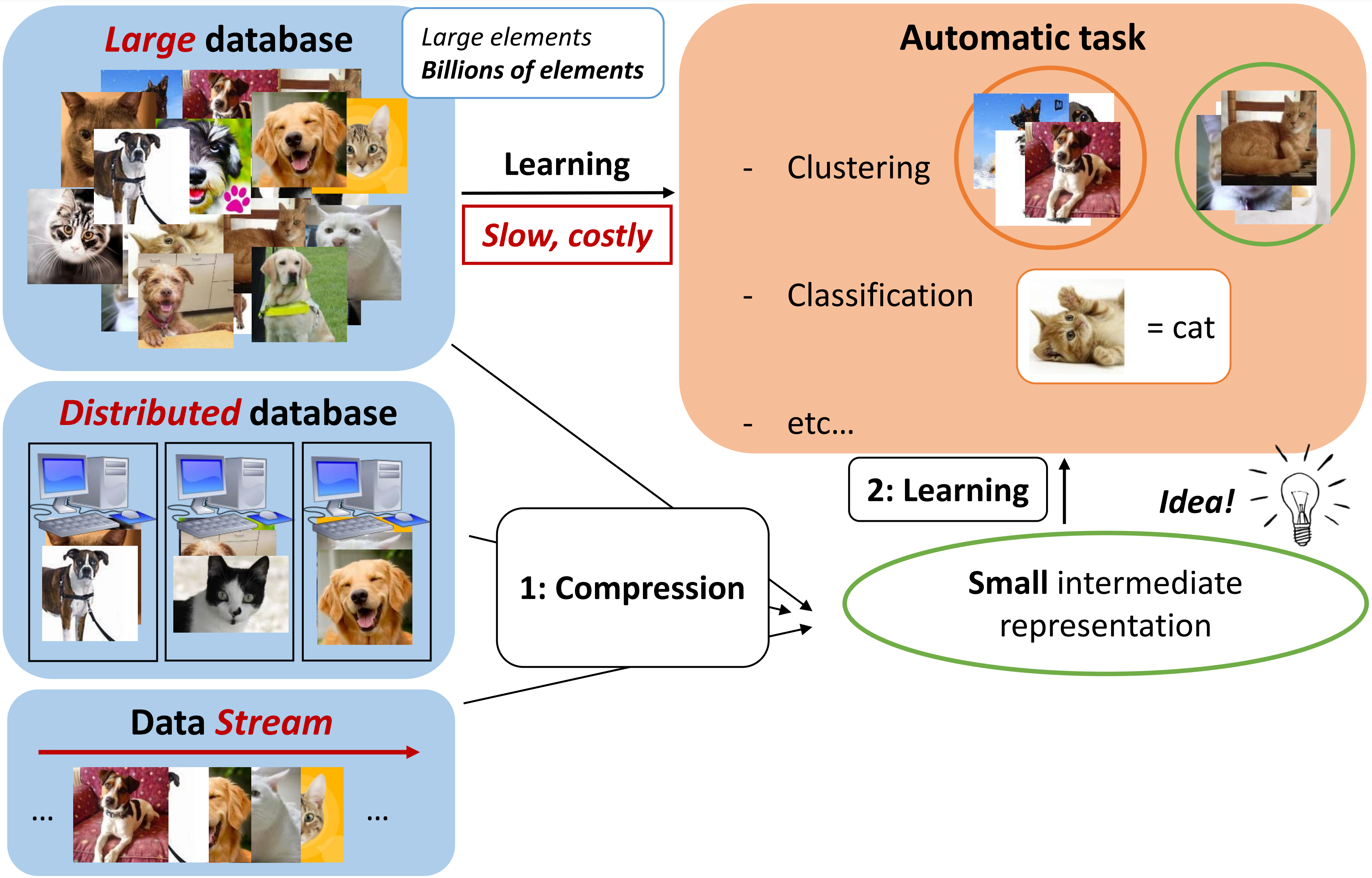
Idea!

Small intermediate representation

Data ***Stream***

…  …

Context: machine learning



Context: machine learning

Large **database**

Large elements

Billions of elements

Learning

Slow, costly

Automatic task

- Clustering

- Classification

= cat

- etc…

Distributed **database**

1: Compression

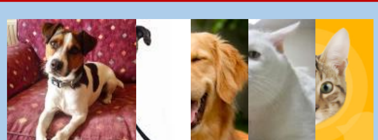
2: Learning

Idea!

Small intermediate representation

Data *Stream*

…



…

Desired properties

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

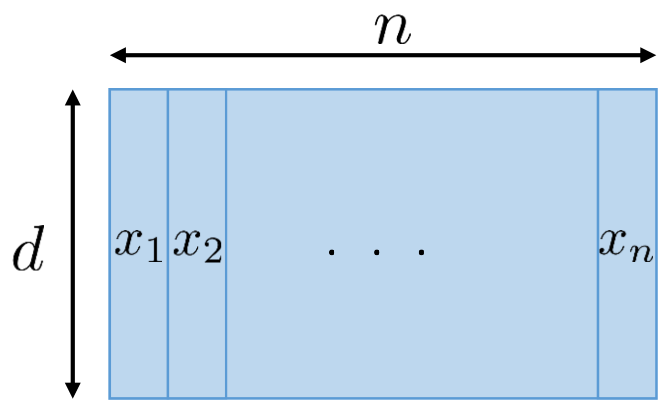
Three compression schemes

Database

Three compression schemes

Database

Feature extraction



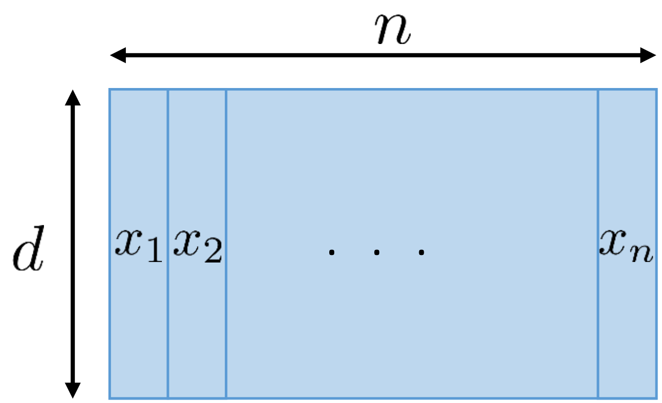
Data = Collection of vectors

Compression ?

Three compression schemes

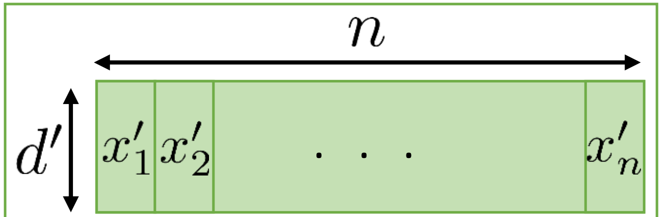
Database

Feature extraction



Data = Collection of vectors

Compression ?



Dimensionality reduction

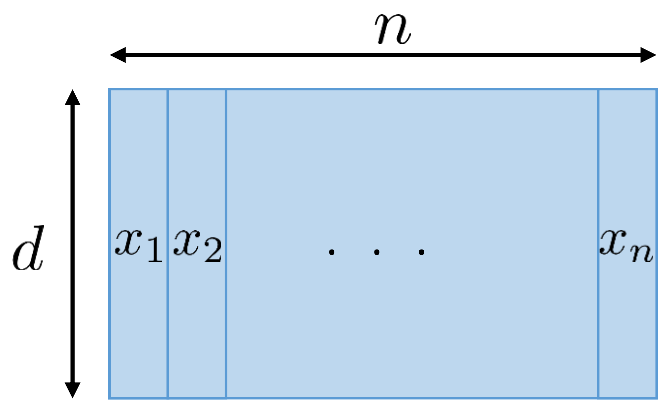
See eg *[Calderbank 2009, Boutsidis 2010]*

- Random Projection
- Feature selection

Three compression schemes

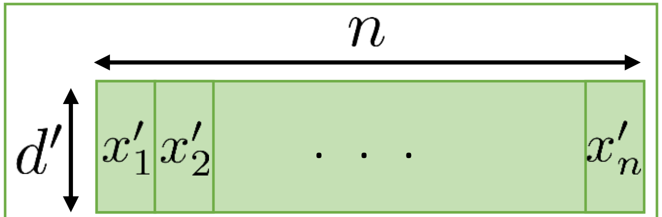
Database

Feature extraction



Data = Collection of vectors

Compression ?



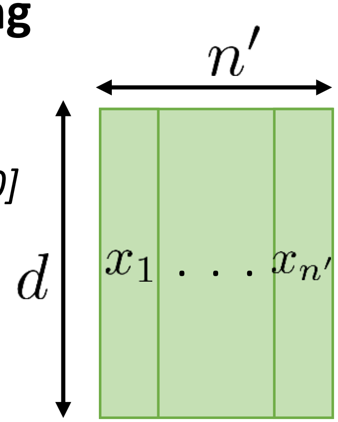
Dimensionality reduction

See eg *[Calderbank 2009, Boutsidis 2010]*

- Random Projection
- Feature selection

Subsampling coresets

See eg *[Feldman 2010]*

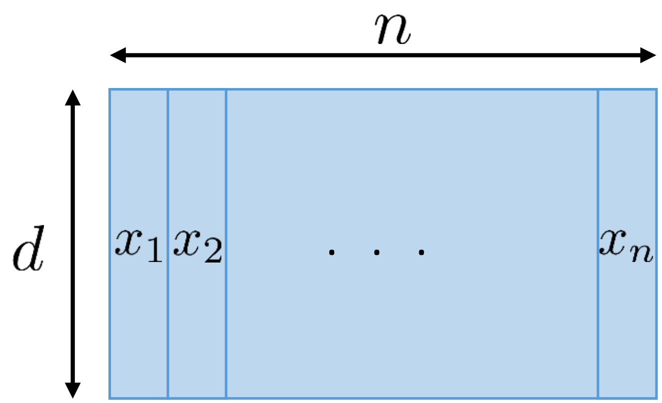


- Uniform sampling (naive)
- Adaptive sampling…

Three compression schemes

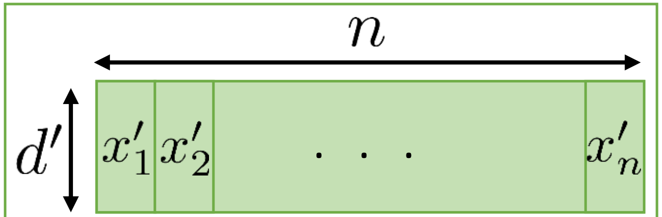
Database

Feature extraction



Data = Collection of vectors

Compression ?



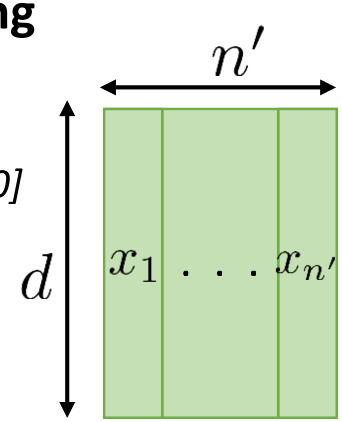
Dimensionality reduction

See eg *[Calderbank 2009, Boutsidis 2010]*

- Random Projection
- Feature selection

Subsampling coresets

See eg *[Feldman 2010]*

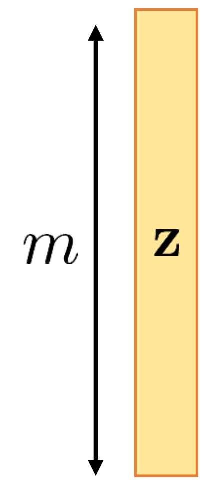


- Uniform sampling (naive)
- Adaptive sampling…

Linear sketch

See *[Thaper 2002] [Cormode 2011]*

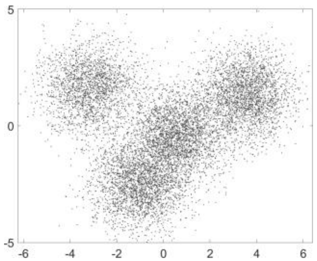
Distributed, streaming



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

Isotropic GMM estimation *[Bourrier 2013]*

Practical illustration: sketched Gaussian Mixture Model estimation with Id cov. *[Bourrier 2013]*

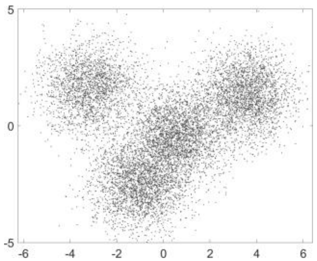
Data 

$x_1, ..., x_n \in \mathbb{R}^d$

Isotropic GMM estimation *[Bourrier 2013]*

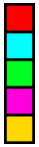
Practical illustration: sketched Gaussian Mixture Model estimation with Id cov. *[Bourrier 2013]*

Data



$x_1, ..., x_n \in \mathbb{R}^d$

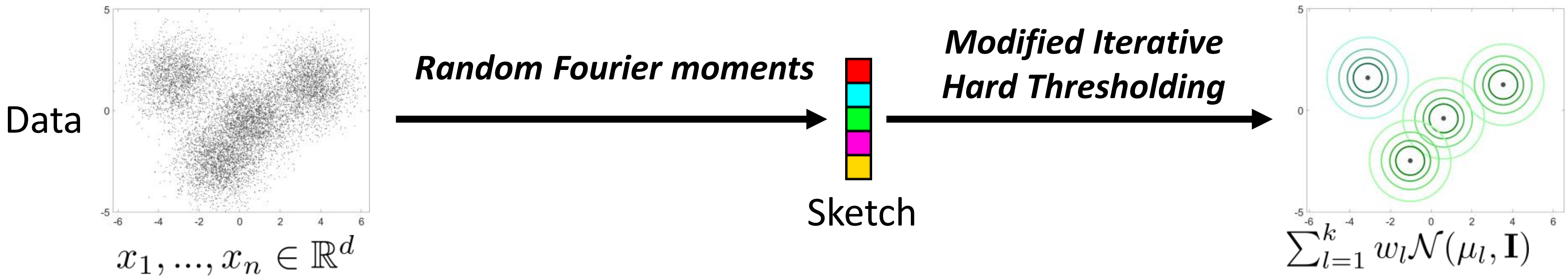
Random Fourier moments



Sketch

Isotropic GMM estimation *[Bourrier 2013]*

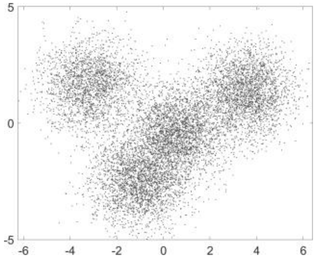
Practical illustration: sketched Gaussian Mixture Model estimation with Id cov. *[Bourrier 2013]*



Isotropic GMM estimation *[Bourrier 2013]*

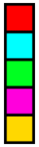
Practical illustration: sketched Gaussian Mixture Model estimation with Id cov. *[Bourrier 2013]*

Data



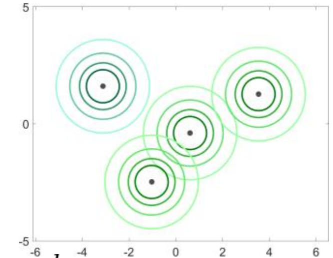
$x_1, \dots, x_n \in \mathbb{R}^d$

Random Fourier moments



Sketch

Modified Iterative Hard Thresholding



$\sum_{l=1}^{k} w_l \mathcal{N}(\mu_l, \mathbf{I})$

Observation: *necessarily*…

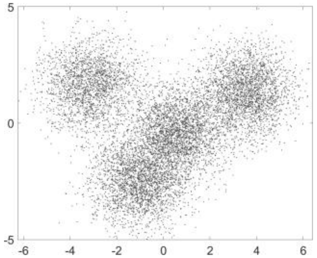
Any *linear* sketch = **empirical moments**

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

Isotropic GMM estimation *[Bourrier 2013]*

Practical illustration: sketched Gaussian Mixture Model estimation with Id cov. *[Bourrier 2013]*

Data

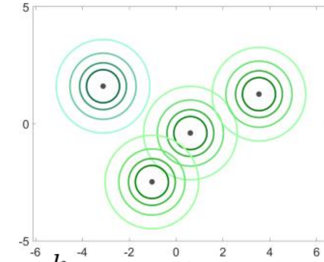


$x_1, ..., x_n \in \mathbb{R}^d$

Random Fourier moments

Sketch

Modified Iterative Hard Thresholding



$\sum_{l=1}^{k} w_l \mathcal{N}(\mu_l, \mathbf{I})$

Observation: *necessarily*…

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

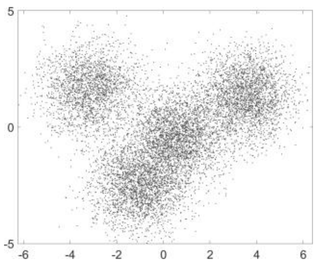
$$\boxed{\hat{\mathbf{z}} = \hat{\mathbb{E}}\Phi(X)} = \frac{1}{n}\sum_i \Phi(x_i)$$

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

Isotropic GMM estimation *[Bourrier 2013]*

Practical illustration: sketched Gaussian Mixture Model estimation with Id cov. *[Bourrier 2013]*

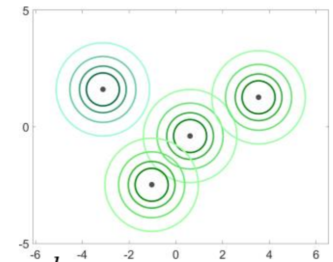
Data



$x_1, \ldots, x_n \in \mathbb{R}^d$

Random Fourier moments → Sketch

Modified Iterative Hard Thresholding →



$\sum_{l=1}^{k} w_l \mathcal{N}(\mu_l, \mathbf{I})$

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

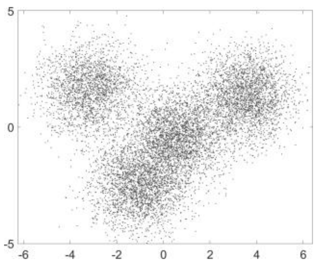
$$\boxed{\min_\theta \|\hat{\mathbf{z}} - \mathbb{E}_\theta \Phi(X)\|}$$

True moments (param. θ)

Isotropic GMM estimation *[Bourrier 2013]*

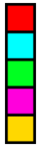
Practical illustration: sketched Gaussian Mixture Model estimation with Id cov. *[Bourrier 2013]*

Data



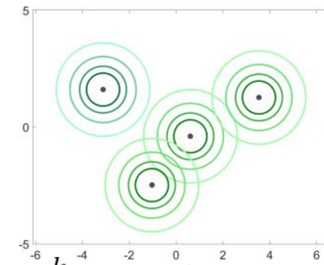
$x_1, \ldots, x_n \in \mathbb{R}^d$

Random Fourier moments



Sketch

Modified Iterative Hard Thresholding



$\sum_{l=1}^{k} w_l \mathcal{N}(\mu_l, \mathbf{I})$

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)

Good empirical properties of the « sketching » function Φ

- **« Sufficient » dimension m (size of the sketch)**
- **Randomly designed**

Contributions

Questions

Contributions

Questions

- Generalize to other mixture models? New algorithm?
- **Theoretical guarantees?**

Contributions

Questions

- Generalize to other mixture models? New algorithm?
- **Theoretical guarantees?**

Contributions of this thesis

Contributions

Questions

- Generalize to other mixture models? New algorithm?
- **Theoretical guarantees?**

Contributions of this thesis

- **Algorithmic**: heuristic **greedy algorithm** for **any** sketched mixture model estimation
  - General GMM estimation
  - Sketched k-means
  - **Mixture of multivariate elliptic α-stable distributions estimation**

Contributions

Questions

- Generalize to other mixture models? New algorithm?
- **Theoretical guarantees?**

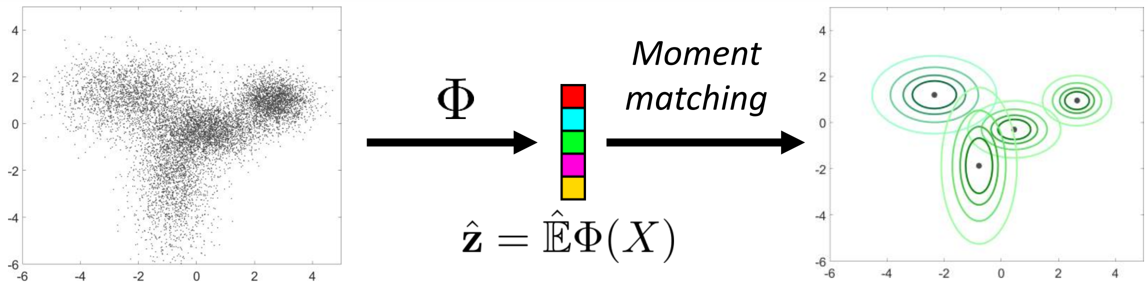
Contributions of this thesis

- **Algorithmic**: heuristic **greedy algorithm** for **any** sketched mixture model estimation
  - General GMM estimation
  - Sketched k-means
  - **Mixture of multivariate elliptic α-stable distributions estimation**
- **Theoretical:** Information-preservation guarantees
  - Recovery conditions for **generic models**
  - Additional focus on mixture models

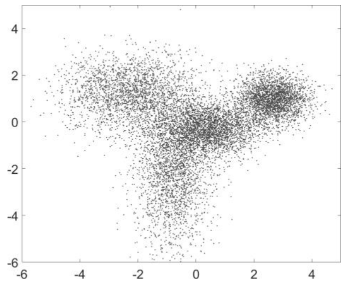
Outline

Outline

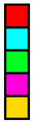
Sketched mixture model estimation



Sketched mixture model estimation

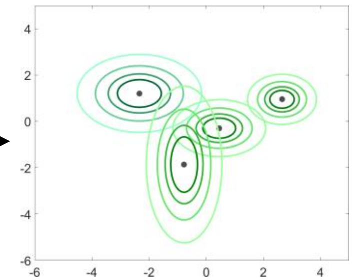


$$\Phi$$



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

Moment matching



Goal

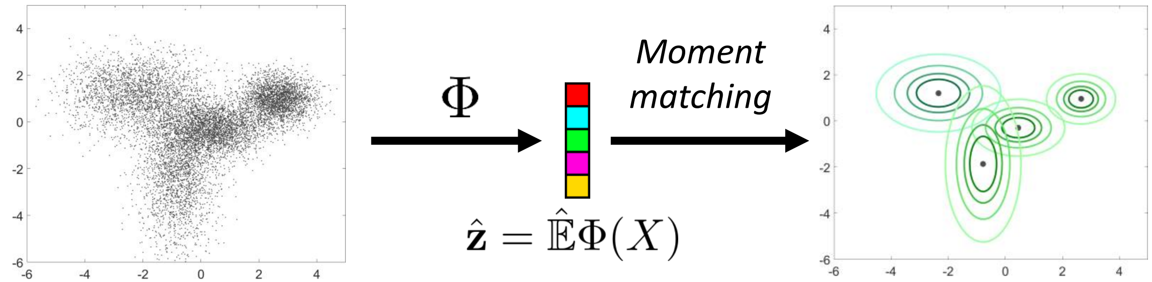
- Estimate mixture model:

$$\boxed{x_i \sim \sum_{l=1}^{k} w_l \pi_{\theta_l}}$$

$$w_l \geq 0, \ \ \sum_l w_l = 1$$

Ex: $\pi_\theta = \mathcal{N}(\mu, \Sigma)$

Sketched mixture model estimation



Goal

- Estimate mixture model:

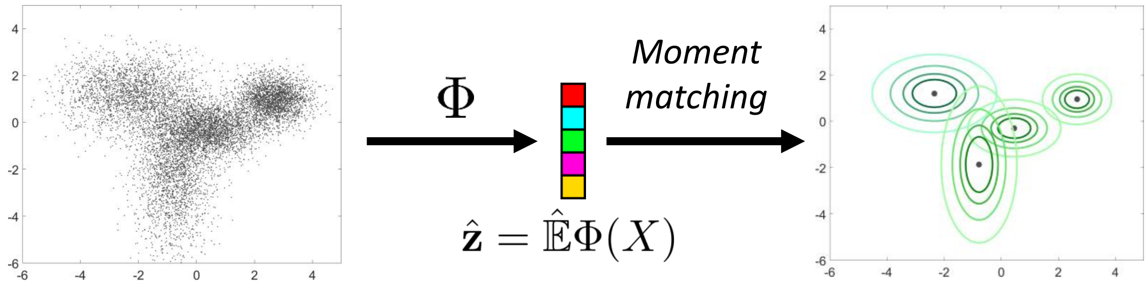
$$\boxed{x_i \sim \sum_{l=1}^{k} w_l \pi_{\theta_l}}$$

$$w_l \geq 0, \quad \sum_l w_l = 1$$

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

Ex: $\pi_\theta = \mathcal{N}(\mu, \Sigma)$

Sketched mixture model estimation



Goal

- Estimate mixture model:

$$\boxed{x_i \sim \sum_{l=1}^{k} w_l \pi_{\theta_l}}$$

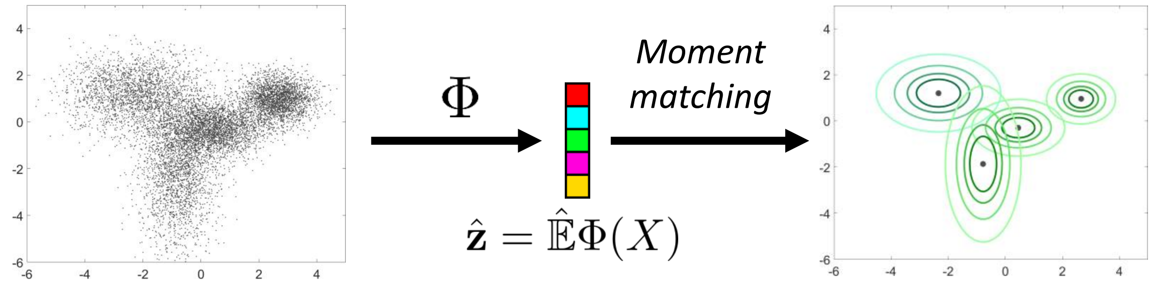
$$w_l \geq 0, \;\; \sum_l w_l = 1$$

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

Ex: $\pi_\theta = \mathcal{N}(\mu, \Sigma)$

Method: moment matching

Sketched mixture model estimation



Goal

- Estimate mixture model:

$$\boxed{x_i \sim \sum_{l=1}^k w_l \pi_{\theta_l}}$$

$$w_l \geq 0, \quad \sum_l w_l = 1$$

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

Ex: $\pi_\theta = \mathcal{N}(\mu, \Sigma)$

Method: moment matching

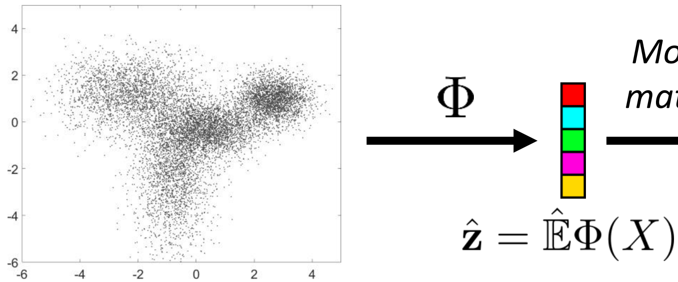
Written as

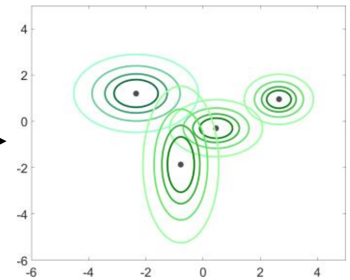
$$\boxed{\min_{\theta_l, w} \|\hat{\mathbf{z}} - \sum_{l=1}^k w_l f(\theta_l)\|_2}$$

where

$$\boxed{f(\theta) := \mathbb{E}_{X\sim\pi_\theta}\Phi(X)}$$

Sketched mixture model estimation





Goal

- Estimate mixture model:

$$\boxed{x_i \sim \sum_{l=1}^{k} w_l \pi_{\theta_l}}$$

$$w_l \geq 0, \quad \sum_l w_l = 1$$

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

Ex: $\pi_\theta = \mathcal{N}(\mu, \Sigma)$

Method: moment matching

Written as

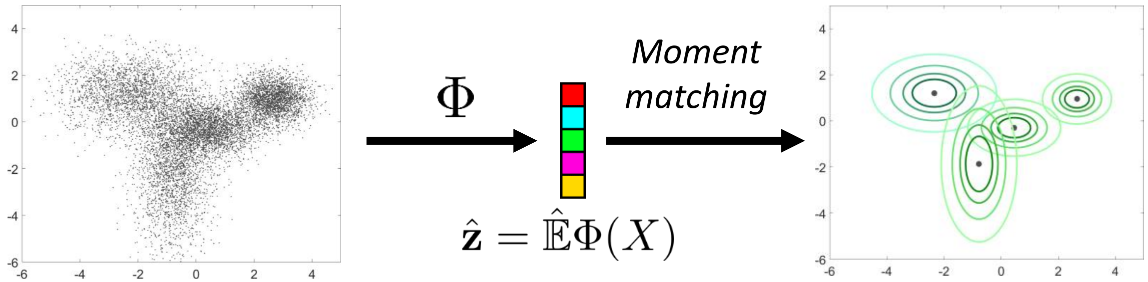
$$\boxed{\min_{\theta_l, w} \|\hat{\mathbf{z}} - \sum_{l=1}^{k} w_l f(\theta_l)\|_2}$$

where

$$\boxed{f(\theta) := \mathbb{E}_{X \sim \pi_\theta} \Phi(X)}$$

- Non-convex minimization

Sketched mixture model estimation



Goal

- Estimate mixture model:

$$\boxed{x_i \sim \sum_{l=1}^k w_l \pi_{\theta_l}}$$

$$w_l \geq 0, \quad \sum_l w_l = 1$$

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

Ex: $\pi_\theta = \mathcal{N}(\mu, \Sigma)$

Method: moment matching

Written as

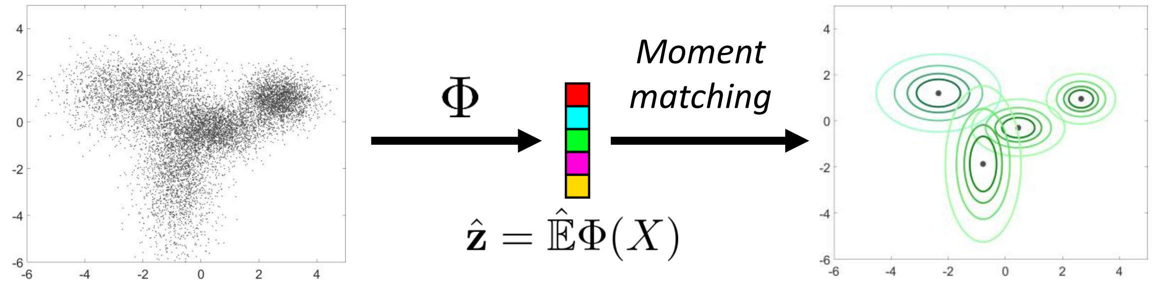
$$\boxed{\min_{\theta_l, w} \left\|\hat{\mathbf{z}} - \sum_{l=1}^k w_l f(\theta_l)\right\|_2}$$

where

$$\boxed{f(\theta) := \mathbb{E}_{X\sim\pi_\theta}\Phi(X)}$$

- Non-convex minimization
- Convex relaxation? (super-resolution)

Sketched mixture model estimation



Goal

- Estimate mixture model:

$$\boxed{x_i \sim \sum_{l=1}^k w_l \pi_{\theta_l}}$$

$$w_l \geq 0, \quad \sum_l w_l = 1$$

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

Ex: $\pi_\theta = \mathcal{N}(\mu, \Sigma)$

Method: moment matching

Written as

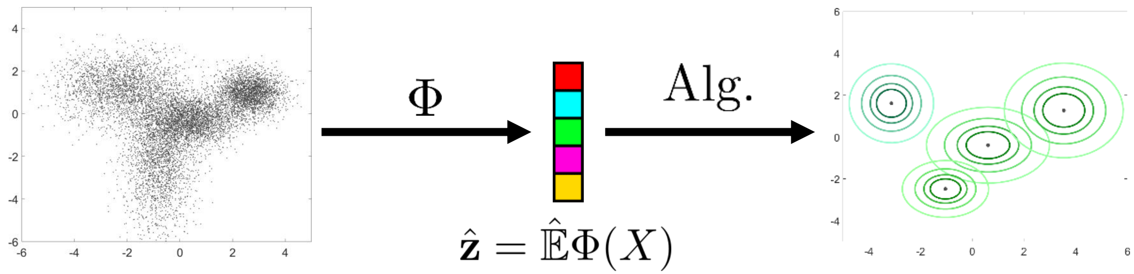
$$\boxed{\min_{\theta_l, w} \left\| \hat{\mathbf{z}} - \sum_{l=1}^k w_l f(\theta_l) \right\|_2}$$

where

$$\boxed{f(\theta) := \mathbb{E}_{X \sim \pi_\theta} \Phi(X)}$$

- Non-convex minimization
- Convex relaxation? (super-resolution)
- **Proposed approach: greedy heuristic**

Algorithm

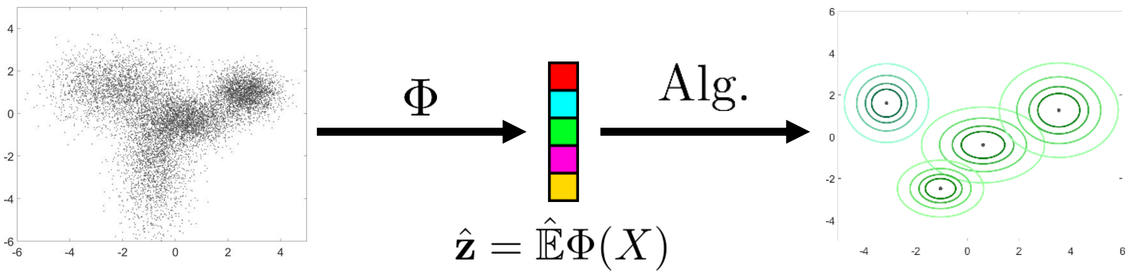


Algorithm: Compressive Learning OMPR (CL-OMPR)

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

[Jain 2011]

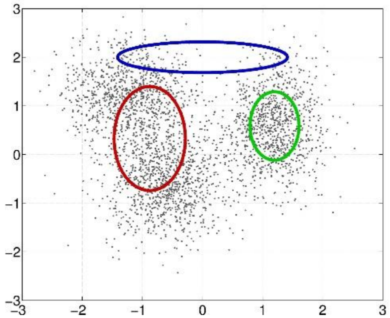
Algorithm



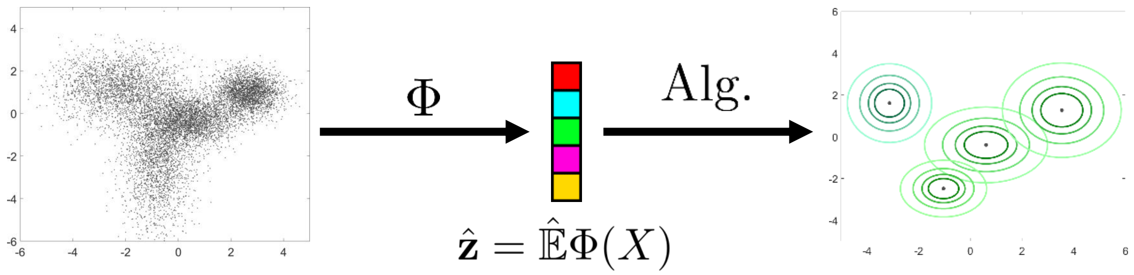
Algorithm: Compressive Learning OMPR (CL-OMPR)

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

[Jain 2011]



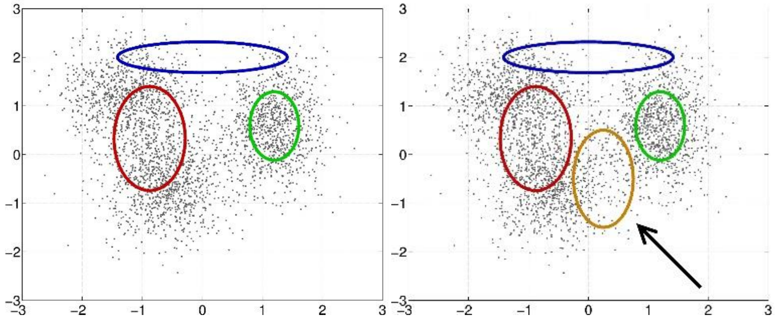
Algorithm



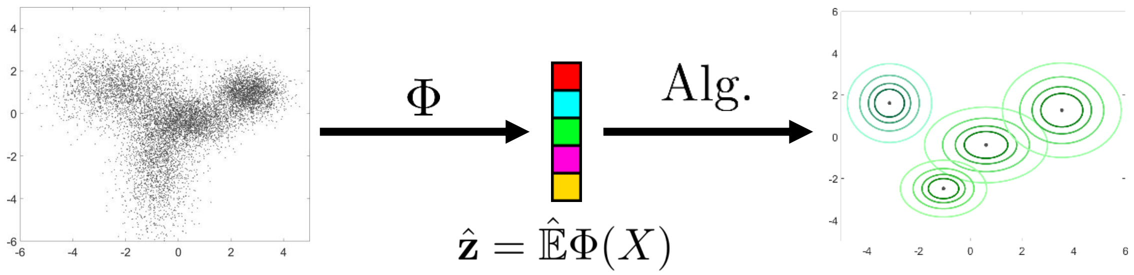
Algorithm: Compressive Learning OMPR (CL-OMPR)

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

[Jain 2011]



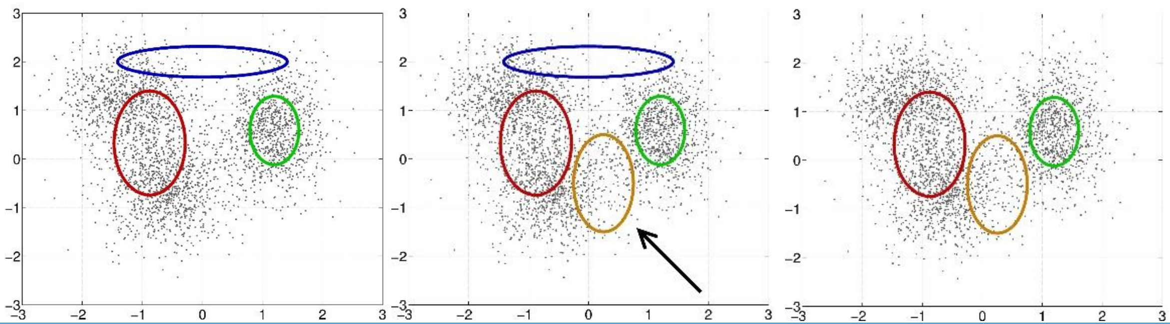
Algorithm

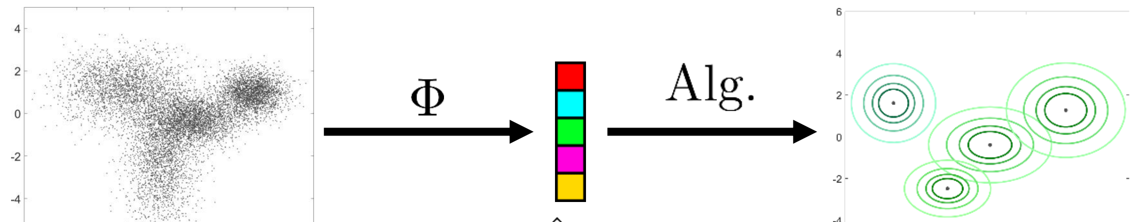


Algorithm: Compressive Learning OMPR (CL-OMPR)

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

[Jain 2011]



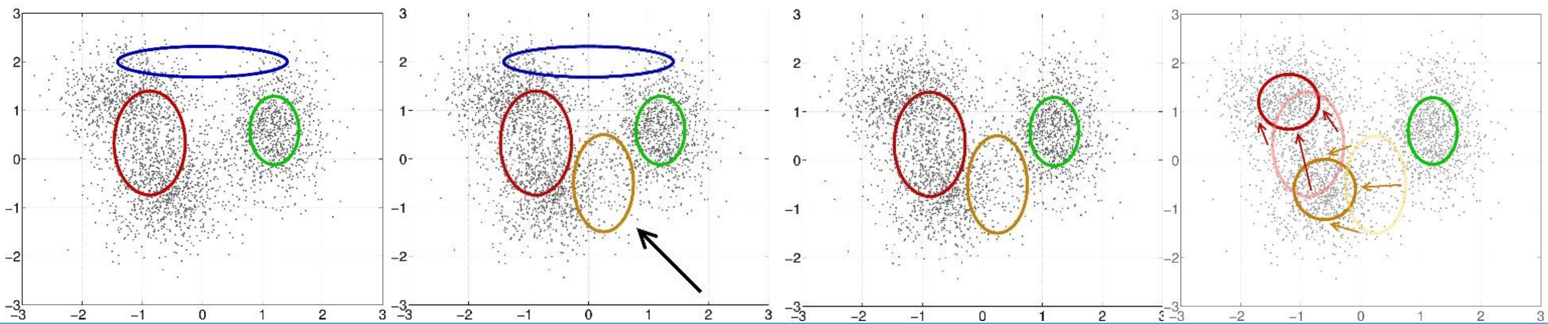


$$\hat{\mathbf{z}} = \hat{\mathbb{E}}\Phi(X)$$

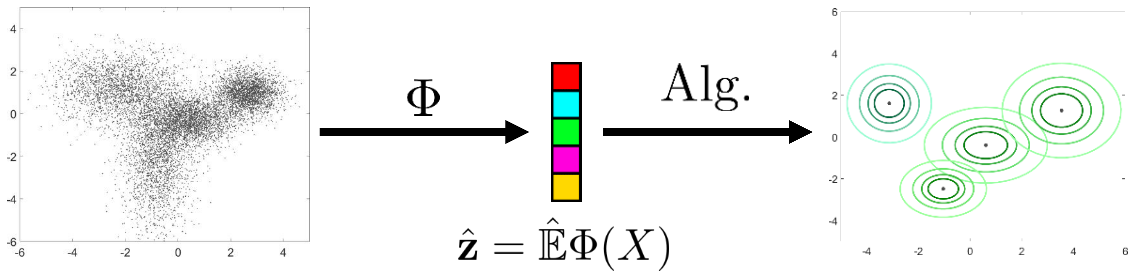
Algorithm: Compressive Learning OMPR (CL-OMPR)

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

[Jain 2011]

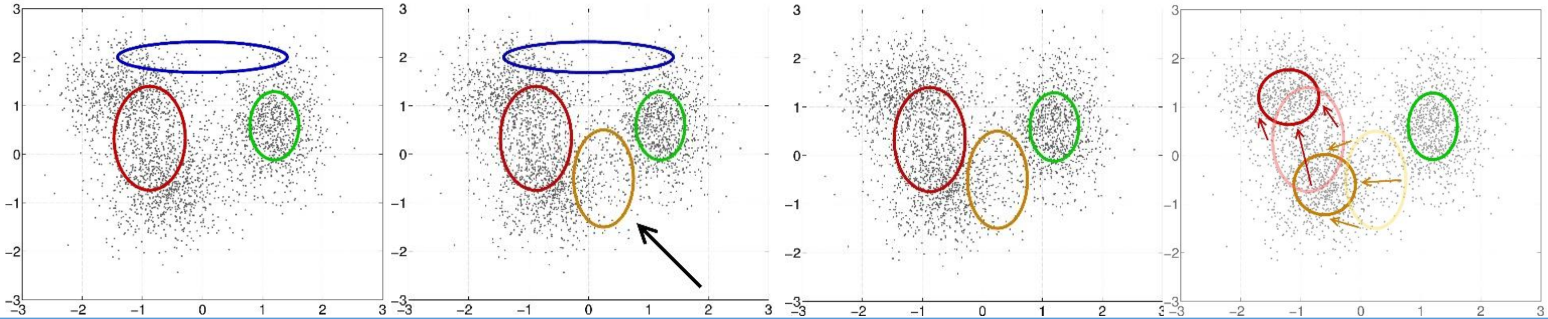


Algorithm



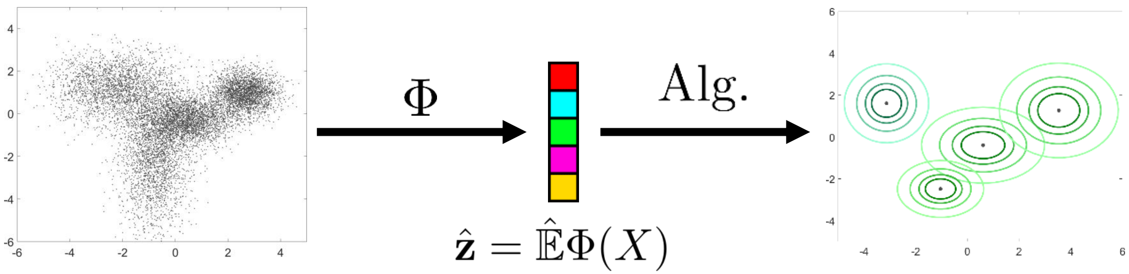
Algorithm: Compressive Learning OMPR (CL-OMPR)

Continuous (off-the-grid) adaptation of *Orthogonal Matching Pursuit with Replacement* *[Jain 2011]*



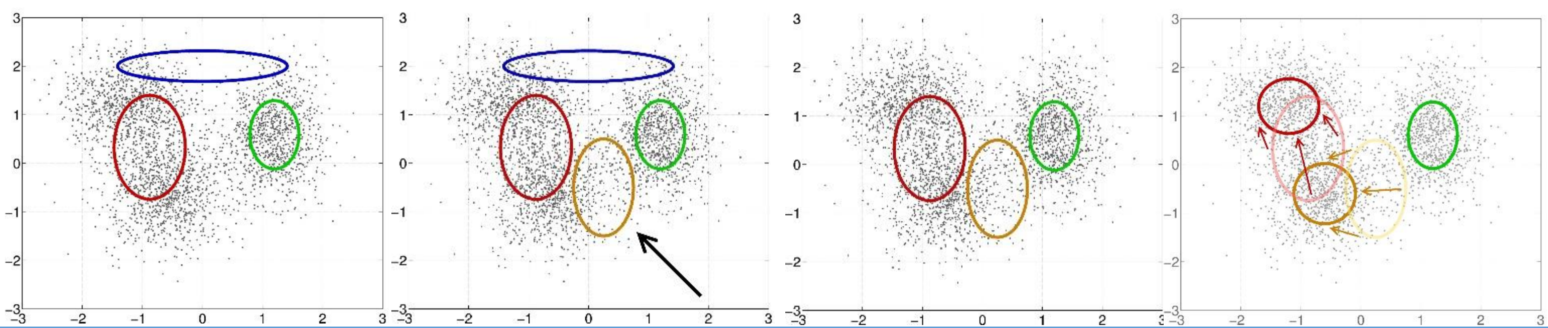
Can be applied if: $f(\theta) = \mathbb{E}_{\pi_\theta}\Phi(X)$ **has a closed-form, differentiable expression**

Algorithm



Algorithm: Compressive Learning OMPR (CL-OMPR)

Continuous (off-the-grid) adaptation of *Orthogonal Matching Pursuit with Replacement [Jain 2011]*

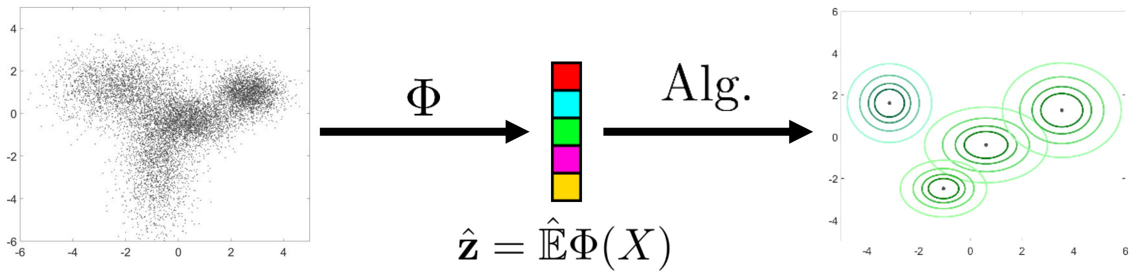


Can be applied if: $f(\theta) = \mathbb{E}_{\pi_\theta}\Phi(X)$ **has a closed-form, differentiable expression**

In experiments:

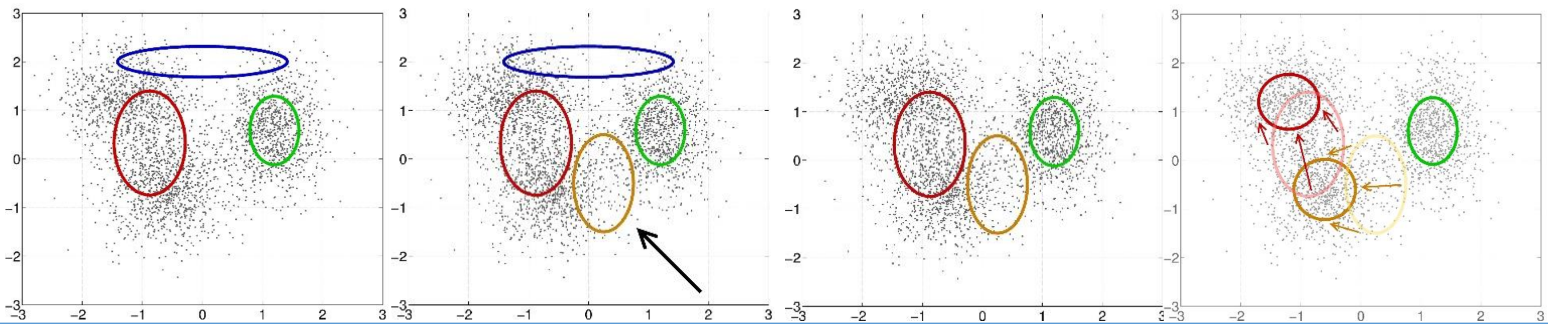
Φ : Random Fourier sampling *[Bourrier 2013]* (with new distribution of frequencies)

Algorithm



Algorithm: Compressive Learning OMPR (CL-OMPR)

Continuous (off-the-grid) adaptation of *Orthogonal Matching Pursuit with Replacement* *[Jain 2011]*



Can be applied if: $f(\theta) = \mathbb{E}_{\pi_\theta}\Phi(X)$ **has a closed-form, differentiable expression**

In experiments:

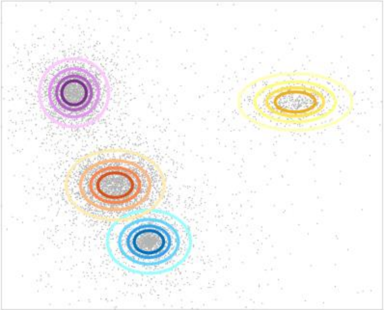
Φ: Random Fourier sampling *[Bourrier 2013]* (with new distribution of frequencies)

Model such that: π_θ **has a closed-form characteristic function**

Outline

Models

GMM diagonal cov.



Sketched mixture model estimation

Available at *sketchml.gforge.inria.fr*

$$\pi_\theta = \mathcal{N}(\mu, diag(\sigma))$$
$$\theta = (\mu, \sigma) \in \mathbb{R}^{2d}$$

Classic approach on full data

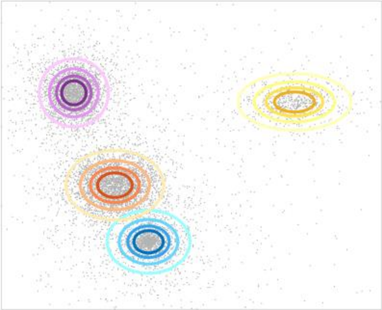
Algorithm : **EM**

[Dempster 1977]

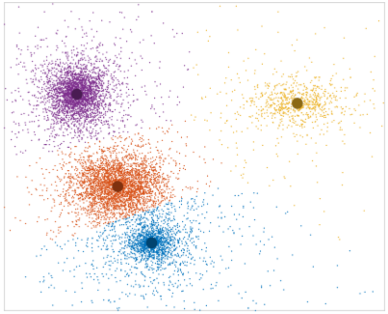
(VLFeat's `gmm`*)*

Models

GMM diagonal cov.



Mixture of Diracs



Sketched mixture model estimation

Available at *sketchml.gforge.inria.fr*

$$\pi_\theta = \mathcal{N}(\mu, diag(\sigma))$$
$$\theta = (\mu, \sigma) \in \mathbb{R}^{2d}$$

$$\pi_\theta = \delta_\theta \qquad \theta \in \mathbb{R}^d$$
(clustered distribution = noisy mixture of Diracs)

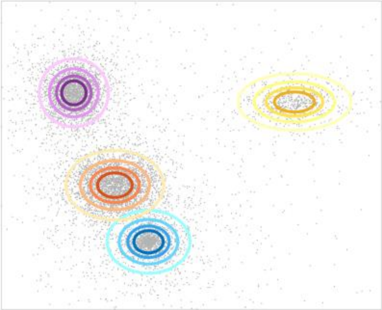
Classic approach on full data

Algorithm : **EM**
[Dempster 1977]
(VLFeat's `gmm`*)*

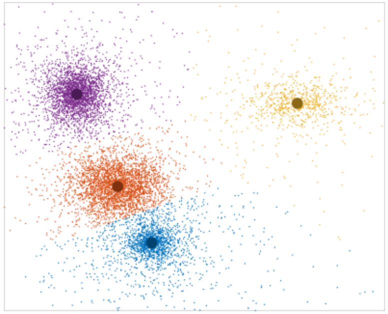
Algorithm : **k-means**
[Lloyd 1982]
(Matlab's `kmeans`*)*

Models

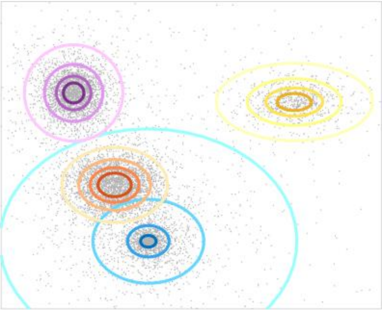
GMM diagonal cov.



Mixture of Diracs



Mixture of stable dist.



Sketched mixture model estimation

Available at *sketchml.gforge.inria.fr*

$$\pi_\theta = \mathcal{N}(\mu, diag(\sigma))$$
$$\theta = (\mu, \sigma) \in \mathbb{R}^{2d}$$

$$\pi_\theta = \delta_\theta \qquad \theta \in \mathbb{R}^d$$
(clustered distribution = noisy mixture of Diracs)

$$\pi_\theta = \mathcal{S}_\alpha(\mu, diag(\sigma))$$
$$\theta = (\mu, \sigma, \alpha) \in \mathbb{R}^{2d+1}$$

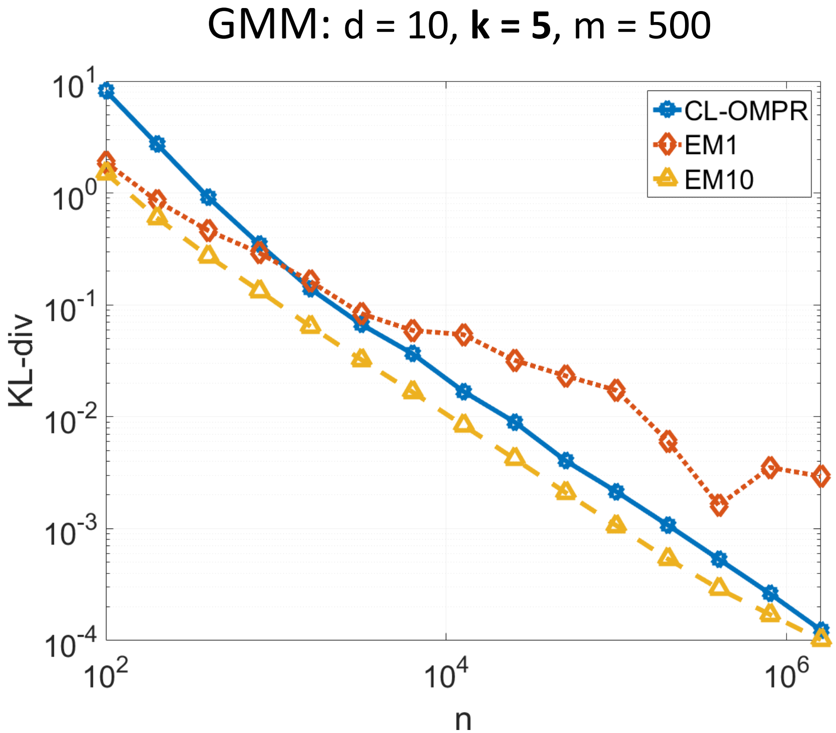
Classic approach on full data

Algorithm : **EM**
[Dempster 1977]
(VLFeat's `gmm`*)*

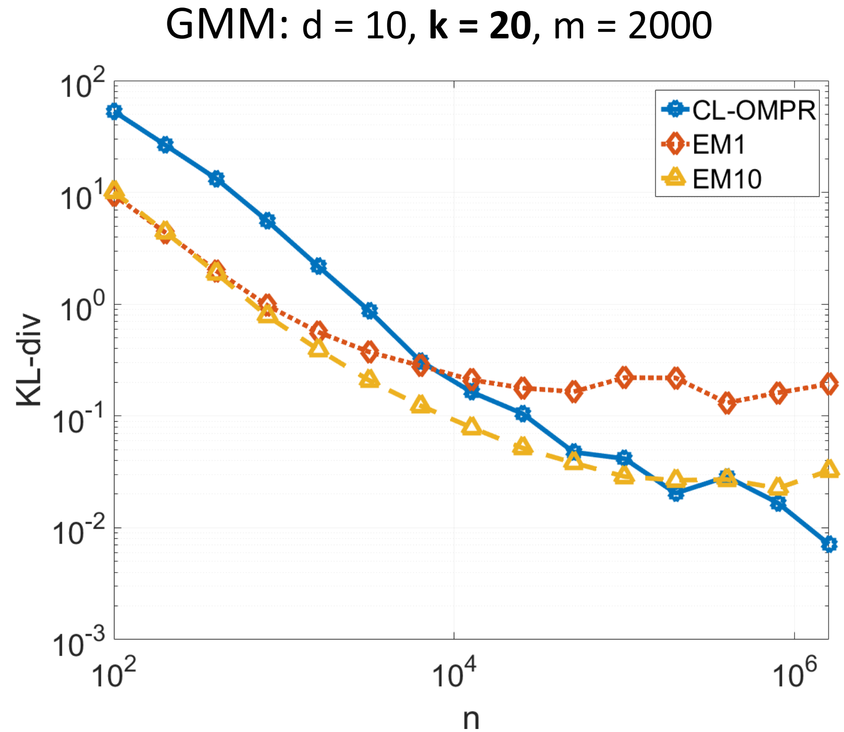
Algorithm : **k-means**
[Lloyd 1982]
(Matlab's `kmeans`*)*

None!
1-D *method with MCMC: can be very long…*

Large-scale evaluation on synthetic data



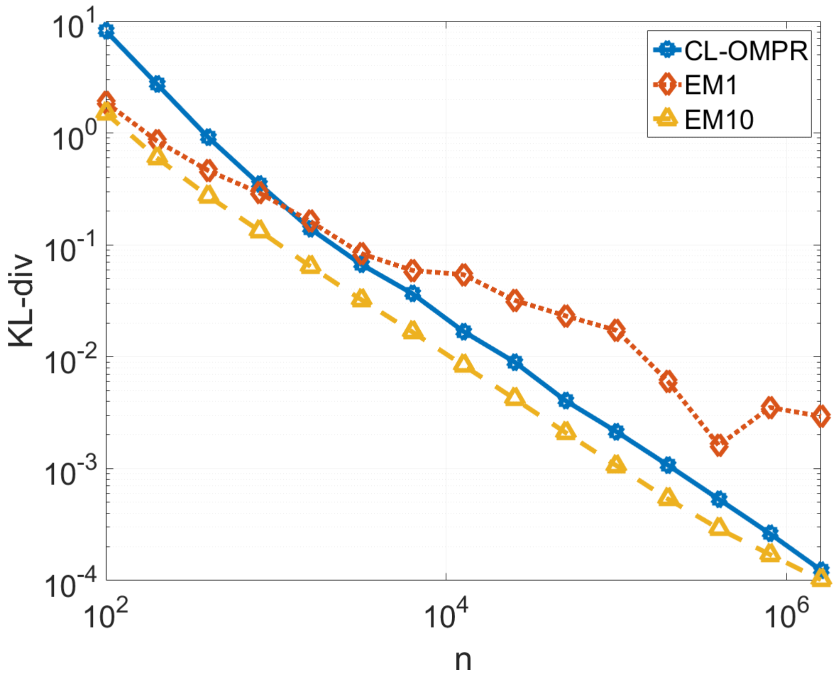
Size of database



Size of database

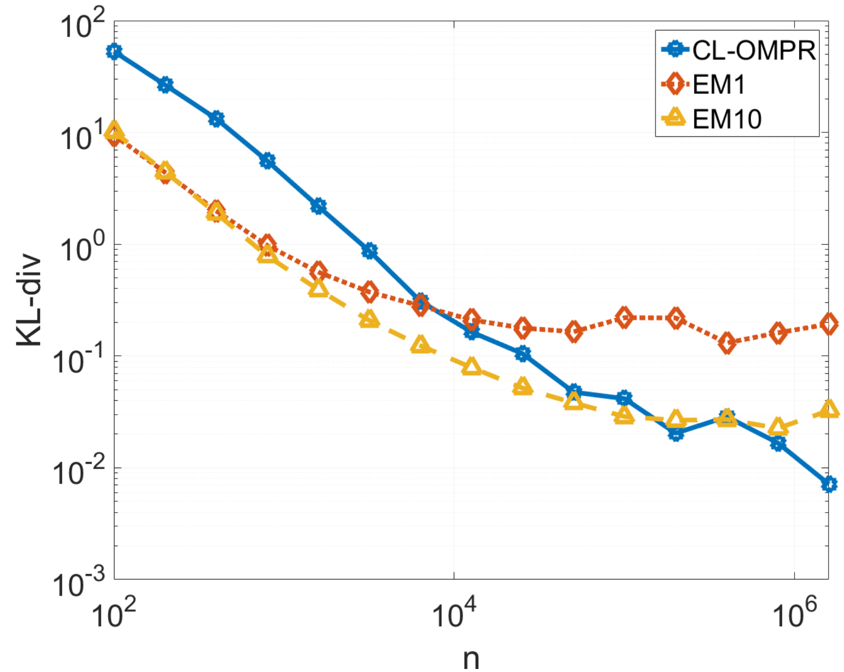
Large-scale evaluation on synthetic data

GMM: d = 10, **k = 5**, m = 500



Size of database

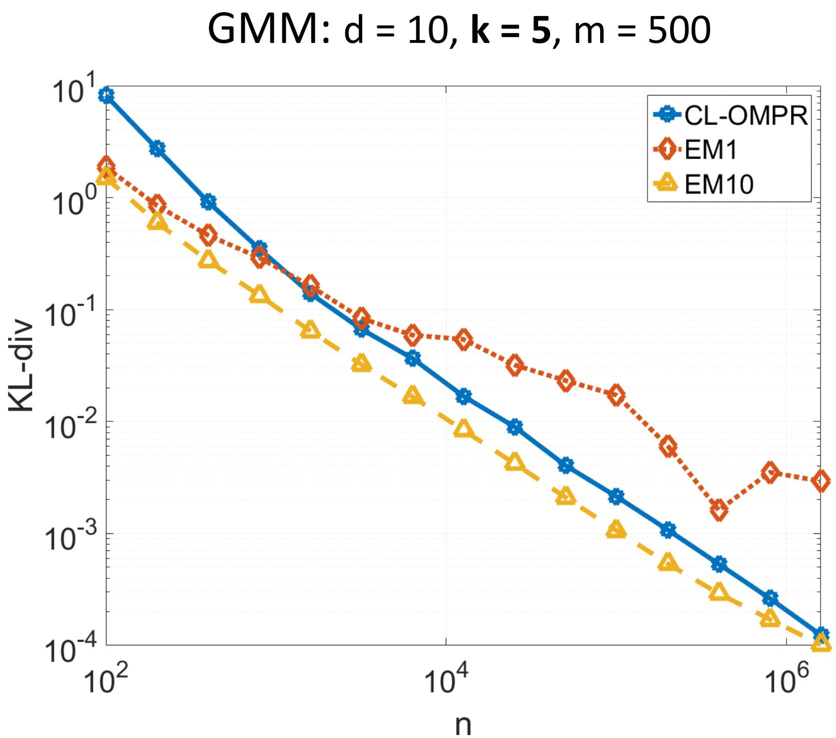
GMM: d = 10, **k = 20**, m = 2000



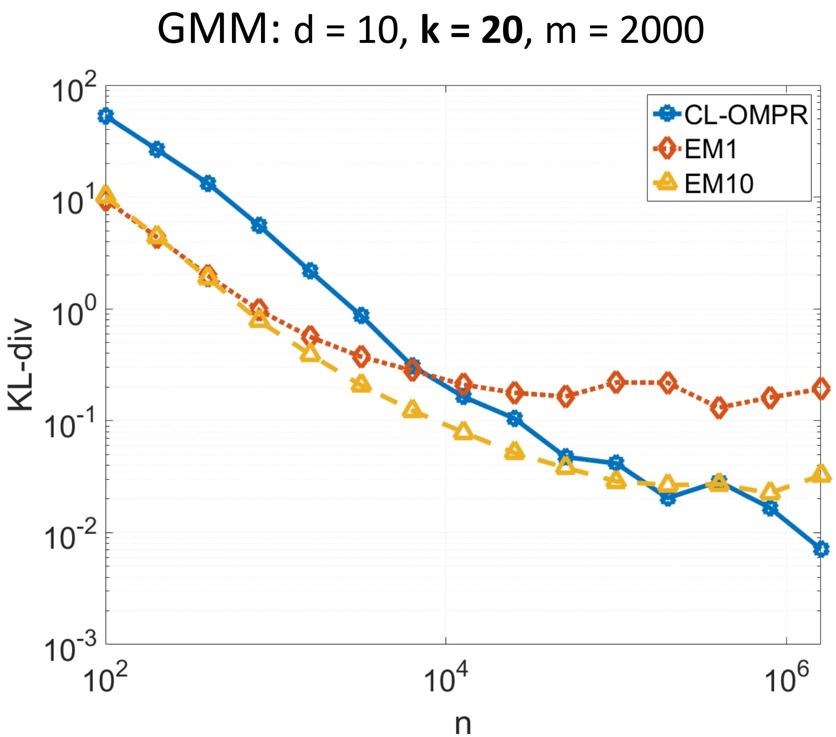
Size of database

- **Does not need** replicates (despite some randomness in CL-OMPR)

Large-scale evaluation on synthetic data



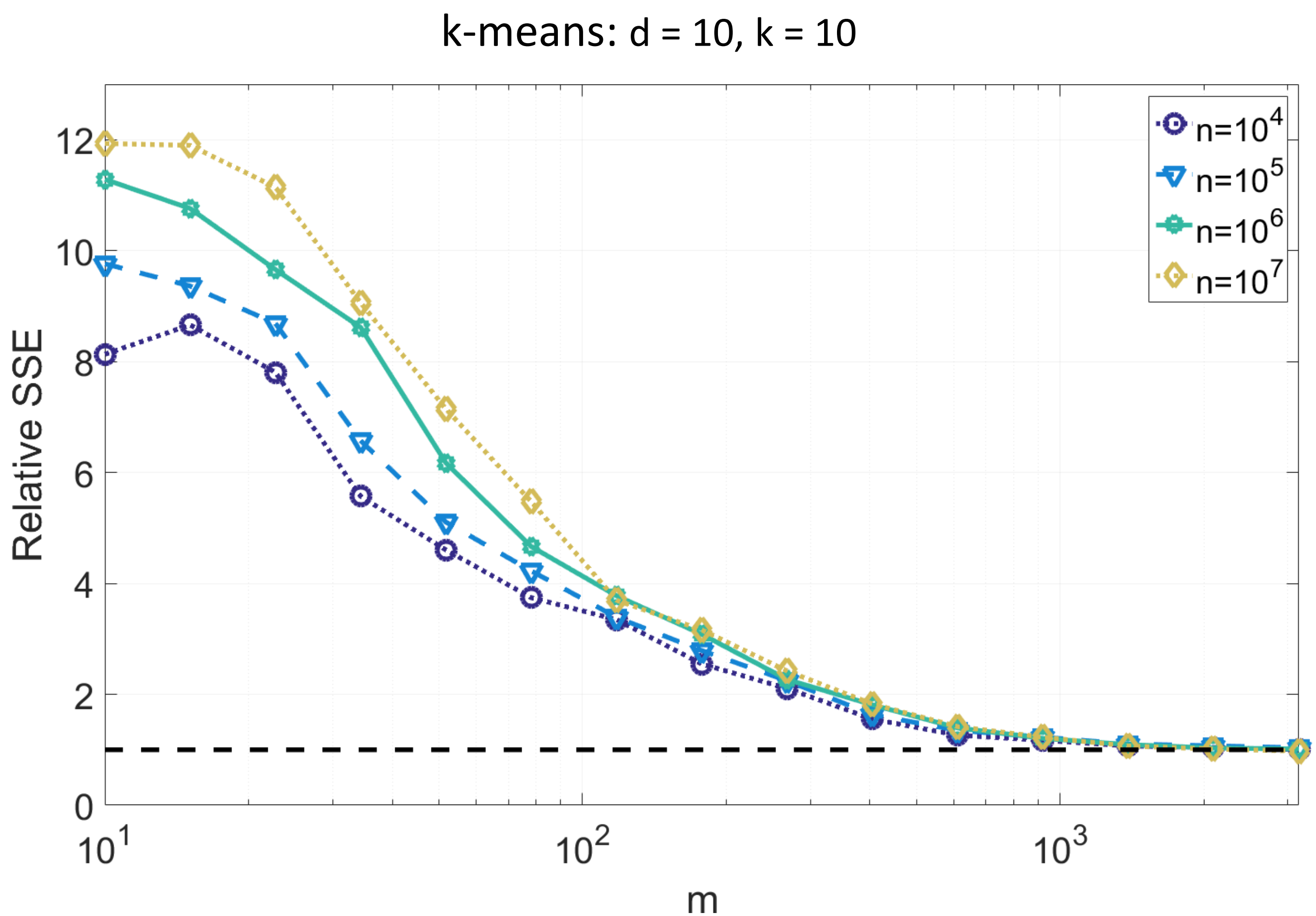
Size of database



Size of database

- **Does not need** replicates (despite some randomness in CL-OMPR)
- Comparatively better on large databases *(despite fixed sketch size)*

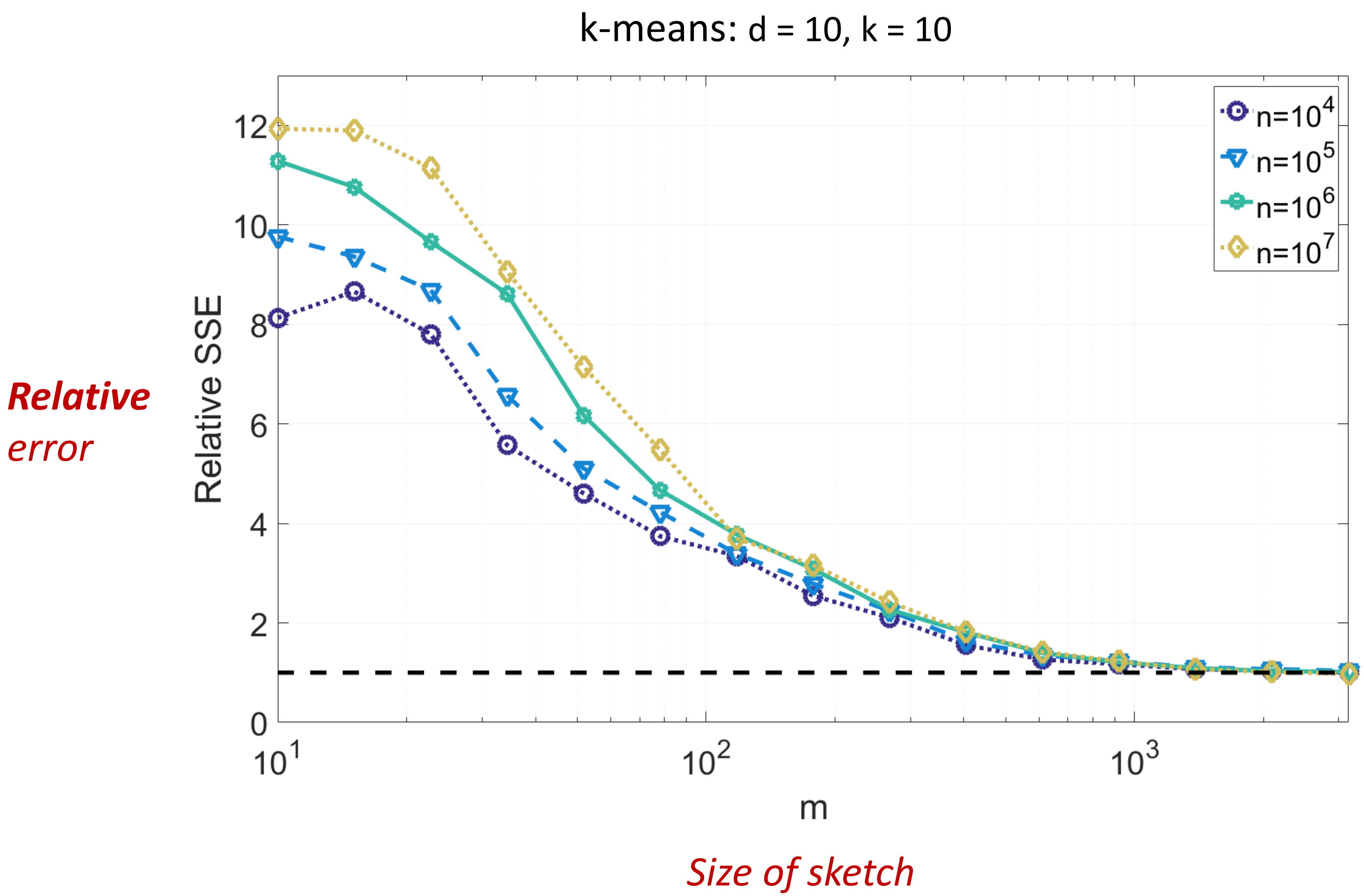
Large-scale evaluation on synthetic data



Relative *error*

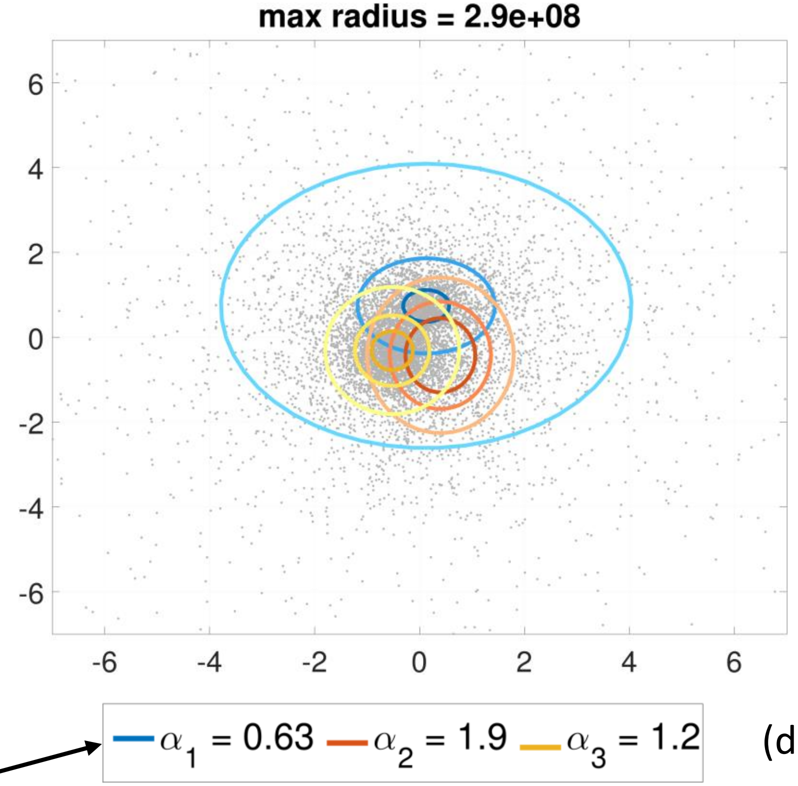
Size of sketch

Large-scale evaluation on synthetic data



- Size of sketch m **independent** of size of data n
  - Intuitively: dependent on complexity of the problem k, d …

Alpha stable on synthetic data: toy example



(d=10, k=3)

Very heavy tailed…

Toy example

- CL-OMPR able to precisely estimate all parameters

(10^{-2} precision in approx. 80 sec)

*(reported result for **1D** approaches with MCMC: 10^{-1} precision in 1.5 hours)*

Application on real data

- Efficient at large scales even on real data?

Application on real data

- Efficient at large scales even on real data?

*Classic method for **speaker verification*** *[Reynolds 2000] (for proof of concept) NIST 2005 database, MFCCs.*

GMM (d=12, k=64, m=10000)

Results (EER, lower is better)

- EM on **300 000** MFCCs: **29.53**
- Sketch on **200 millions** MFCCs: **28.96**

(120 000-fold compression)

Application on real data

- Efficient at large scales even on real data?

*Classic method for **speaker verification** [Reynolds 2000] (for proof of concept) NIST 2005 database, MFCCs.*

GMM (d=12, k=64, m=10000)

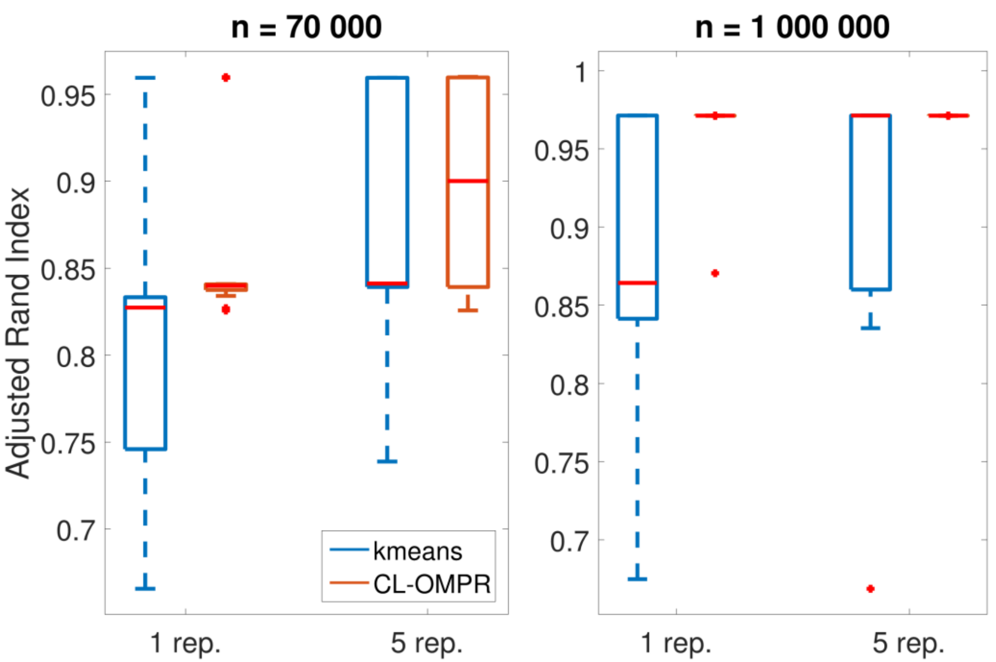
Results (EER, lower is better)

- EM on **300 000** MFCCs: **29.53**
- Sketch on **200 millions** MFCCs: **28.96**

(120 000-fold compression)

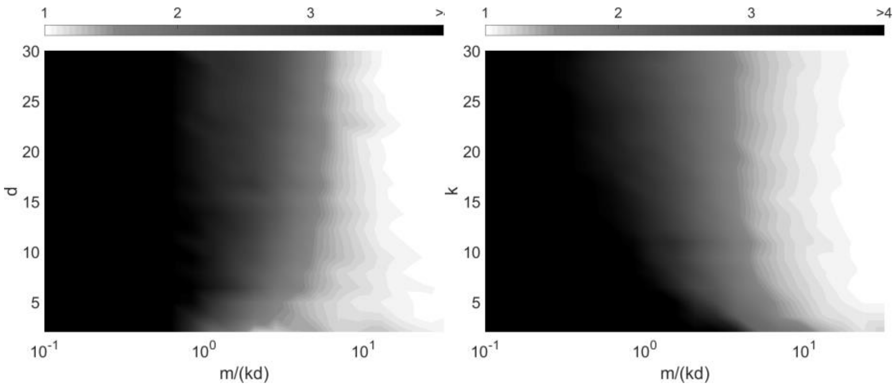
***Spectral clustering** for classification [Uw 2001], augmented MNIST database [Loosli 2007].*

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



How big a sketch ?

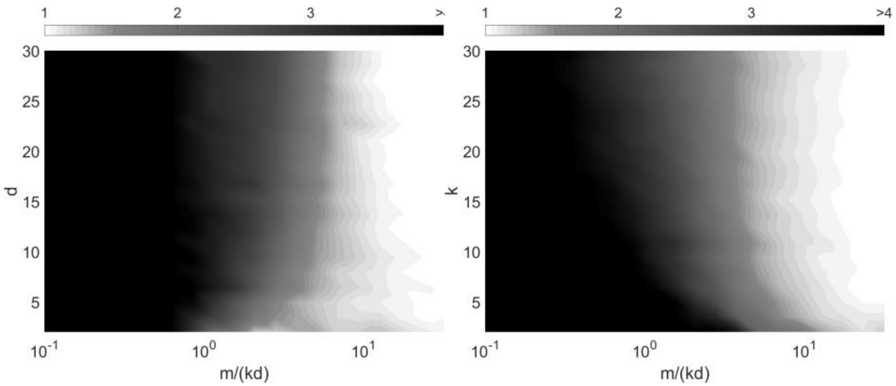
k-means



Relative sketch size m/(kd)

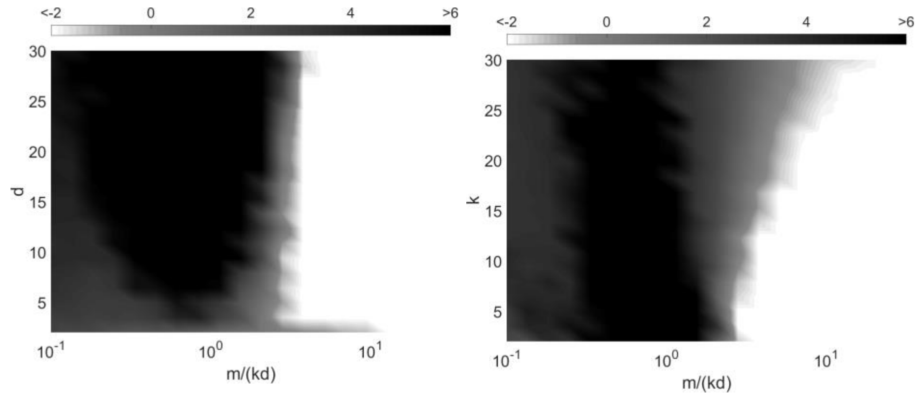
How big a sketch ?

k-means



Relative sketch size m/(kd)

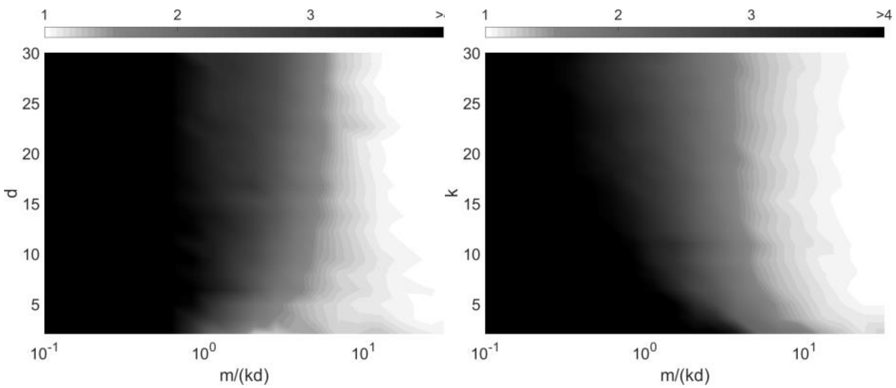
GMM



Relative sketch size

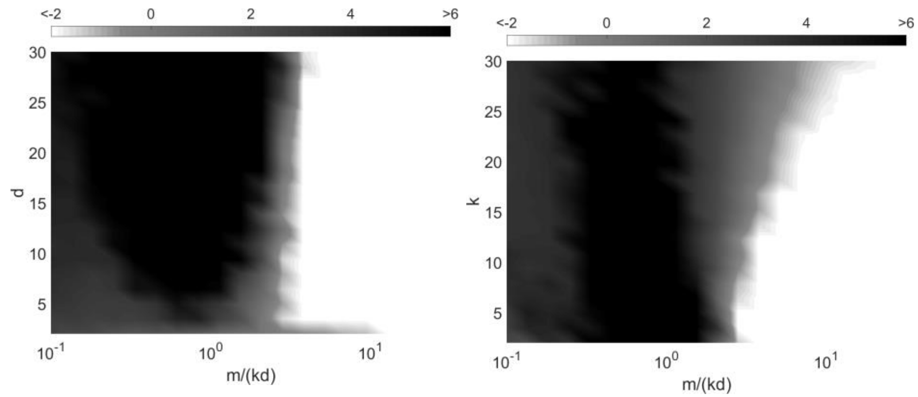
How big a sketch ?

k-means



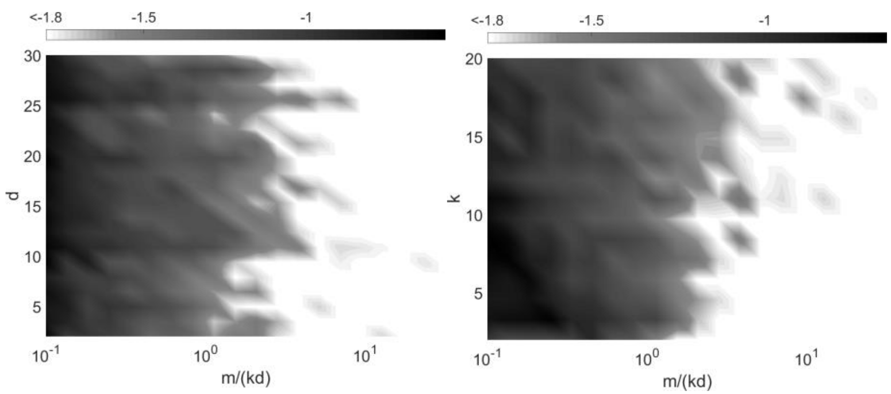
Relative sketch size m/(kd)

GMM



Relative sketch size

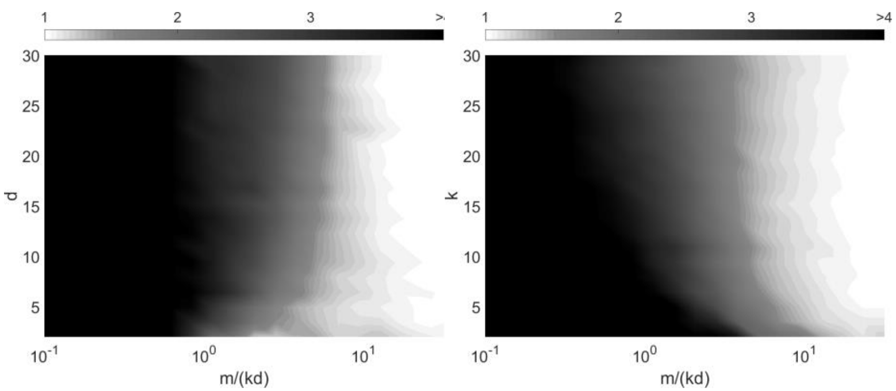
Stable distributions



Relative sketch size

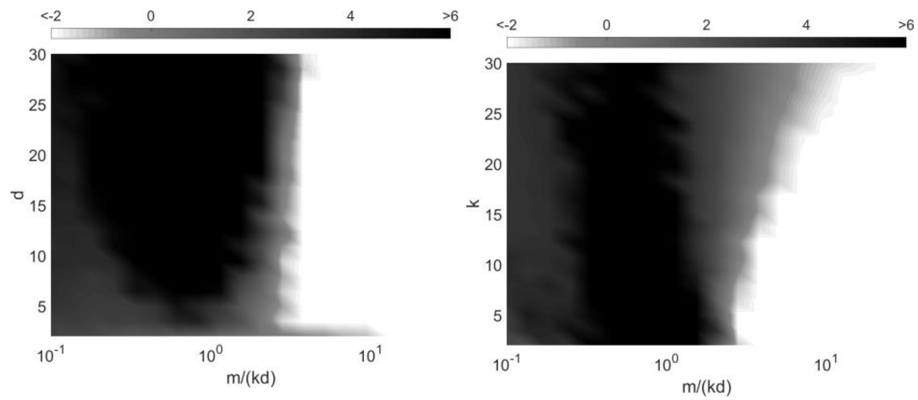
How big a sketch ?

k-means



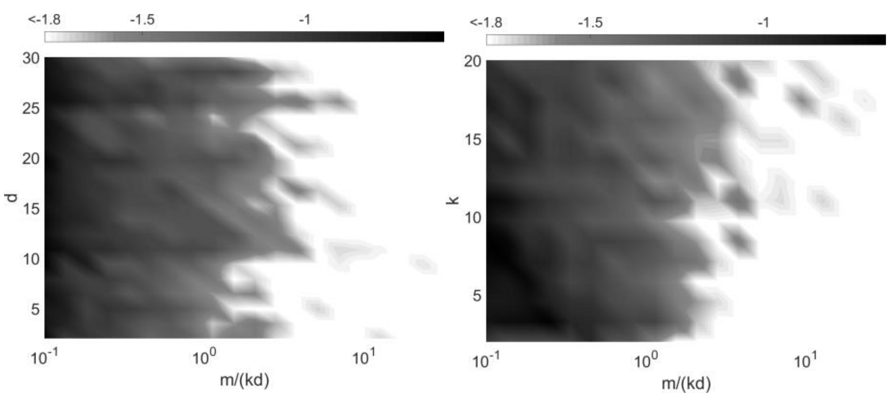
Relative sketch size m/(kd)

GMM



Relative sketch size

Stable distributions



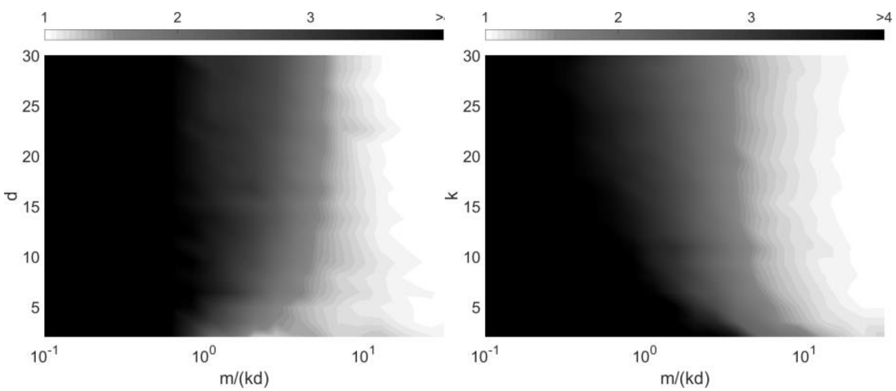
Relative sketch size

Sufficient sketch size

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

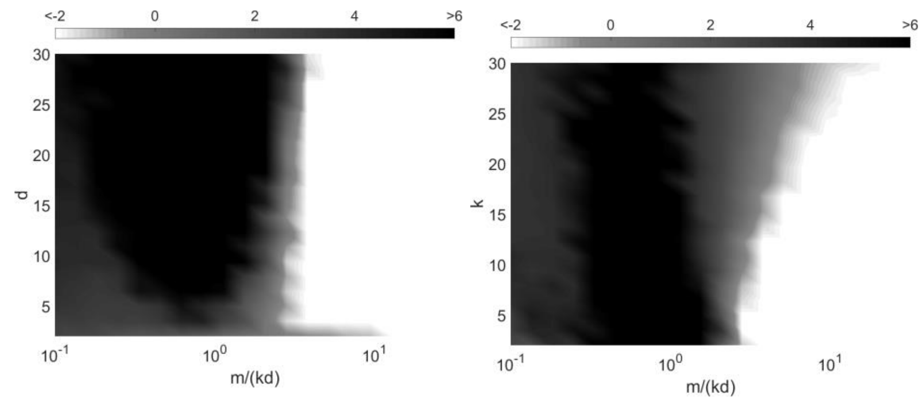
How big a sketch ?

k-means



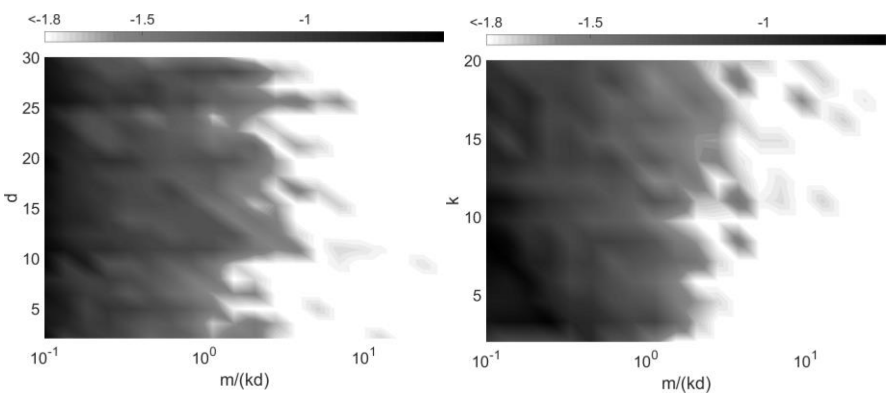
Relative sketch size m/(kd)

GMM



Relative sketch size

Stable distributions



Relative sketch size

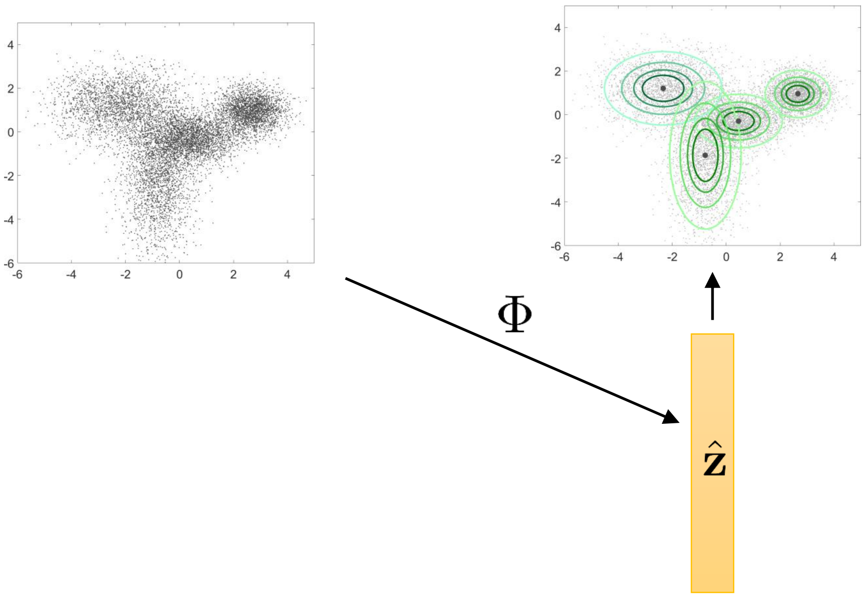
Sufficient sketch size

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

Can we characterize that?

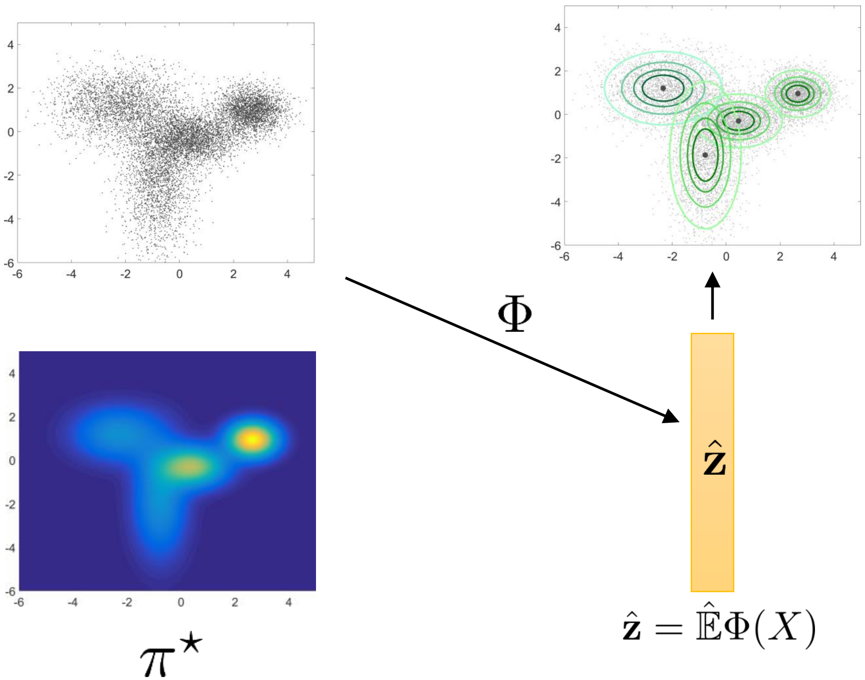
Outline

Linear inverse problem



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

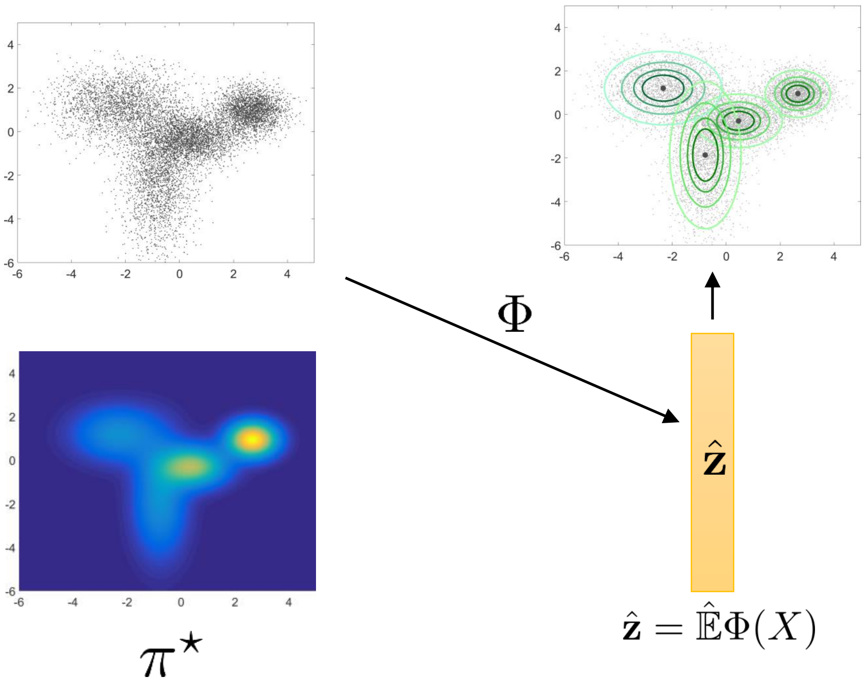
Linear inverse problem



Assumption on the data

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

Linear inverse problem

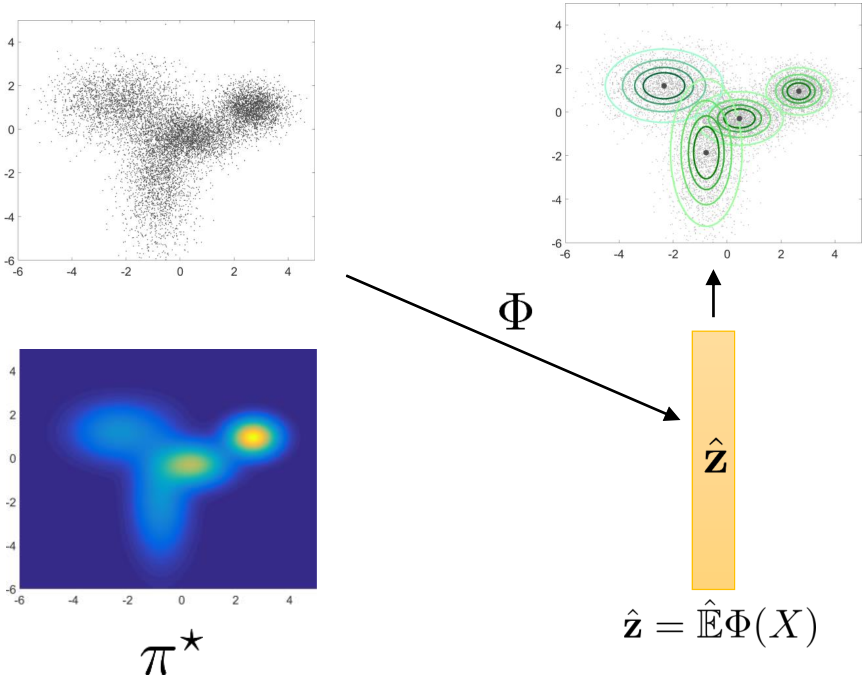


Assumption on the data

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

Reformulation of the sketching

Linear inverse problem



Assumption on the data

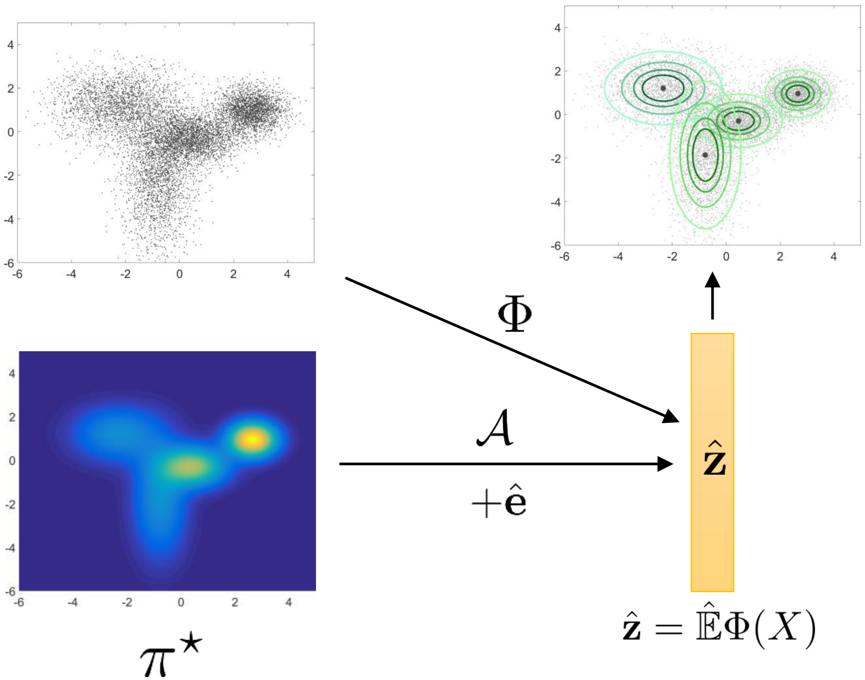
- True distribution: $x_1, \ldots, x_n \overset{i.i.d.}{\sim} \pi^\star$

Reformulation of the sketching

- Linear operator:

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

Linear inverse problem



Assumption on the data

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

Reformulation of the sketching

- Linear operator:

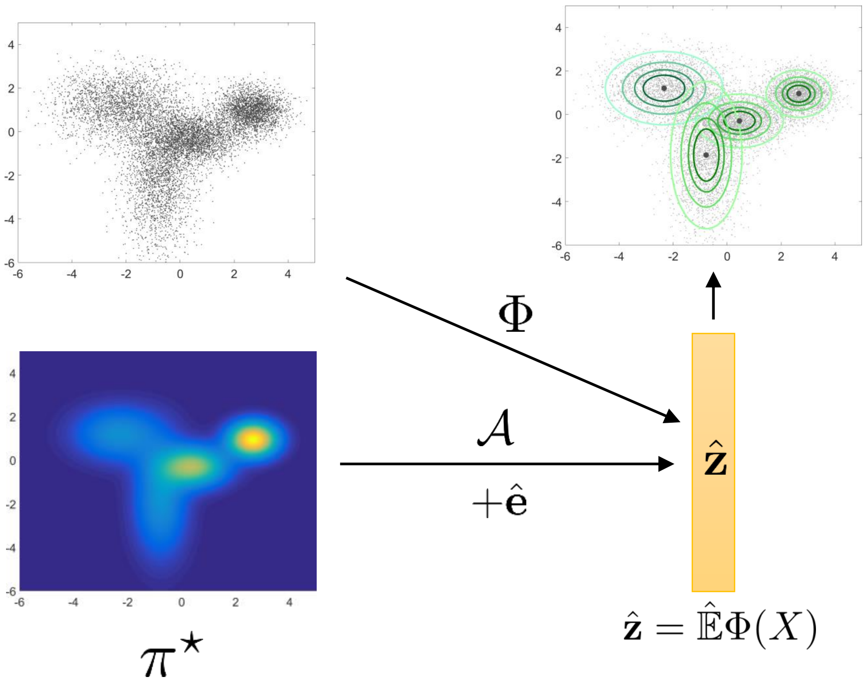
$$\boxed{\mathcal{A}\pi = \mathbb{E}_{X\sim\pi}\Phi(X)}$$

- « 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 *by Law of Large Numbers*

Linear inverse problem



- **Data = distribution**
- **Sketch = noisy linear measurement *of the distribution*** **(non-linear in data)**

Assumption on the data

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

Reformulation of the sketching

- Linear operator:

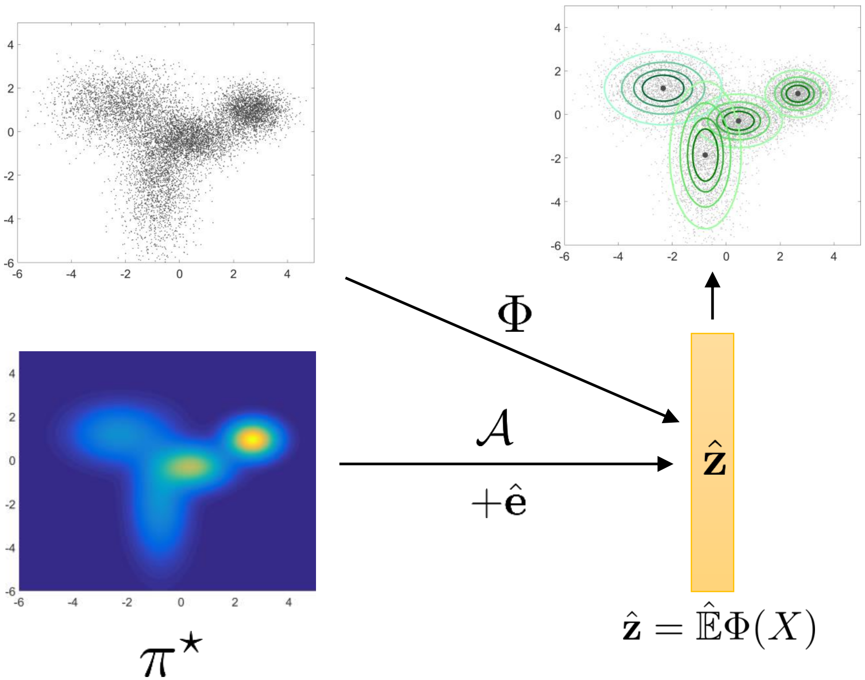
$$\mathcal{A}\pi = \mathbb{E}_{X\sim\pi}\Phi(X)$$

- « Noisy » linear measurement:

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

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

Linear inverse problem



- **Data = distribution**
- **Sketch = noisy linear measurement *of the distribution*** **(non-linear in data)**
- **Estimation problem = *linear inverse problem***

Assumption on the data

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

Reformulation of the sketching

- Linear operator:

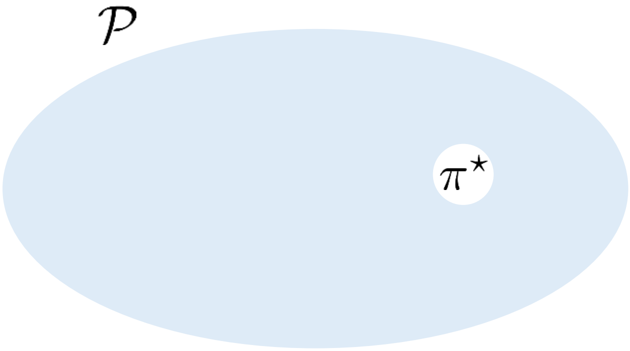
$$\boxed{\mathcal{A}\pi = \mathbb{E}_{X \sim \pi}\Phi(X)}$$

- « Noisy » linear measurement:

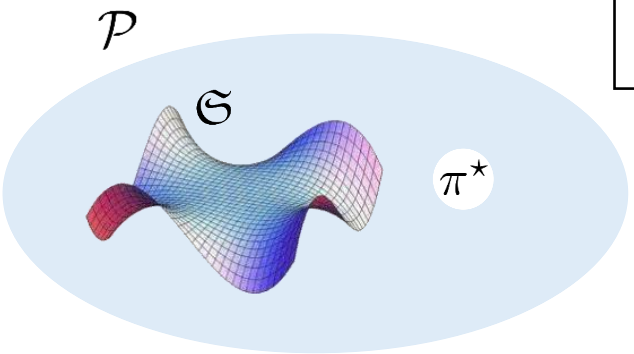
$$\boxed{\hat{\mathbf{z}} = \mathcal{A}\pi^\star + \hat{\mathbf{e}}}$$

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

Information preservation guarantees

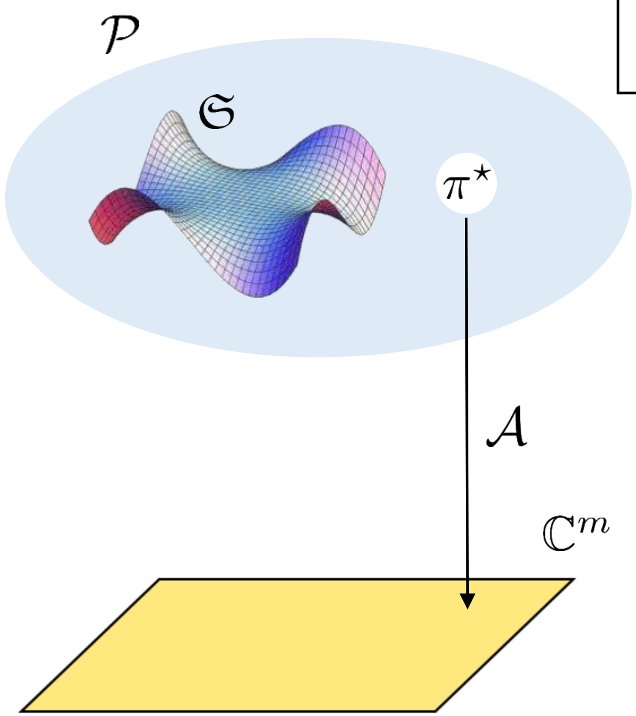


Information preservation guarantees



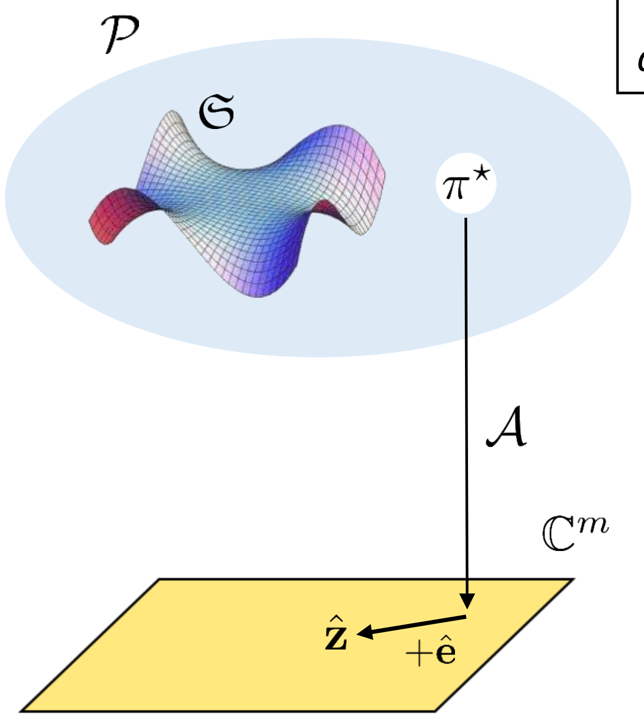
$\mathfrak{S}$: Model set of « simple » distributions (eg. GMMs)

Information preservation guarantees



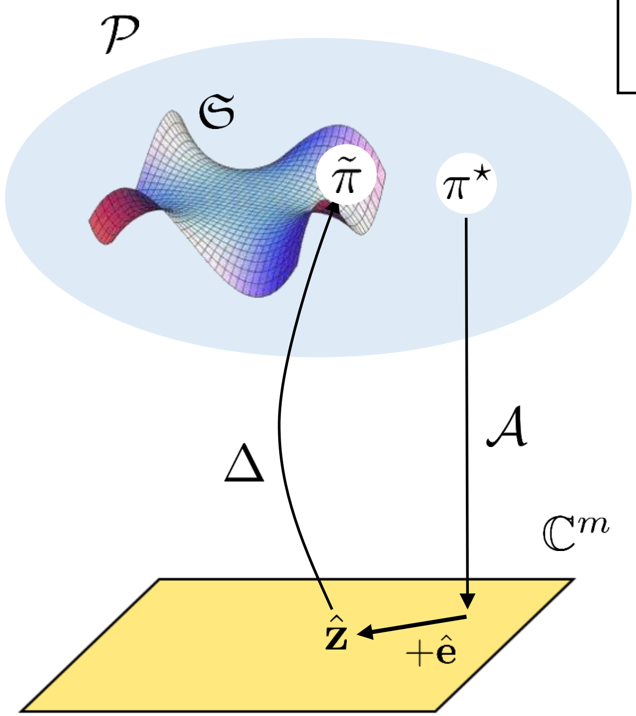
$\mathfrak{S}$: Model set of « simple » distributions (eg. GMMs)

Information preservation guarantees



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

Information preservation guarantees



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

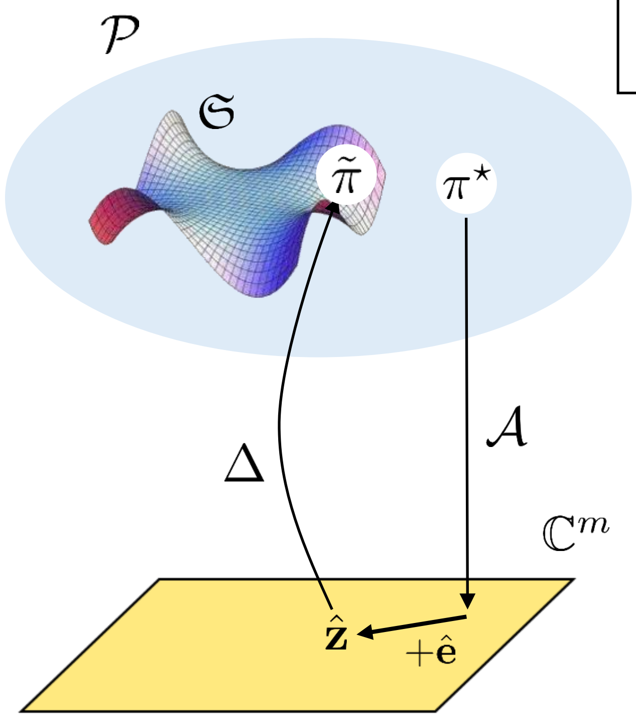
Goal

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

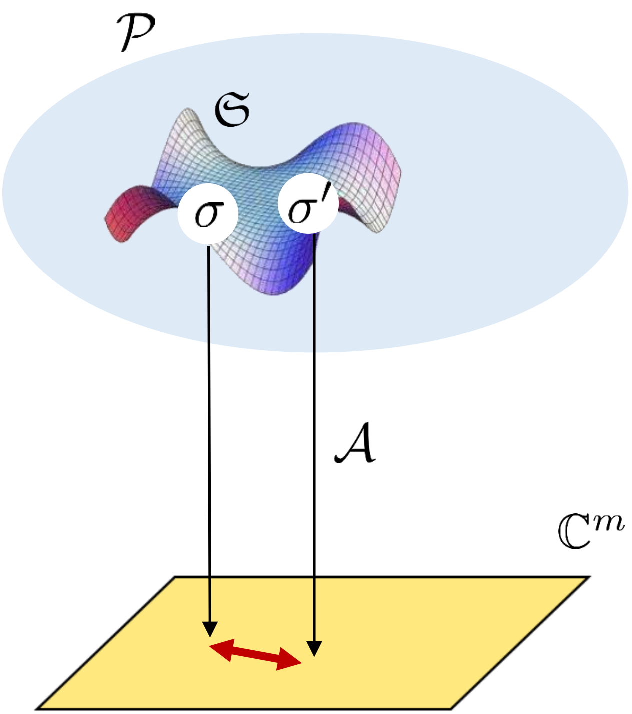
« Instance-optimal » decoder

Information preservation guarantees

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



$\Longrightarrow$



Goal

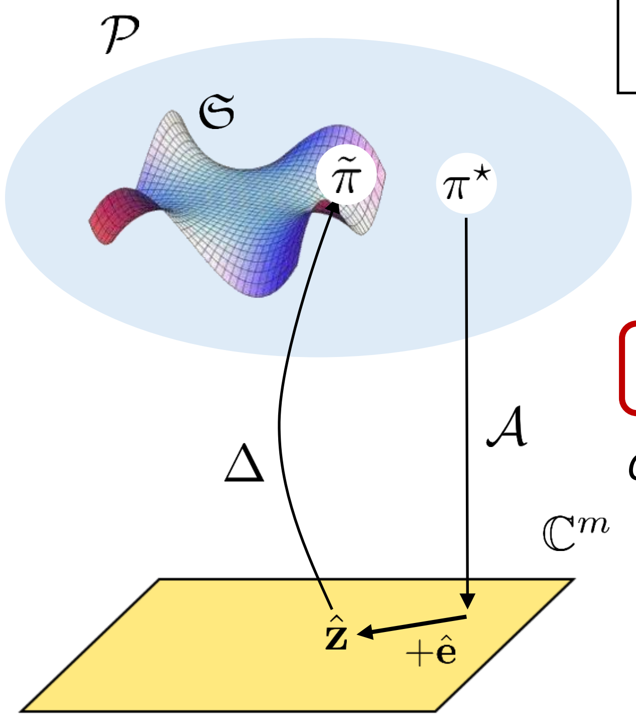
Prove the existence of a *decoder* Δ robust to noise and stable to modeling error.

« Instance-optimal » decoder

Lower Restricted Isometry Property

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

Information preservation guarantees



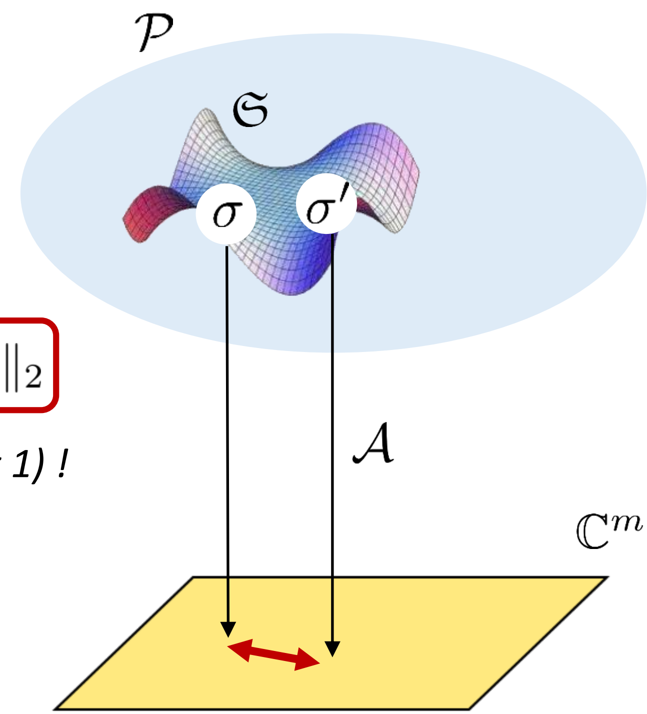
$\mathfrak{S}$: Model set of « simple » distributions (eg. GMMs)

$\Longrightarrow$

$\Longleftarrow$

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

Cost function used in practice (Part 1) !



Goal

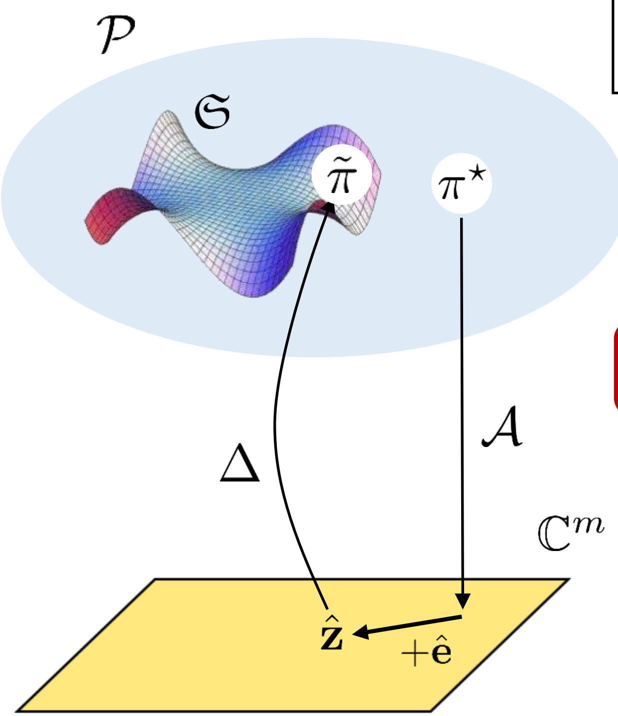
Prove the existence of a *decoder* Δ robust to noise and stable to modeling error.

« Instance-optimal » decoder

Lower Restricted Isometry Property

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

Information preservation guarantees



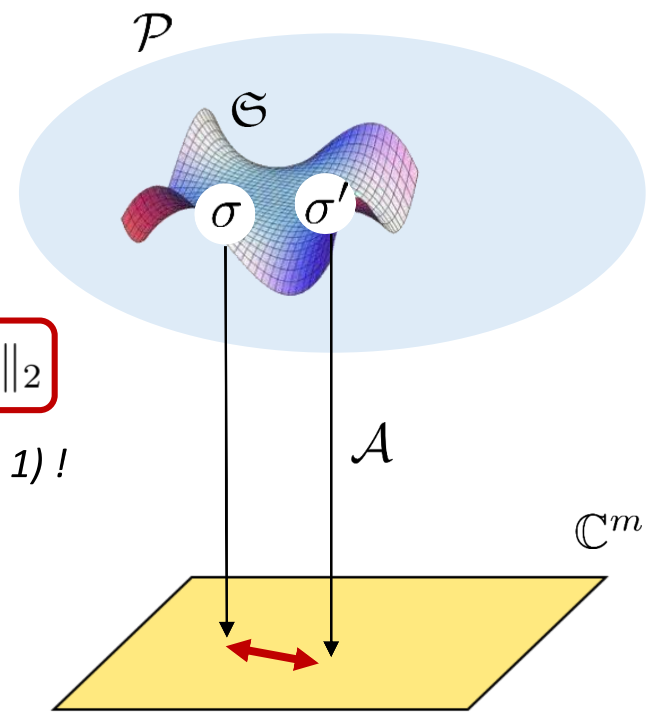
$\mathfrak{S}$: Model set of « simple » distributions (eg. GMMs)

$\Longrightarrow$

$\Longleftarrow$

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

Cost function used in practice (Part 1) !



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

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

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

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

Main result

Main hypothesis

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

Result

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

Quality of pointwise LRIP

Dimensionality of the model

W.h.p.

Modeling error

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

Main result

Main hypothesis

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

Result

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

Quality of pointwise LRIP

Dimensionality of the model

W.h.p.

Modeling error

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

- Classic CS: finite dimension: **Known**
- Here: infinite dimension: **Technical**

Main result

Main hypothesis

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

Result

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

W.h.p.

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

- Classic CS: finite dimension: **Known**
- Here: infinite dimension: **Technical**

Under **simplified hypothesis**:

$m \approx n$

(applied to mixture of stable dist.)

Outline

Compressive statistical learning

Key assumption for **mixture models:** ***separation of components***

Compressive statistical learning

Key assumption for **mixture models:** ***separation of components***

k-means with mixtures of Diracs

Compressive statistical learning

Key assumption for **mixture models:** ***separation of components***

k-means with mixtures of Diracs

Hypotheses

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

Compressive statistical learning

Key assumption for **mixture models:** ***separation of components***

k-means with mixtures of Diracs

Hypotheses

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

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

Compressive statistical learning

Key assumption for **mixture models:** ***separation of components***

k-means with mixtures of Diracs

Hypotheses

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

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

Sketch

- *Adjusted* Fourier features *(for technical reasons)*

Compressive statistical learning

Key assumption for **mixture models:** ***separation of components***

k-means with mixtures of Diracs

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

Hypotheses

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

Sketch

- *Adjusted* Fourier features *(for technical reasons)*

Result

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

Compressive statistical learning

Key assumption for **mixture models: *separation of components***

k-means with mixtures of Diracs

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

Hypotheses

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

Sketch

- *Adjusted* Fourier features *(for technical reasons)*

Result

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

Sketch size

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

Compressive statistical learning

Key assumption for **mixture models: *separation of components***

k-means with mixtures of Diracs

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

Hypotheses

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

Sketch

- *Adjusted* Fourier features *(for technical reasons)*

Result

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

Sketch size

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

GMM with known covariance

Compressive statistical learning

Key assumption for **mixture models:** ***separation of components***

k-means with mixtures of Diracs

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

Hypotheses

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

Sketch

- *Adjusted* Fourier features *(for technical reasons)*

Result

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

Sketch size

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

GMM with known covariance

Hypotheses

- Sufficiently separated means
- Bounded domain for means

Compressive statistical learning

Key assumption for **mixture models:** ***separation of components***

k-means with mixtures of Diracs

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

Hypotheses

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

Sketch

- *Adjusted* Fourier features *(for technical reasons)*

Result

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

Sketch size

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

GMM with known covariance

Hypotheses

- Sufficiently separated means
- Bounded domain for means

Sketch

- Fourier features

Compressive statistical learning

Key assumption for **mixture models: *separation of components***

k-means with mixtures of Diracs

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

Hypotheses

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

Sketch

- *Adjusted* Fourier features *(for technical reasons)*

Result

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

Sketch size

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

GMM with known covariance

Hypotheses

- Sufficiently separated means
- Bounded domain for means

Sketch

- Fourier features

Result

- With respect to **log-likelihood**

Compressive statistical learning

Key assumption for **mixture models:** ***separation of components***

k-means with mixtures of Diracs

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

Hypotheses

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

Sketch

- *Adjusted* Fourier features *(for technical reasons)*

Result

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

Sketch size

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

GMM with known covariance

Hypotheses

- Sufficiently separated means
- Bounded domain for means

Sketch

- Fourier features

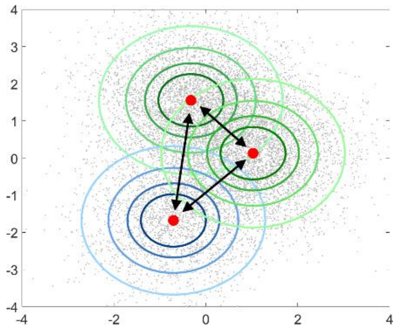
Result

- With respect to **log-likelihood**

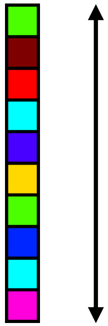
Sketch size

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

GMM trade-off



Trade-off



Separation of means

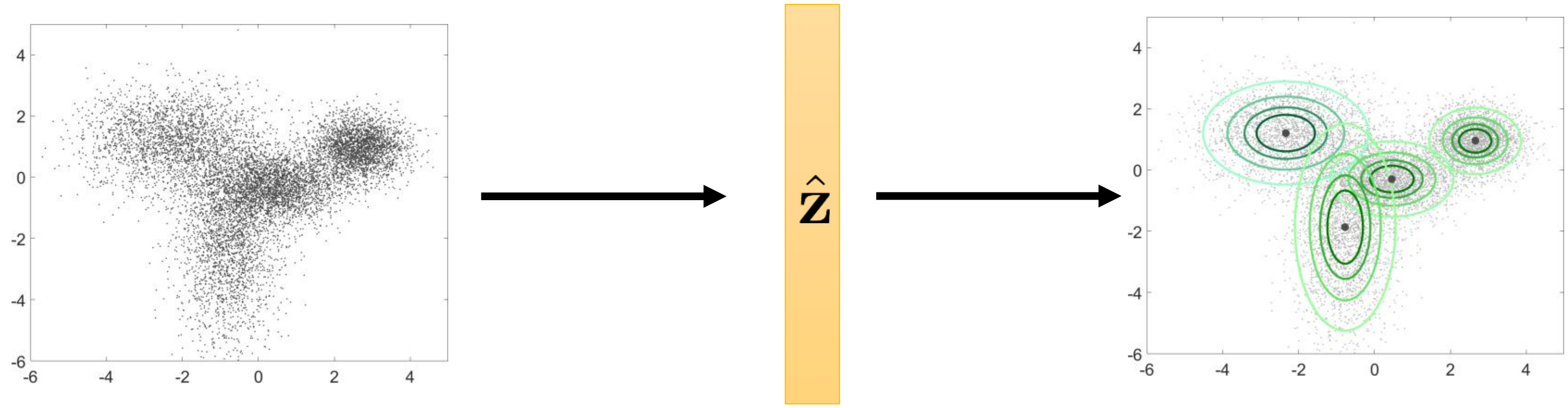
Size of sketch

More
High
Freq.

Separation of means	Number of measurements
$\mathcal{O}\left(\sqrt{d \log k}\right)$	$m \geq \mathcal{O}\left(k^2 d^2 \cdot \mathtt{polylog}(k,d)\right)$
$\mathcal{O}\left(\sqrt{d + \log k}\right)$	$m \geq \mathcal{O}\left(k^3 d^2 \cdot \mathtt{polylog}(k,d)\right)$
$\mathcal{O}\left(\sqrt{\log k}\right)$	$m \geq \mathcal{O}\left(k^2 d^2 e^d \cdot \mathtt{polylog}(k,d)\right)$

Outline

Sketch learning



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

Summary of contributions

- Practical illustration: **flexible greedy algorithm for any sketched mixture model estimation**

Summary of contributions

- Practical illustration: **flexible greedy algorithm for any 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 greedy algorithm for any sketched mixture model estimation**
  - GMM with diagonal covariance
  - k-means (mixture of Diracs)
  - *Mixture of multivariate elliptic stable distributions*
- Validation on real and synthetic data

Summary of contributions

- Practical illustration: **flexible greedy algorithm for any sketched mixture model estimation**
  - GMM with diagonal covariance
  - k-means (mixture of Diracs)
  - *Mixture of multivariate elliptic stable distributions*
- Validation on real and synthetic data

- Information-preservation guarantees for **sketched density estimation**

Summary of contributions

- Practical illustration: **flexible greedy algorithm for any sketched mixture model estimation**
  - GMM with diagonal covariance
  - k-means (mixture of Diracs)
  - *Mixture of multivariate elliptic stable distributions*
- Validation on real and synthetic data

- Information-preservation guarantees for **sketched density estimation**
  - Infinite dimensional **Compressive Sensing** (Restricted isometry property)
  - **Kernel methods** on distributions (Kernel mean, Random features)

Summary of contributions

- Practical illustration: **flexible greedy algorithm for any sketched mixture model estimation**
  - GMM with diagonal covariance
  - k-means (mixture of Diracs)
  - *Mixture of multivariate elliptic stable distributions*
- Validation on real and synthetic data

- Information-preservation guarantees for **sketched density estimation**
  - 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 greedy algorithm for any sketched mixture model estimation**
  - GMM with diagonal covariance
  - k-means (mixture of Diracs)
  - *Mixture of multivariate elliptic stable distributions*
- Validation on real and synthetic data

- Information-preservation guarantees for **sketched density estimation**
  - Infinite dimensional **Compressive Sensing** (Restricted isometry property)
  - **Kernel methods** on distributions (Kernel mean, Random features)
- Generic assumptions of *low-dimensionality* of the model set
- Focus on mixture models
  - Estimator of mixture of multivariate elliptic stable distributions
  - Statistical learning with controlled sketch size for k-means, sketched GMM with known covariance

Outlooks : sketch

- Obtain algorithmic guarantees?

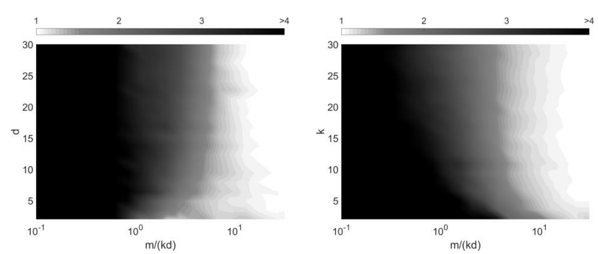
Outlooks : sketch

- Obtain algorithmic guarantees?
  - Similar algorithms can be found in e.g. **super-resolution** with other interpretations (Frank-Wolfe, conditional gradient…) *[eg Bredies 2012…]*
  - Convergence guarantees as $k \to \infty$, no guarantees for exactly k-sparse measures…

Outlooks : sketch

- Obtain algorithmic guarantees?
  - Similar algorithms can be found in e.g. **super-resolution** with other interpretations (Frank-Wolfe, conditional gradient…) *[eg Bredies 2012…]*
  - Convergence guarantees as $k \to \infty$, no guarantees for exactly k-sparse measures…

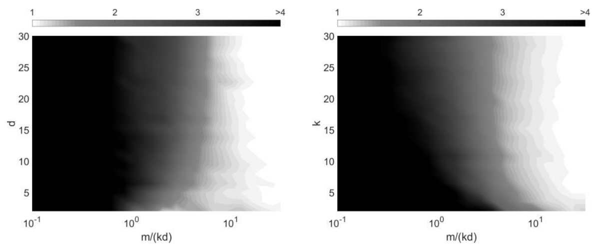
- Bridge observed gap between theory and practice ?



Outlooks : sketch

- Obtain algorithmic guarantees?
  - Similar algorithms can be found in e.g. **super-resolution** with other interpretations (Frank-Wolfe, conditional gradient…) *[eg Bredies 2012…]*
  - Convergence guarantees as $k \to \infty$, no guarantees for exactly k-sparse measures…

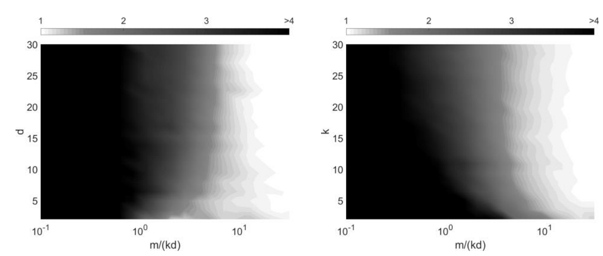
- Bridge observed gap between theory and practice ?
  - Does *not* come from coverings numbers
  - Improve pointwise concentration?



Outlooks : sketch

- Obtain algorithmic guarantees?
  - Similar algorithms can be found in e.g. **super-resolution** with other interpretations (Frank-Wolfe, conditional gradient…) *[eg Bredies 2012…]*
  - Convergence guarantees as $k \to \infty$, no guarantees for exactly k-sparse measures…

- Bridge observed gap between theory and practice ?
  - Does *not* come from coverings numbers
  - Improve pointwise concentration?
  - **Recent result:** $k^2 d^2 \to k^3 d$

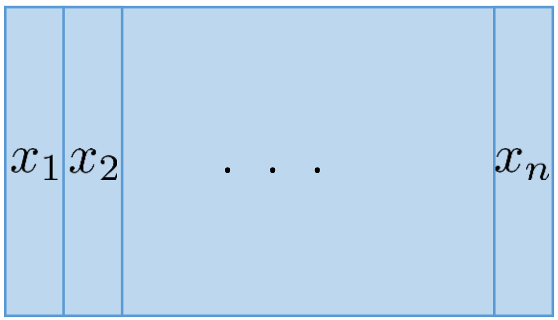


Outlooks : beyond sketches

- Combine with **dimension reduction** for **HD** data?
  - First map in low-d, then sketch

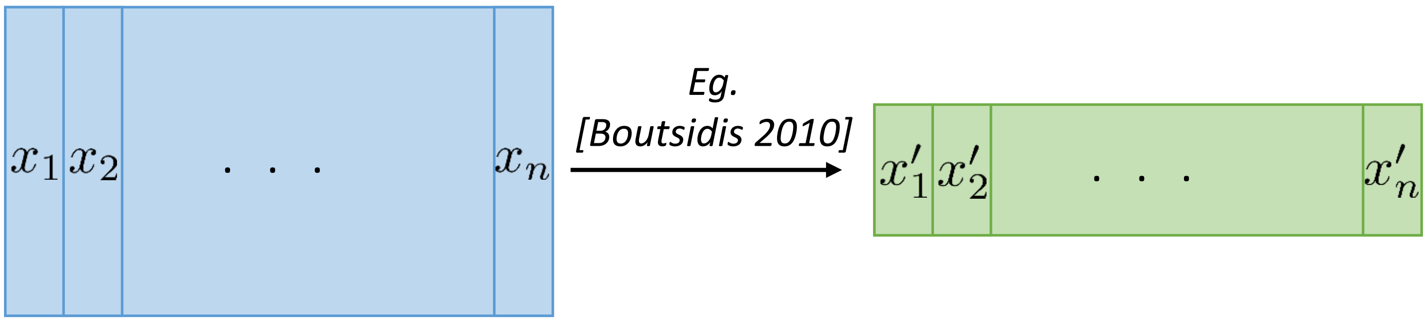
Outlooks : beyond sketches

- Combine with **dimension reduction** for **HD** data?
  - First map in low-d, then sketch



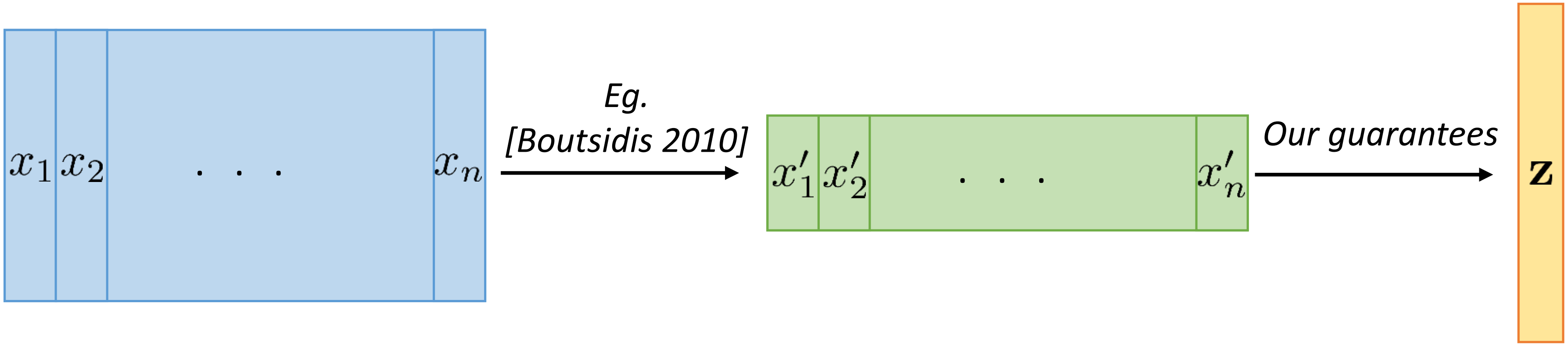
Outlooks : beyond sketches

- Combine with **dimension reduction** for **HD** data?
  - First map in low-d, then sketch



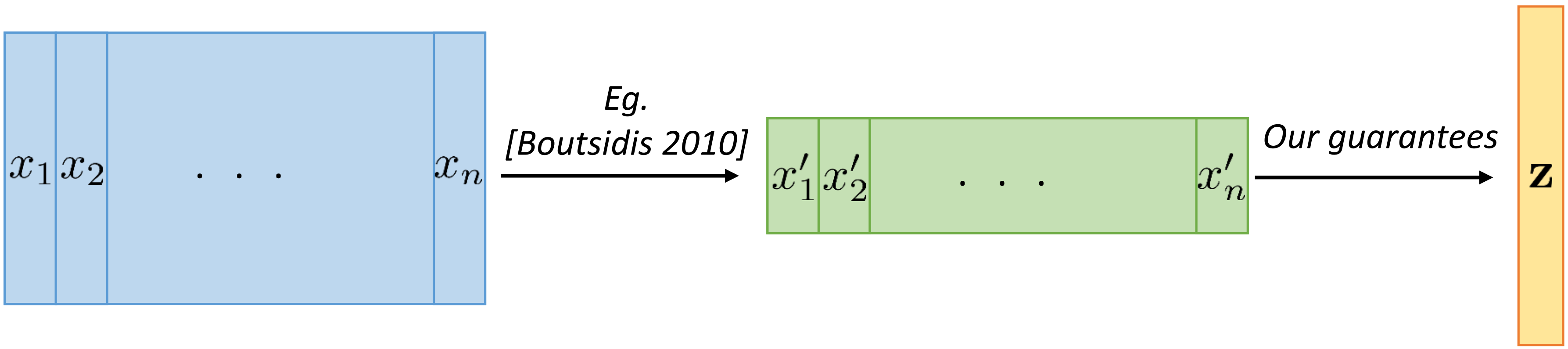
Outlooks : beyond sketches

- Combine with **dimension reduction** for **HD** data?
  - First map in low-d, then sketch



Outlooks : beyond sketches

- Combine with **dimension reduction** for **HD** data?
  - First map in low-d, then sketch

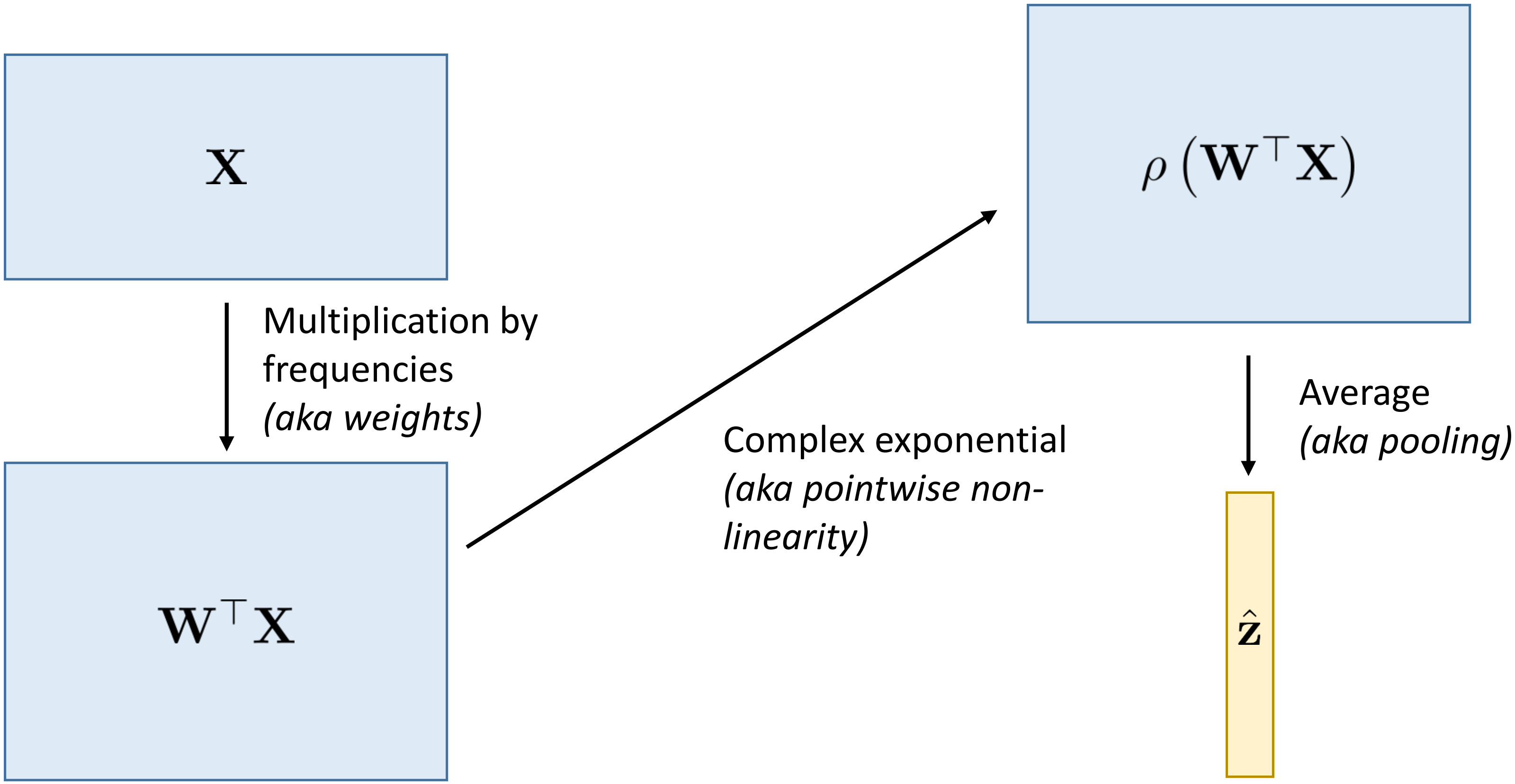


- Extend framework to other tasks?
  - « Sketchify » other kernel methods?

Oliva2016

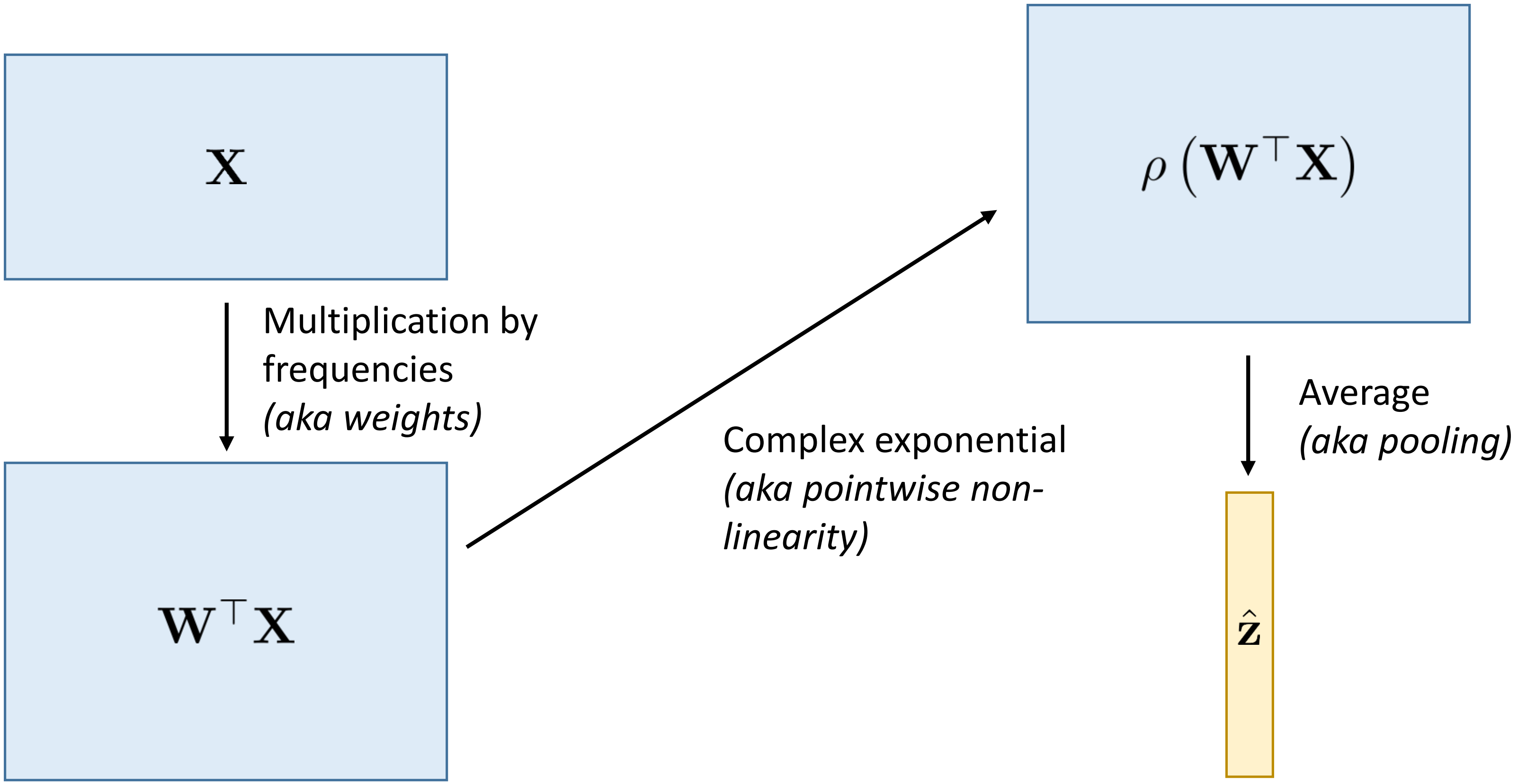
Outlooks : beyond sketches

- Extension to multi-layer sketches ? (Neural networks…)



Outlooks : beyond sketches

- Extension to multi-layer sketches ? (Neural networks…)
  - Equivalence between LRIP and instance optimality still valid for **non-linear operators** !



Thank you !