Sketching for Large-Scale Learning of Mixture Models

Nicolas Keriven

Université Rennes 1, Inria Rennes Bretagne-atlantique

Adv. Rémi Gribonval

Outline

1 **Introduction**

2 Practical Approach

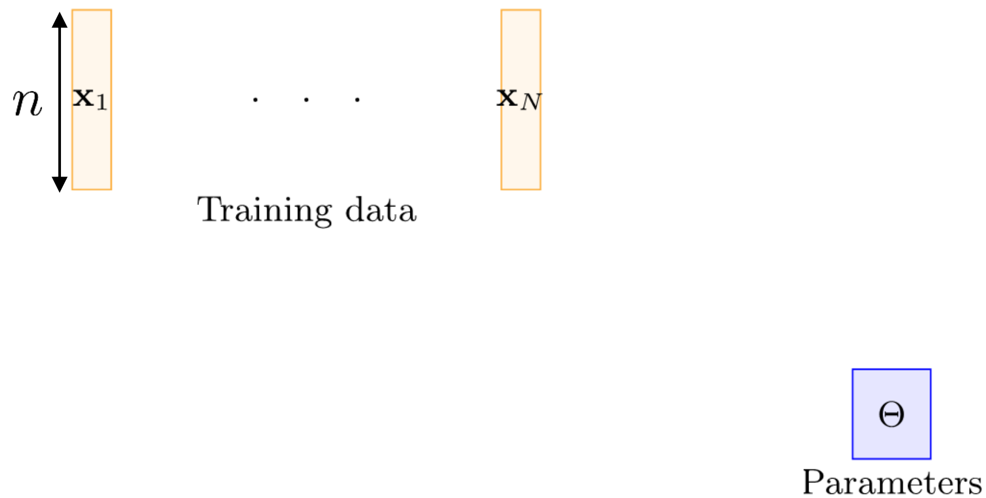
3 Results

4 Theoretical analysis

5 Conclusion and outlooks

Paths to Compressive Learning

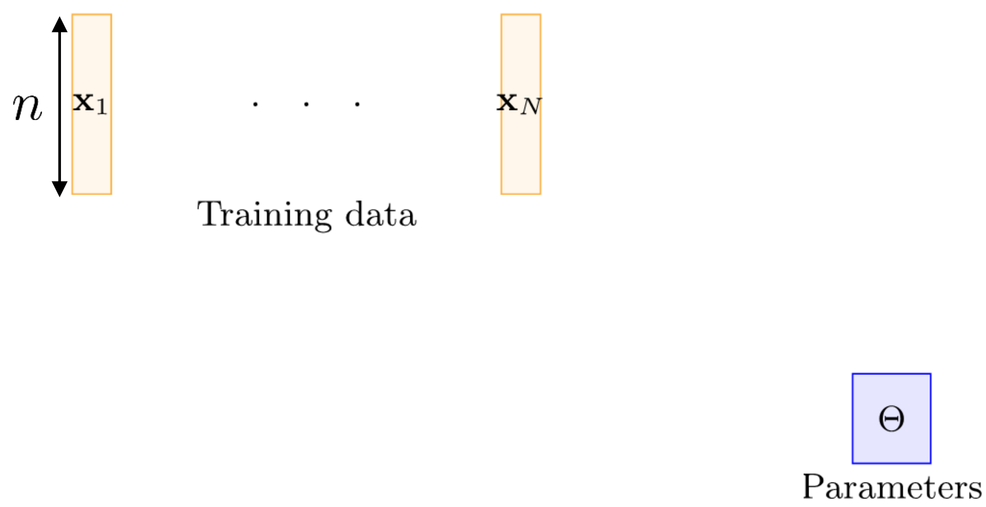
Goal : Compute parameters Θ from a **large** database.



Paths to Compressive Learning

Goal : Compute parameters Θ from a **large** database.

- PCA : $\mathbf{x} \in Span(\theta_1, ..., \theta_k)$
- Classification : $< w_\Theta, \Phi(\mathbf{x}) >$
- Regression : $\mathbf{y} = f_\Theta(\mathbf{x})$
- Density estimation : $\mathbf{x} \sim p_\Theta$

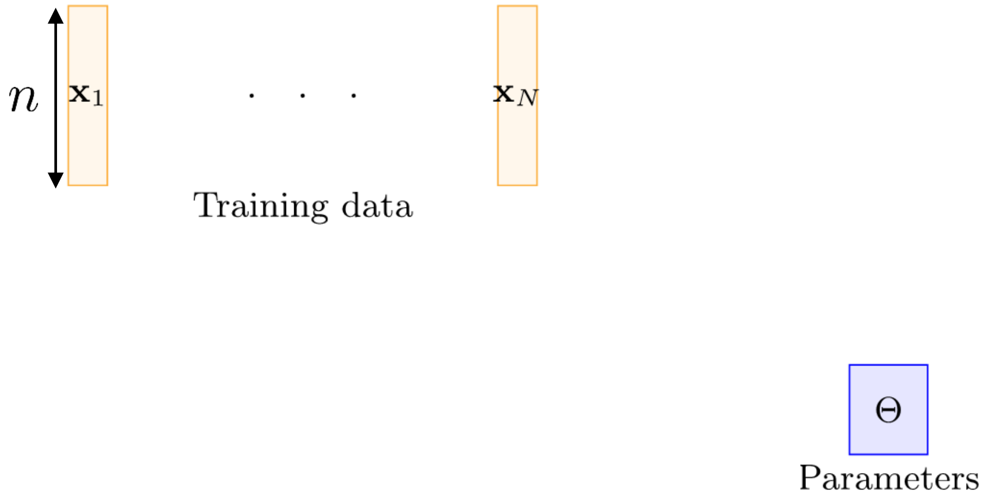


Paths to Compressive Learning

Goal : Compute parameters Θ from a **large** database.

- PCA : $\mathbf{x} \in Span(\theta_1, ..., \theta_k)$
- Classification : $< w_\Theta, \Phi(\mathbf{x}) >$
- Regression : $\mathbf{y} = f_\Theta(\mathbf{x})$
- Density estimation : $\mathbf{x} \sim p_\Theta$

Idea : compress the database beforehand.



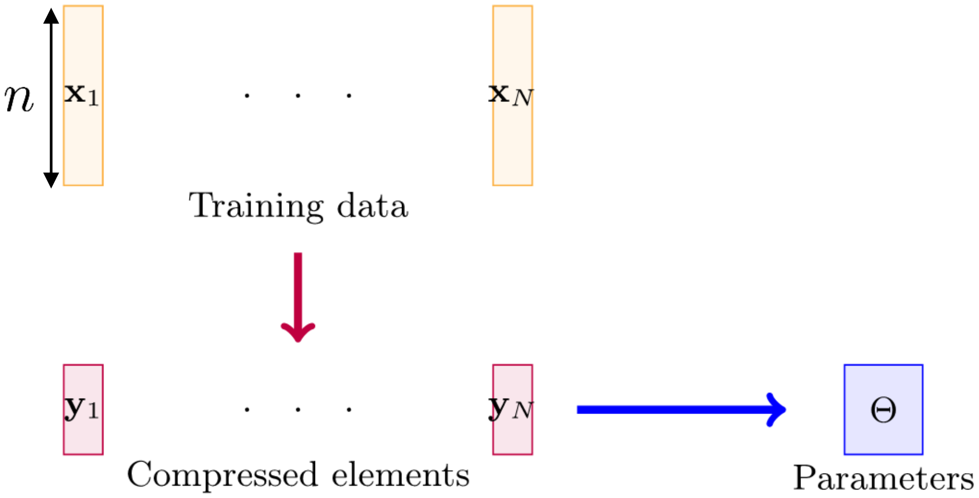
Paths to Compressive Learning

Goal : Compute parameters Θ from a **large** database.

- PCA : $\mathbf{x} \in Span(\theta_1, ..., \theta_k)$
- Classification : $< w_\Theta, \Phi(\mathbf{x}) >$
- Regression : $\mathbf{y} = f_\Theta(\mathbf{x})$
- Density estimation : $\mathbf{x} \sim p_\Theta$

Idea : compress the database beforehand.

- Large n (tall)
  - See e.g. [Calderbank 2009]

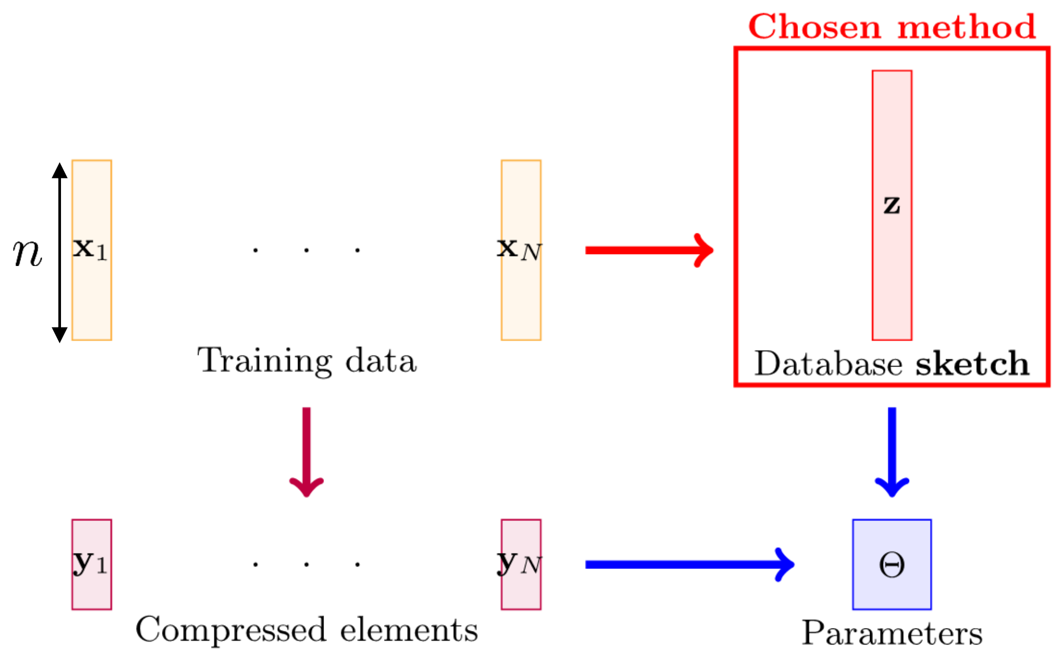


Paths to Compressive Learning

Goal : Compute parameters Θ from a **large** database.

- PCA : $\mathbf{x} \in Span(\theta_1, ..., \theta_k)$
- Classification : $< w_\Theta, \Phi(\mathbf{x}) >$
- Regression : $\mathbf{y} = f_\Theta(\mathbf{x})$
- Density estimation : $\mathbf{x} \sim p_\Theta$

Idea : compress the database beforehand.



- Large n (tall)
  - See e.g. [Calderbank 2009]
- Large N (fat) **« Big data »**
  - See e.g. [Cormode 2011]

Inspiration

- **Sketch…**

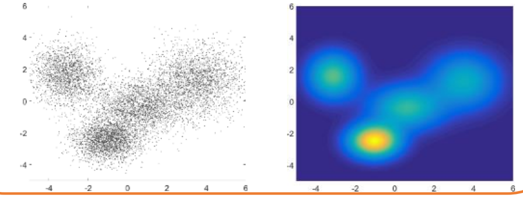
- Contains particular info about the database
- Maintained **online**

  [Cormode 2011]

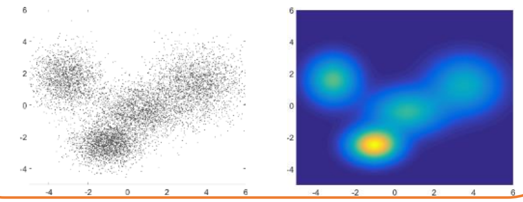
Inspiration

- **Sketch…**
  - Contains particular info about the database
  - Maintained **online**
    [Cormode 2011]
- **…Learning…**

Knowledge about **underlying probability distribution**

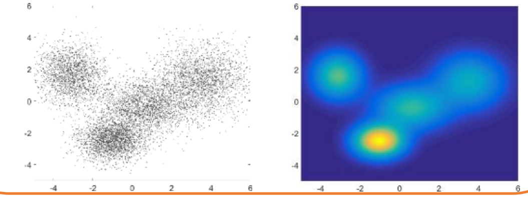


Inspiration

- **Sketch…**
  - Contains particular info about the database
  - Maintained **online**
    [Cormode 2011]
- **…Learning…**
  Knowledge about **underlying probability distribution**
  
- **…by Compressive Sensing**
  Recover « low-dimensional » object from few linear measurements (ex : sparse vector, low-rank matrix…)

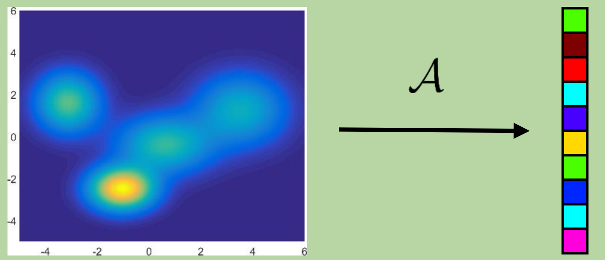
Inspiration

- **Sketch…**
  - Contains particular info about the database
  - Maintained **online** *[Cormode 2011]*
- **…Learning…**
  - Knowledge about **underlying probability distribution**



- **…by Compressive Sensing**
  - Recover « low-dimensional » object from few linear measurements (ex : sparse vector, low-rank matrix…)

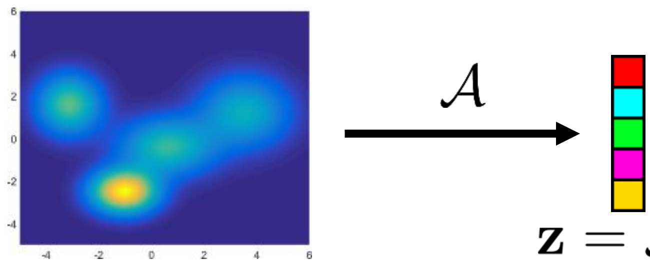
Sketch = measurements of underlying probability distribution



$p \in \mathcal{P}$ $\qquad$ $\mathbf{z} = \mathcal{A}p \in \mathbb{C}^m$

Generalized Moments

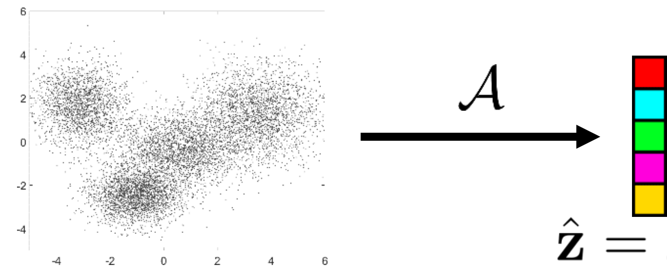
(Generalized) Compressive Sensing



$\mathbf{z} = \mathcal{A}p$

p

Practical Compressive Learning

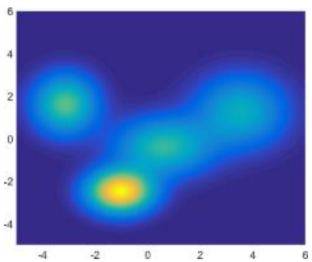


$\hat{\mathbf{z}} = \mathcal{A}\hat{p}$

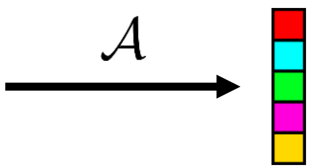
$\hat{p} = (\sum_i \delta_{\mathbf{x}_i})/N$

Generalized Moments

(Generalized) Compressive Sensing

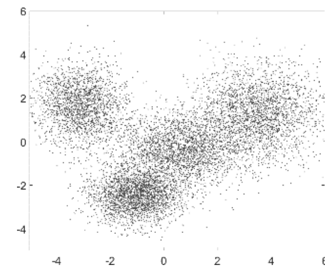


p

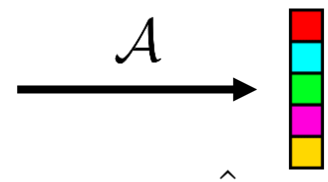


$\mathbf{z} = \mathcal{A}p$

Practical Compressive Learning



$\hat{p} = (\sum_i \delta_{\mathbf{x}_i})/N$

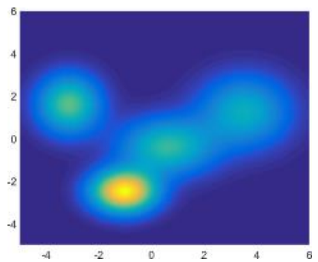


$\hat{\mathbf{z}} = \mathcal{A}\hat{p}$

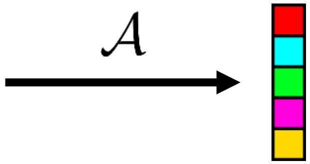
$$\mathbf{z} = \mathcal{A}p = \left[\mathbb{E}_{\mathbf{x}\sim p}\phi_j(\mathbf{x})\right]_{j=1}^{m} \quad \approx \quad \hat{\mathbf{z}} = \mathcal{A}\hat{p} = \left[\frac{1}{N}\sum_{i=1}^{N}\phi_j(\mathbf{x}_i)\right]_{j=1}^{m}$$

Generalized Moments

(Generalized) Compressive Sensing



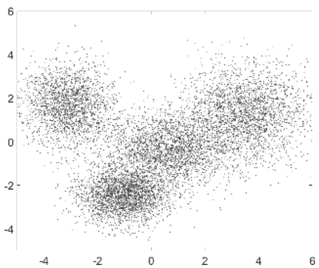
$\mathcal{A}$



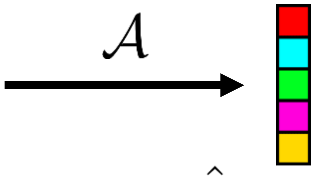
$\mathbf{z} = \mathcal{A}p$

p

Practical Compressive Learning



$\mathcal{A}$



$\hat{\mathbf{z}} = \mathcal{A}\hat{p}$

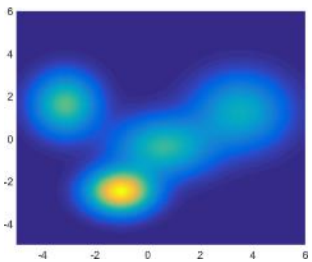
$\hat{p} = (\sum_i \delta_{\mathbf{x}_i})/N$

$$\mathbf{z} = \mathcal{A}p = \left[\mathbb{E}_{\mathbf{x}\sim p}\phi_j(\mathbf{x})\right]_{j=1}^{m} \quad \approx \quad \hat{\mathbf{z}} = \mathcal{A}\hat{p} = \left[\frac{1}{N}\sum_{i=1}^{N}\phi_j(\mathbf{x}_i)\right]_{j=1}^{m}$$

Compressive Sensing :
(Random) Projections

Generalized Moments

(Generalized) Compressive Sensing

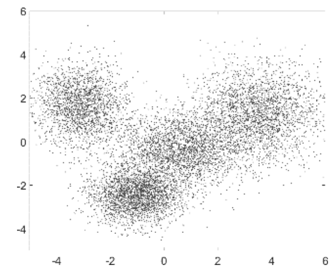


$$\xrightarrow{\mathcal{A}}$$

p

$\mathbf{z} = \mathcal{A}p$

Practical Compressive Learning



$$\xrightarrow{\mathcal{A}}$$

$\hat{p} = \left(\sum_i \delta_{\mathbf{x}_i}\right)/N$

$\hat{\mathbf{z}} = \mathcal{A}\hat{p}$

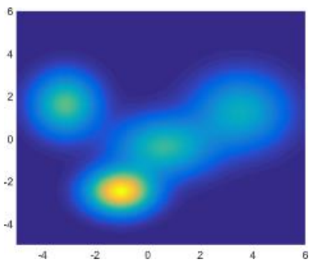
$$\mathbf{z} = \mathcal{A}p = \left[\mathbb{E}_{\mathbf{x}\sim p}\phi_j(\mathbf{x})\right]_{j=1}^{m} \quad \approx \quad \hat{\mathbf{z}} = \mathcal{A}\hat{p} = \left[\tfrac{1}{N}\sum_{i=1}^{N}\phi_j(\mathbf{x}_i)\right]_{j=1}^{m}$$

Compressive Sensing :
(Random) Projections

- **Online** ✓
- Distributed ✓

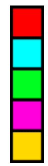
Generalized Moments

(Generalized) Compressive Sensing



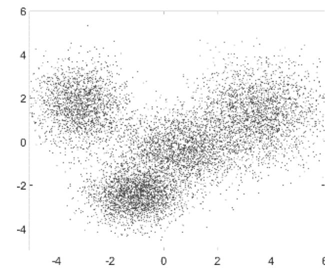
p

$$\xrightarrow{\mathcal{A}}$$



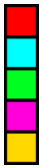
$\mathbf{z} = \mathcal{A}p$

Practical Compressive Learning



$\hat{p} = (\sum_i \delta_{\mathbf{x}_i})/N$

$$\xrightarrow{\mathcal{A}}$$



$\hat{\mathbf{z}} = \mathcal{A}\hat{p}$

$$\mathbf{z} = \mathcal{A}p = \left[\mathbb{E}_{\mathbf{x}\sim p}\phi_j(\mathbf{x})\right]_{j=1}^{m} \approx \hat{\mathbf{z}} = \mathcal{A}\hat{p} = \left[\frac{1}{N}\sum_{i=1}^{N}\phi_j(\mathbf{x}_i)\right]_{j=1}^{m}$$

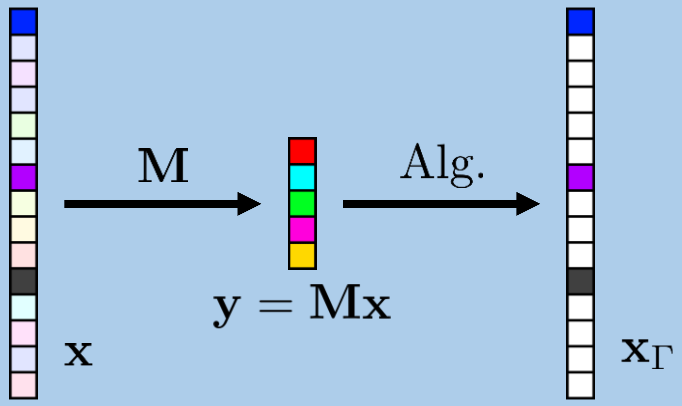
Compressive Sensing :
(Random) Projections

Robustness of learning Alg. ?

- **Online** ✓
- Distributed ✓

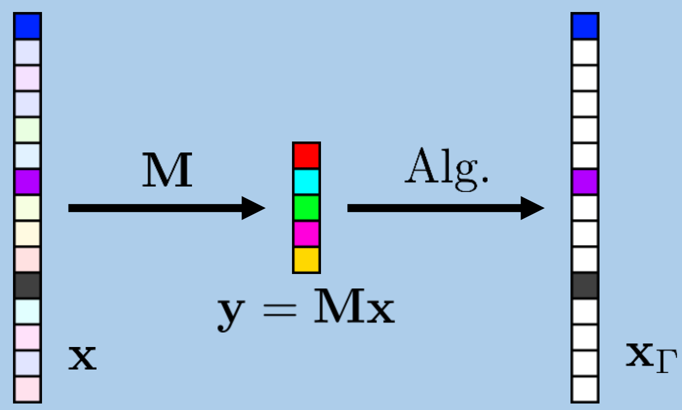
Mixture Model Estimation

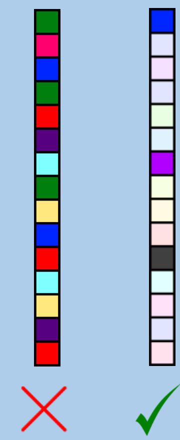
Usual Compressive Sensing



Mixture Model Estimation

Usual Compressive Sensing





Sparsity

$$\mathbf{x} \approx \mathbf{x}_\Gamma = \sum_{i \in \Gamma} x_i \mathbf{e_i}$$

Mixture Model Estimation

Usual Compressive Sensing

$\mathbf{x}$

$\xrightarrow{\mathbf{M}}$

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

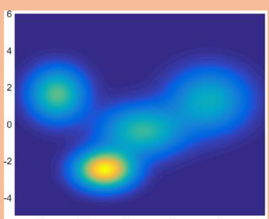
$\xrightarrow{\text{Alg.}}$

$\mathbf{x}_\Gamma$

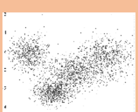
Sparsity

$\mathbf{x} \approx \mathbf{x}_\Gamma = \sum_{i \in \Gamma} x_i \mathbf{e_i}$

Generalized Compressive Sensing



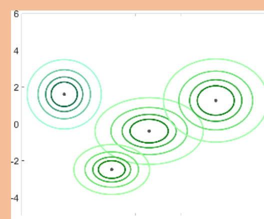
p

(or )

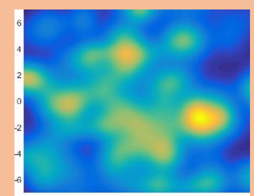
$\xrightarrow{\mathcal{A}}$

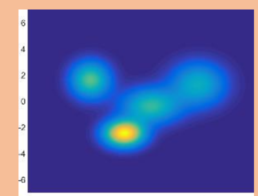
$\mathbf{z} = \mathcal{A}p$

$\xrightarrow{\text{Alg.}}$



$p_{\Theta,\alpha}$





$p \approx p_{\Theta,\alpha} = \sum_k \alpha_k p_{\theta_k}$

Infinite, continuous dictionary !

Outline

- Practical Approach (Section 2 & 3)
  - Greedy algorithm inspired by Compressive Sensing
  - Application to K-means, GMM with diagonal covariance

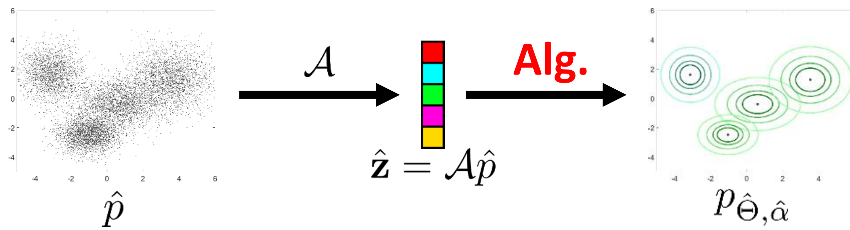
Outline

- Practical Approach (Section 2 & 3)
  - Greedy algorithm inspired by Compressive Sensing
  - Application to K-means, GMM with diagonal covariance

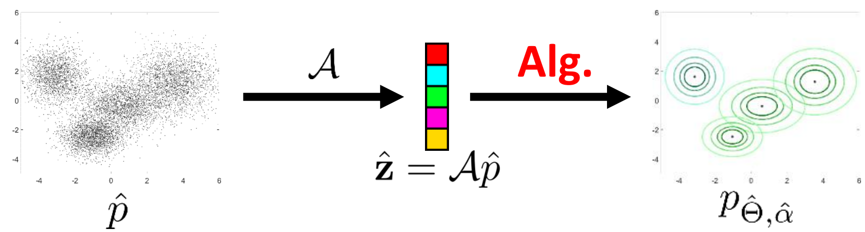
- Theoretical Analysis (Section 4)
  - Information-preservation guarantee
  - Infinite-dimensional Compressive Sensing

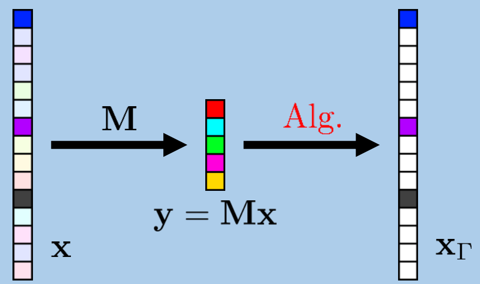
Outline

Cost function



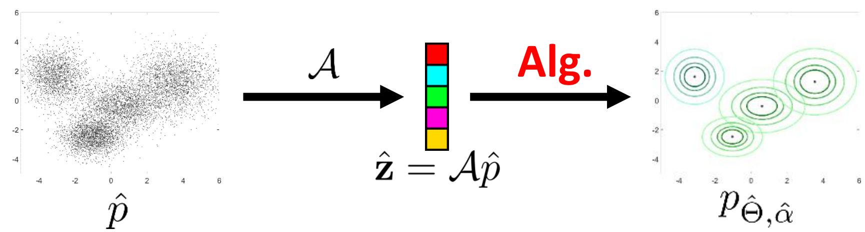
Cost function

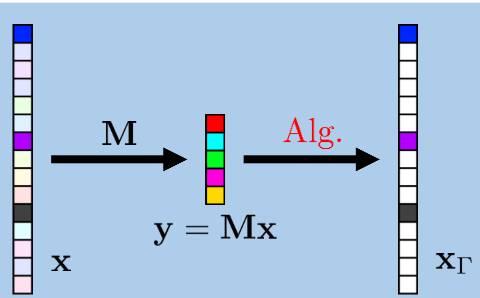




$$\boxed{\min_{\|\mathbf{x}\|_0 \leq s} \|\mathbf{y} - \mathbf{M}\mathbf{x}\|_2}$$

Cost function



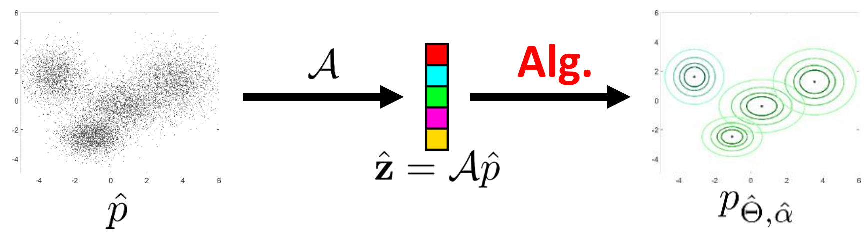


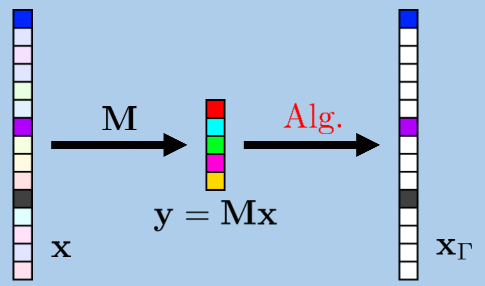
$$\min_{\|\mathbf{x}\|_0 \leq s} \|\mathbf{y} - \mathbf{M}\mathbf{x}\|_2$$

- « Ideal » decoding scheme

See [Foucart 2013]

Cost function



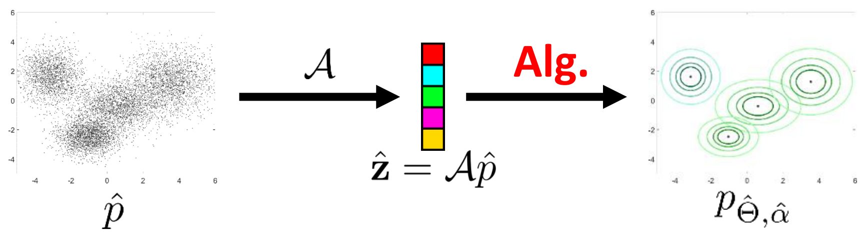


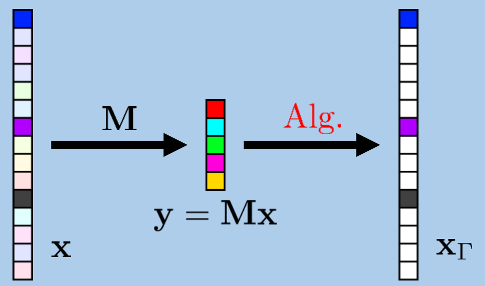
$$\boxed{\min_{\|\mathbf{x}\|_0 \leq s} \|\mathbf{y} - \mathbf{M}\mathbf{x}\|_2}$$

- « Ideal » decoding scheme
- NP-complete

See [Foucart 2013]

Cost function



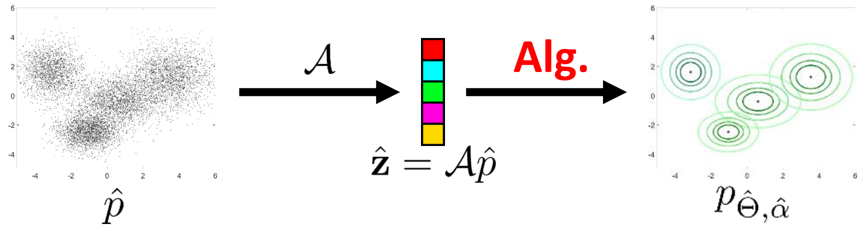


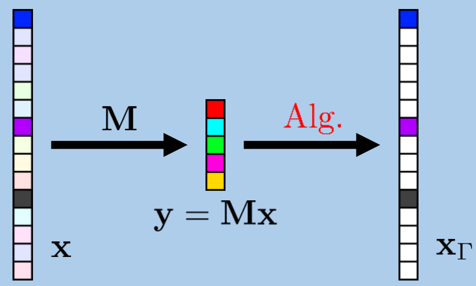
$$\boxed{\min_{\|\mathbf{x}\|_0 \leq s} \|\mathbf{y} - \mathbf{M}\mathbf{x}\|_2}$$

- « Ideal » decoding scheme
- NP-complete
- Two approaches:
  - Convex relaxation
  - Greedy

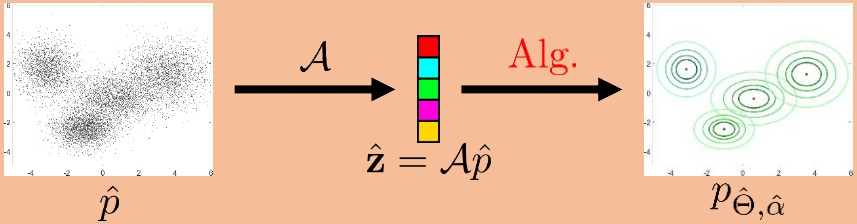
See [Foucart 2013]

Cost function





$$\min_{\|\mathbf{x}\|_0 \leq s} \|\mathbf{y} - \mathbf{M}\mathbf{x}\|_2$$

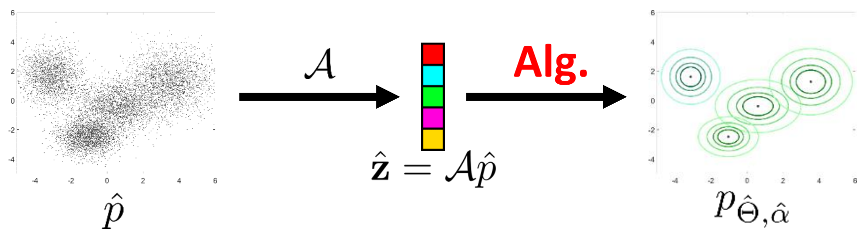


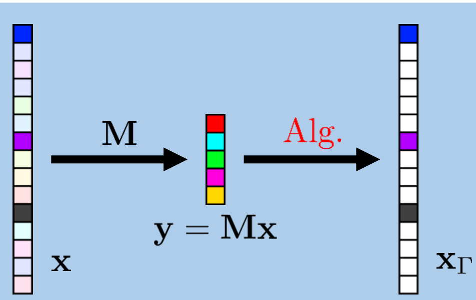
$$\min_{\Theta,\alpha} \|\hat{\mathbf{z}} - \mathcal{A}p_{\Theta,\alpha}\|_2$$

- « Ideal » decoding scheme
- NP-complete
- Two approaches:
  - Convex relaxation
  - Greedy

See [Foucart 2013]

Cost function

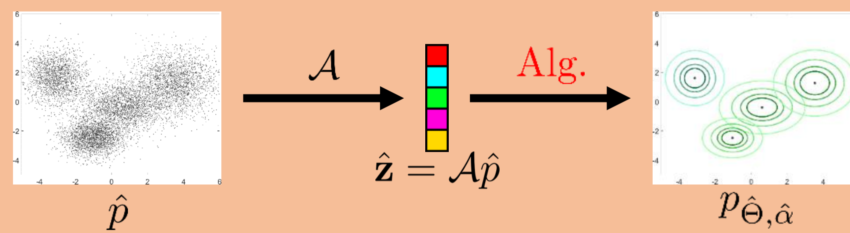




$$\min_{\|\mathbf{x}\|_0 \leq s} \|\mathbf{y} - \mathbf{M}\mathbf{x}\|_2$$

- « Ideal » decoding scheme
- NP-complete
- Two approaches:
  - Convex relaxation
  - Greedy

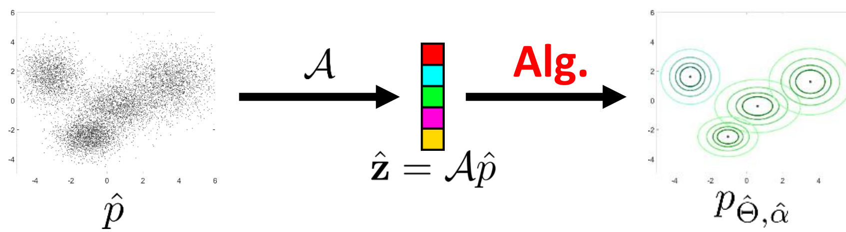
See [Foucart 2013]

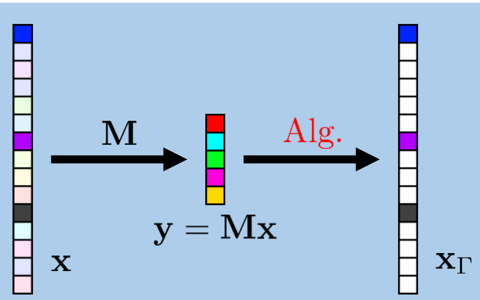


$$\min_{\Theta,\alpha} \|\hat{\mathbf{z}} - \mathcal{A}p_{\Theta,\alpha}\|_2$$

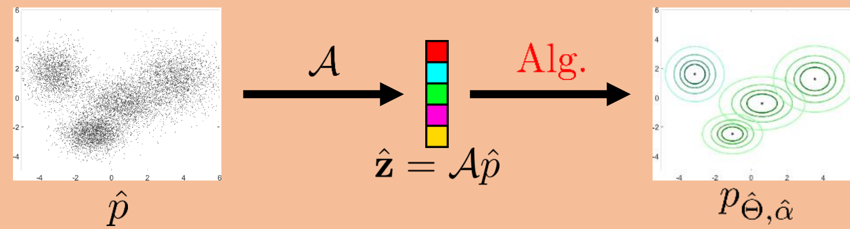
- Ideal decoding scheme ✓
  (Section 4)

Cost function





$$\min_{\|\mathbf{x}\|_0 \leq s} \|\mathbf{y} - \mathbf{M}\mathbf{x}\|_2$$



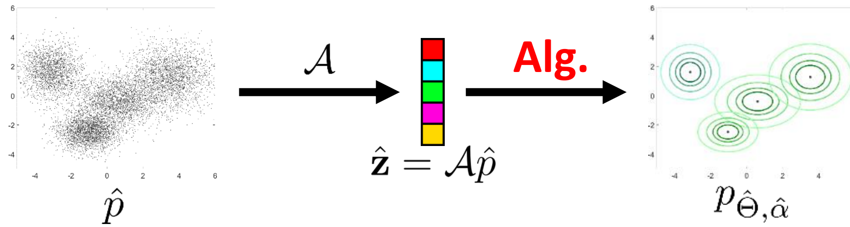
$$\min_{\Theta,\alpha} \|\hat{\mathbf{z}} - \mathcal{A}p_{\Theta,\alpha}\|_2$$

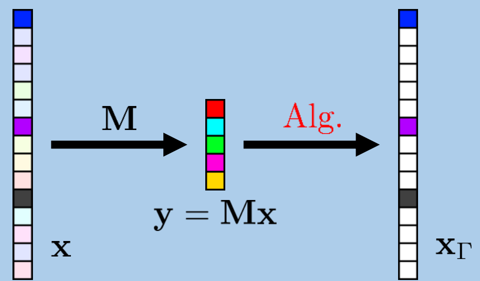
- « Ideal » decoding scheme
- NP-complete
- Two approaches:
  - Convex relaxation
  - Greedy

See [Foucart 2013]

- Ideal decoding scheme ✓ (Section 4)
- Highly non-convex

Cost function

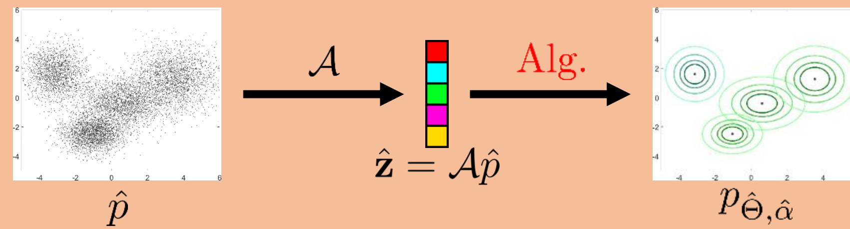




$$\boxed{\min_{\|\mathbf{x}\|_0 \leq s} \|\mathbf{y} - \mathbf{M}\mathbf{x}\|_2}$$

- « Ideal » decoding scheme
- NP-complete
- Two approaches:
  - Convex relaxation
  - Greedy

See [Foucart 2013]



$$\boxed{\min_{\Theta,\alpha} \|\hat{\mathbf{z}} - \mathcal{A}p_{\Theta,\alpha}\|_2}$$

- Ideal decoding scheme (Section 4)
- Highly non-convex
- Two approaches:
  - Convex relaxation *[Bunea 2010]*
  - Greedy Proposed

Proposed algorithm

Orthogonal Matching Pursuit (OMP)
[Mallat 1993, Pati 1993]

1. Add atom most correlated to residual
2. Perform Least-Squares
3. Repeat until desired sparsity

Proposed algorithm

OMP with Replacement (OMPR)

[Jain 2011]

1. Add atom most correlated to residual
2. **Perform Hard-Thresholding** (if necessary)
3. Perform Least-Squares
4. Repeat **twice desired sparsity**

Similar to CoSAMP [Needell 2008] or SubSpace Pursuit [Dai 2009]

Proposed algorithm

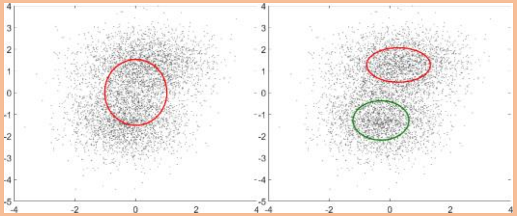
OMP with Replacement (OMPR)
[Jain 2011]

1. Add atom most correlated to residual
2. **Perform Hard-Thresholding** (if necessary)
3. Perform Least-Squares
4. Repeat **twice desired sparsity**

Similar to CoSAMP [Needell 2008] or SubSpace Pursuit [Dai 2009]

Compressive Learning OMPR (CLOMPR) (proposed)

1. Add atom most correlated to residual **with gradient descent**
2. Perform Hard-Thresholding
3. Perform **Non-Negative** Least-Squares
4. **Perform gradient descent on all parameters, initialized with current ones**

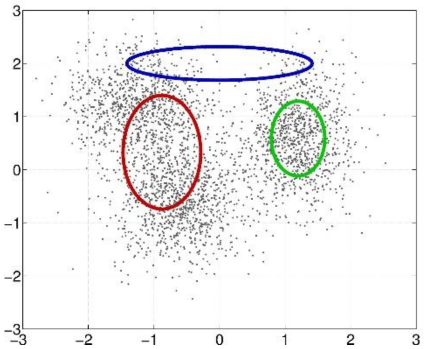


We cannot just add a component

5. Repeat twice desired sparsity

CLOMPR : illustration

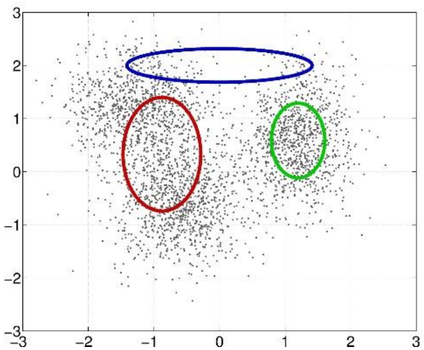
(schematic illustration)



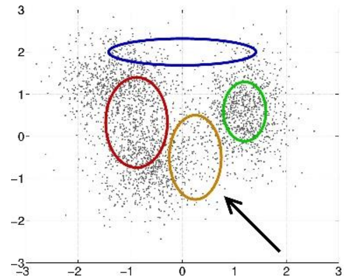
Goal : 3-GMM. Intermediary support.

CLOMPR : illustration

(schematic illustration)



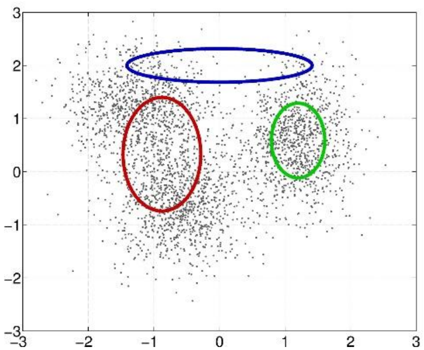
Goal : 3-GMM. Intermediary support.



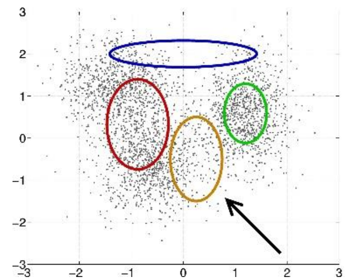
1 : Add atom with gradient descent

CLOMPR : illustration

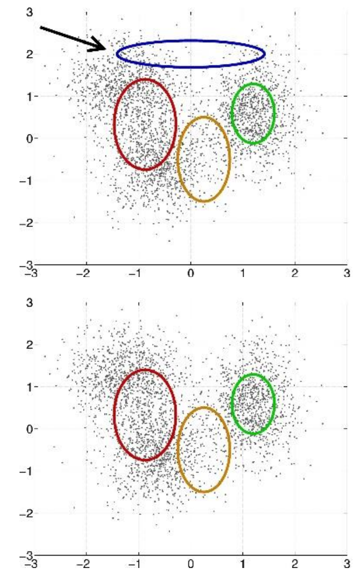
(schematic illustration)



Goal : 3-GMM. Intermediary support.



1 : Add atom with gradient descent

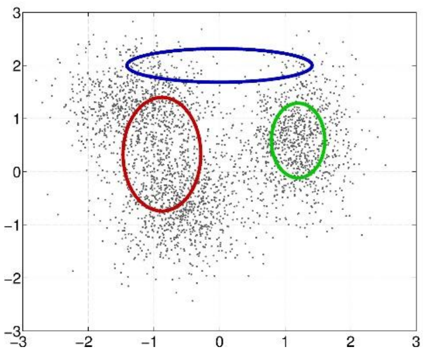


2 : Hard Thresholding

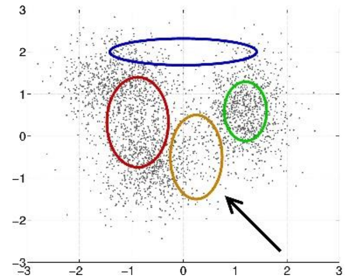
3 : **Non-Negative** Least-Squares

CLOMPR : illustration

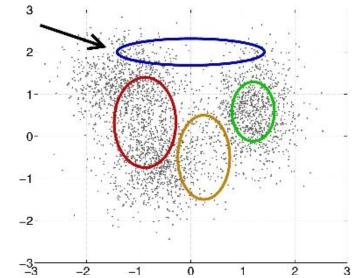
(schematic illustration)

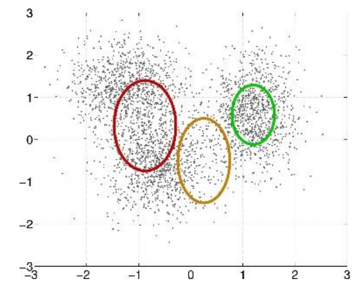


Goal : 3-GMM. Intermediary support.

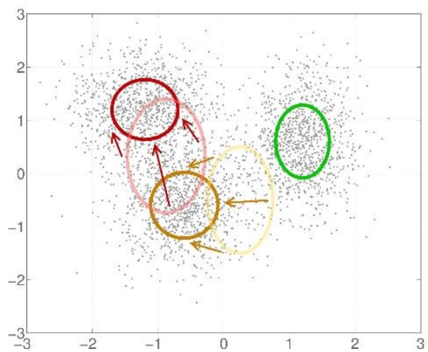


1 : Add atom with gradient descent





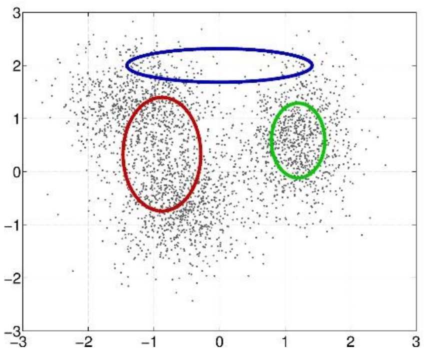
2 : Hard Thresholding
3 : **Non-Negative** Least-Squares



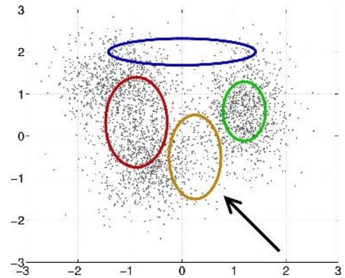
4 : Gradient Descent on all parameters

CLOMPR : illustration

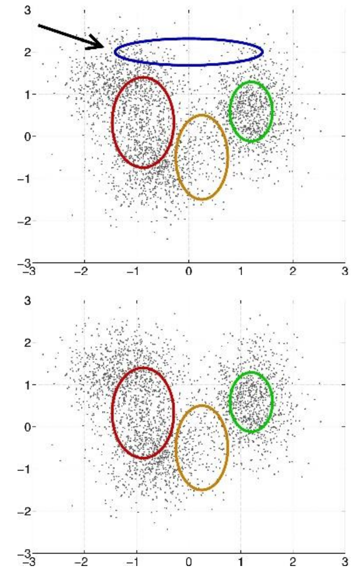
(schematic illustration)



Goal : 3-GMM. Intermediary support.

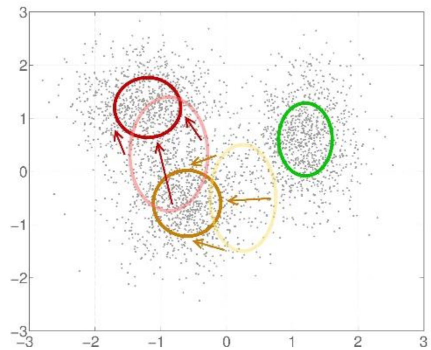


1 : Add atom with gradient descent

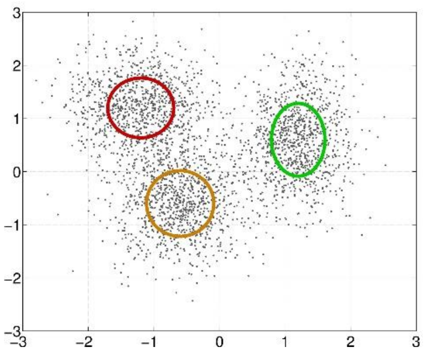


2 : Hard Thresholding

3 : **Non-Negative** Least-Squares



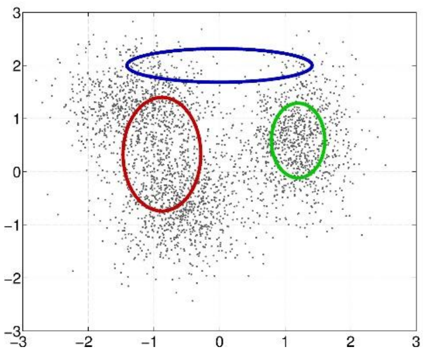
4 : Gradient Descent on all parameters



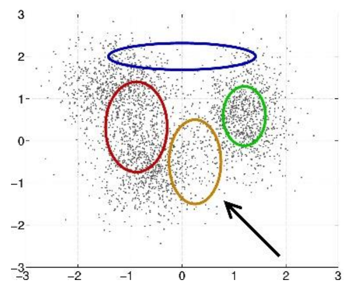
Final support

CLOMPR : illustration

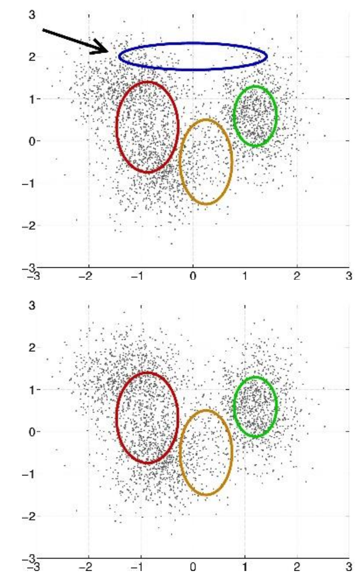
(schematic illustration)



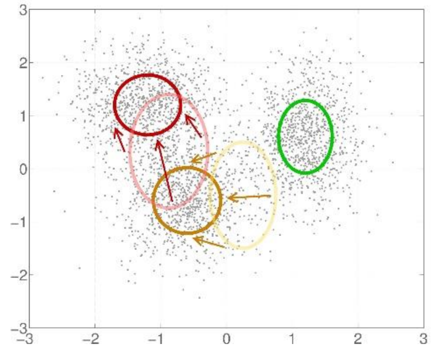
Goal : 3-GMM. Intermediary support.



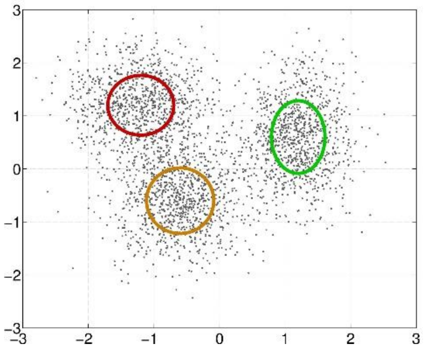
1 : Add atom with gradient descent



2 : Hard Thresholding
3 : **Non-Negative** Least-Squares



4 : Gradient Descent on all parameters



Final support

Define $\mathcal{A}$?

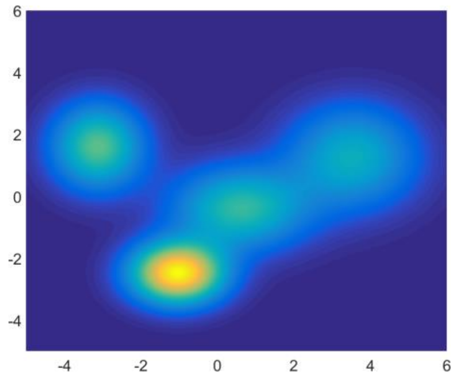
Choice of sketch

To implement CLOMPR, $\mathcal{A}p_\theta$ and $\nabla_\theta \mathcal{A}p_\theta$ must have a closed-form expression w.r.t. θ

Choice of sketch

To implement CLOMPR, $\mathcal{A}p_\theta$ and $\nabla_\theta \mathcal{A}p_\theta$ must have a closed-form expression w.r.t. θ

p is spatially localized



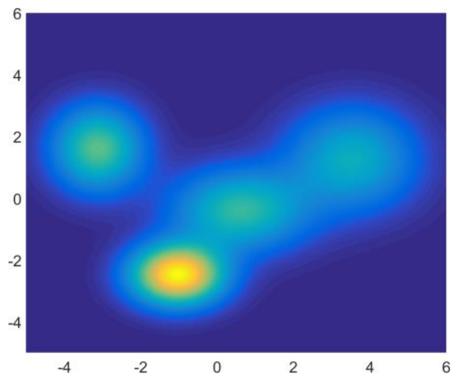
p

Choice of sketch

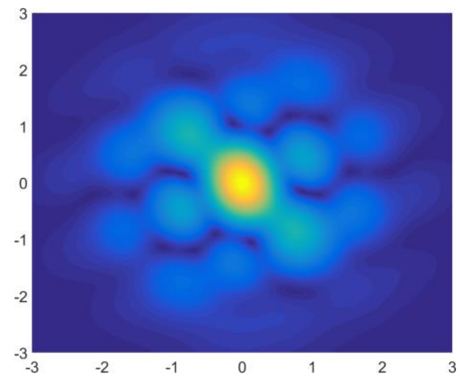
To implement CLOMPR, $\mathcal{A}p_\theta$ and $\nabla_\theta \mathcal{A}p_\theta$ must have a closed-form expression w.r.t. θ

p is spatially localized

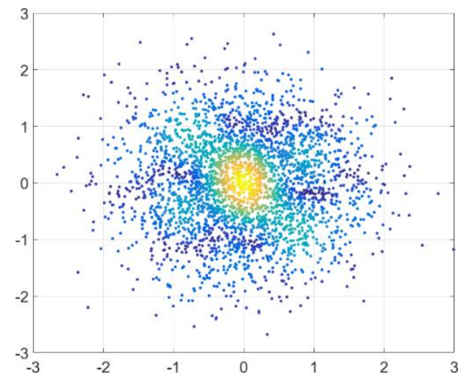
Need incoherent sampling -> Fourier sampling



p



$\psi_p(\omega) = \mathbb{E}_{\mathbf{x}\sim p} e^{-i\omega^T \mathbf{x}}$



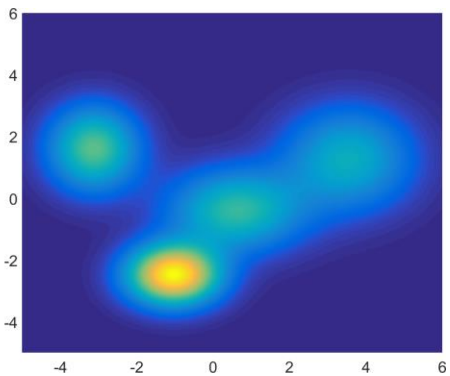
$\mathbf{Z}$

Choice of sketch

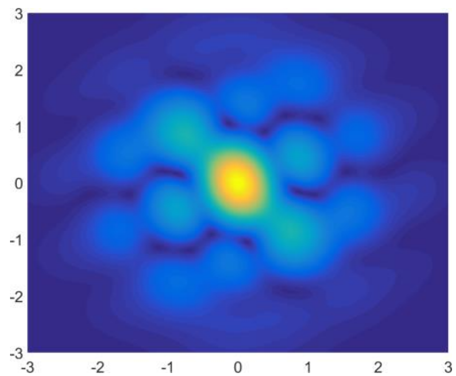
To implement CLOMPR, $\mathcal{A}p_\theta$ and $\nabla_\theta \mathcal{A}p_\theta$ must have a closed-form expression w.r.t. θ

p is spatially localized

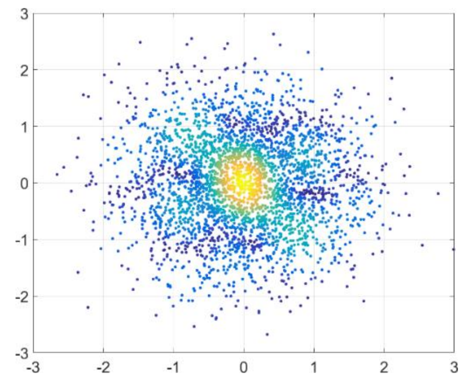
Need incoherent sampling -> Fourier sampling



p



$\psi_p(\omega) = \mathbb{E}_{\mathbf{x}\sim p} e^{-i\omega^T \mathbf{x}}$



$\mathbf{Z}$

$$\mathcal{A}p = \left[\psi_p(\omega_j)\right]_{j=1}^{m}$$

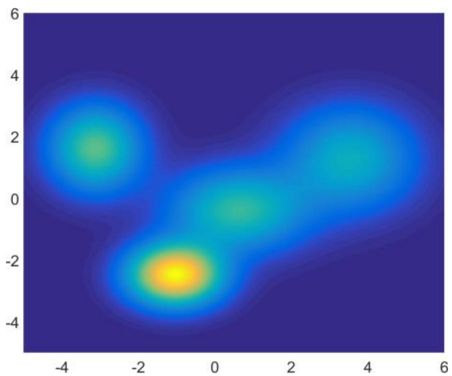
Closed-form for many models !
(including alpha-stable…)

Choice of sketch

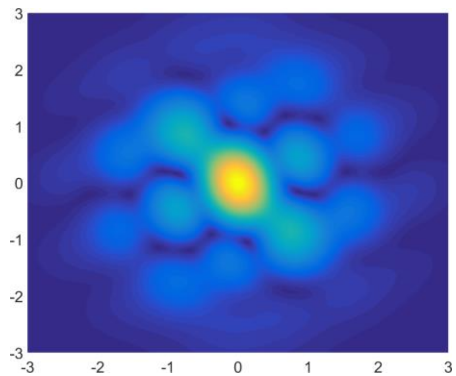
To implement CLOMPR, $\mathcal{A}p_\theta$ and $\nabla_\theta \mathcal{A}p_\theta$ must have a closed-form expression w.r.t. θ

p is spatially localized

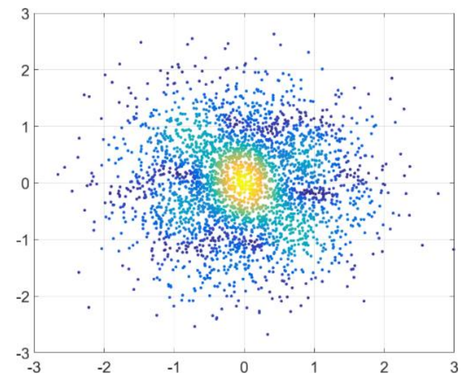
Need incoherent sampling -> Fourier sampling



p



$\psi_p(\omega) = \mathbb{E}_{\mathbf{x}\sim p} e^{-i\omega^T \mathbf{x}}$



$\mathbf{z}$

$$\mathcal{A}p = \left[\psi_p(\omega_j)\right]_{j=1}^{m}$$

$$\omega_j \overset{i.i.d.}{\sim} \Lambda$$

Closed-form for many models !
(including alpha-stable…)

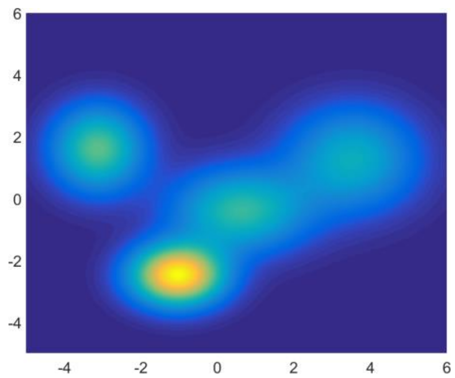
Random Fourier sampling *[Candes 2006]*
Random Fourier features *[Rahimi 2007]*

Choice of sketch

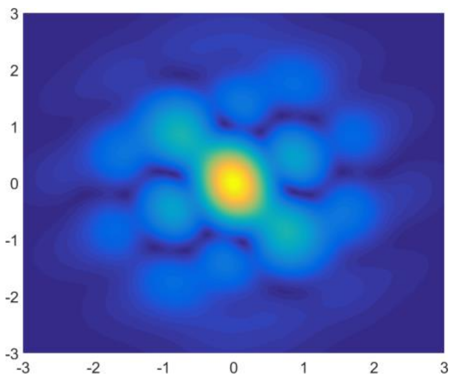
To implement CLOMPR, $\mathcal{A}p_\theta$ and $\nabla_\theta \mathcal{A}p_\theta$ must have a closed-form expression w.r.t. θ

p is spatially localized

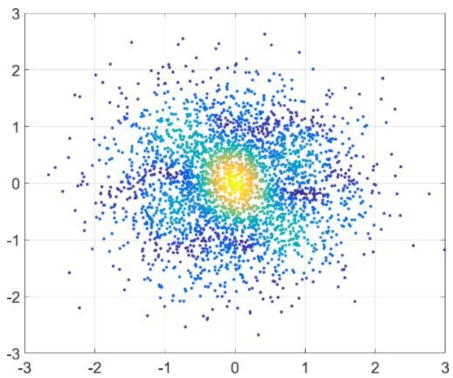
Need incoherent sampling -> Fourier sampling



p



$\psi_p(\omega) = \mathbb{E}_{\mathbf{x}\sim p} e^{-i\omega^T \mathbf{x}}$



$\mathbf{z}$

$$\mathcal{A}p = \left[\psi_p(\omega_j)\right]_{j=1}^{m}$$

$$\omega_j \overset{i.i.d.}{\sim} \Lambda$$

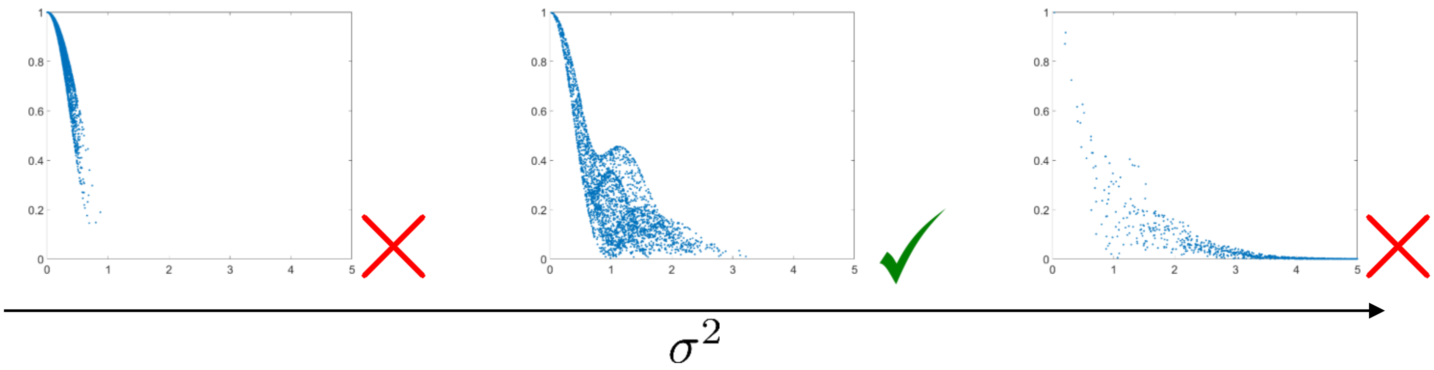
Define Λ ?

Closed-form for many models !
(including alpha-stable...)

Random Fourier sampling *[Candes 2006]*
Random Fourier features *[Rahimi 2007]*

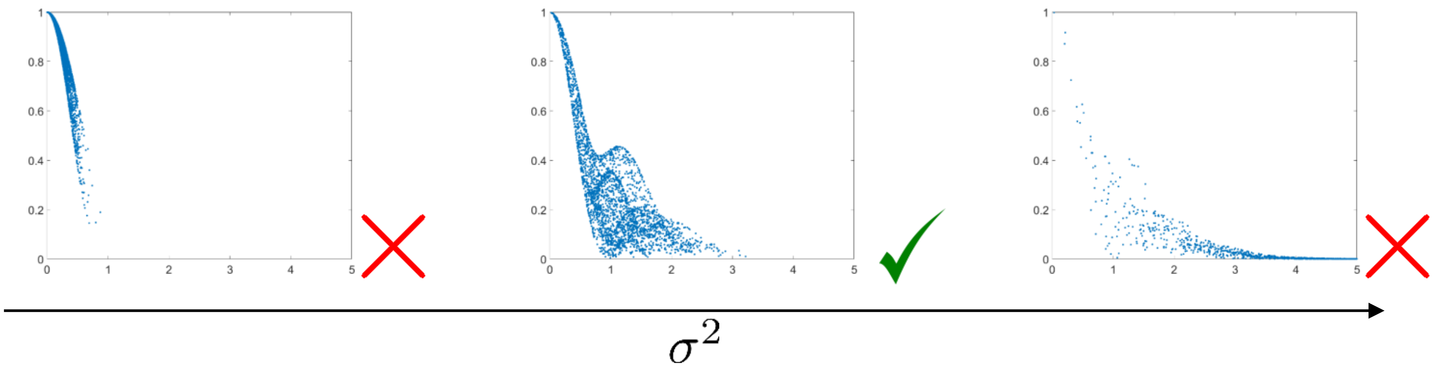
Designing the frequency distribution

Must adjust « scale » of distribution



Designing the frequency distribution

Must adjust « scale » of distribution

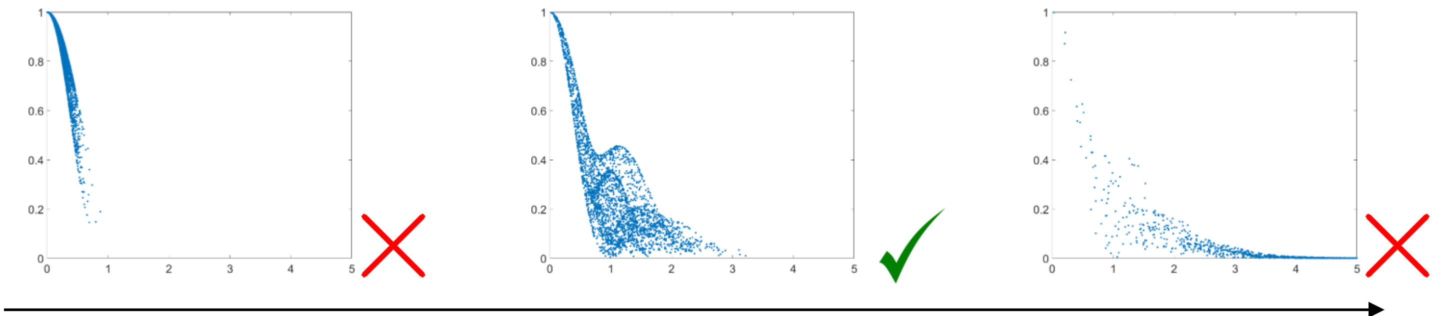


Adjust by hand

- *Not that difficult…*
- *The method is quite robust*

Designing the frequency distribution

Must adjust « scale » of distribution



σ^2

Adjust by hand

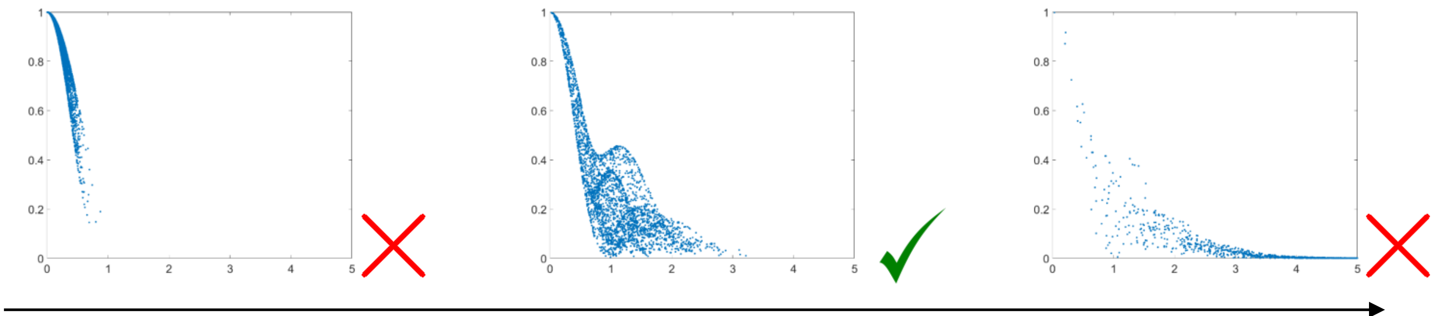
- *Not that difficult…*
- *The method is quite robust*

Cross-validation

- *Can be very long !*
- *Used in practice* *[Sutherland2015]*

Designing the frequency distribution

Must adjust « scale » of distribution



σ^2

Adjust by hand

- *Not that difficult...*
- *The method is quite robust*

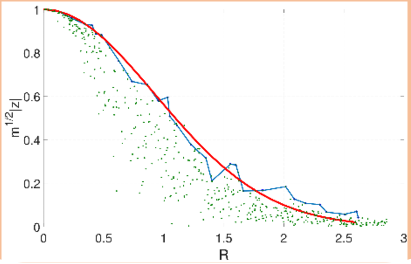
Cross-validation

- *Can be very long !*
- *Used in practice [Sutherland2015]*

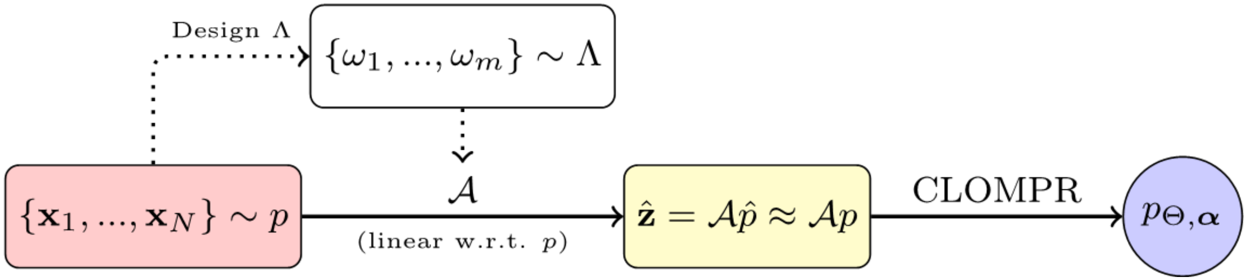
Proposed

Automatic

- *Partial pre-processing*
- *Heuristic based on GMMs-like distributions*



Summary



Given database, m , K

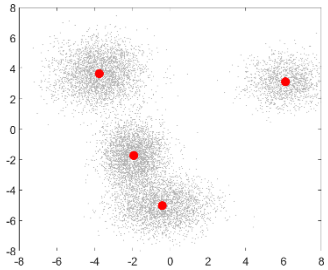
1. Design $\mathcal{A}$
   - Partial pre-processing to choose Λ
   - Draw $(\omega_1, ..., \omega_m) \overset{i.i.d.}{\sim} \Lambda$
2. Compute $\hat{\mathbf{z}} = \frac{1}{N}\left[\sum_i e^{-i\omega_j^T \mathbf{x}_i}\right]_{j=1}^m$
   - Online, distributed, GPU…
3. Derive mixture model $p_{\Theta,\alpha}$ with CLOMPR

Outline

1. Introduction
2. Practical Approach
3. **Results**
4. Theoretical analysis
5. Conclusion and outlooks

Application : K-means, GMM

K-means

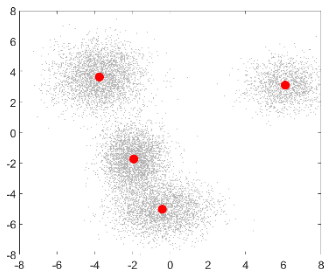


Classic approach

- Goal : $\min_\Theta \sum_{i=1}^N \left(\min_{1 \leq k \leq K} \|\mathbf{x}_i - \theta_k\|_2^2 \right)$
- Algorithm : **Lloyd-Max** *[Lloyd 1982]*
  (Matlab's `kmeans`*)*

Application : K-means, GMM

K-means



Classic approach

- Goal : $\min_\Theta \sum_{i=1}^N \left(\min_{1\leq k\leq K} \|\mathbf{x}_i - \theta_k\|_2^2 \right)$
- Algorithm : **Lloyd-Max** *[Lloyd 1982]*
  (Matlab's `kmeans`*)*

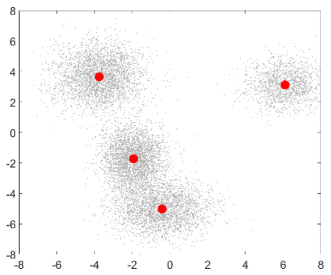
Compressive approach

- Model : $p_\theta = \delta_\theta \qquad \theta \in \mathbb{R}^n$

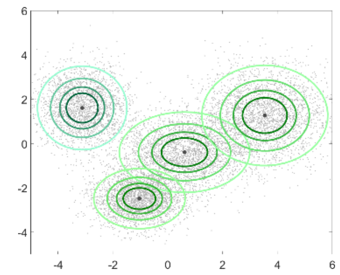
(clustered distribution = noisy mixture of Diracs)

Application : K-means, GMM

K-means



GMM diagonal cov.



Classic approach

- Goal : $\min_\Theta \sum_{i=1}^N \left(\min_{1 \leq k \leq K} \|\mathbf{x}_i - \theta_k\|_2^2 \right)$
- Algorithm : **Lloyd-Max** *[Lloyd 1982]*

  (Matlab's `kmeans`*)*

Classic approach

- Goal : $\min_{\Theta,\alpha} \sum_{i=1}^N (-\log p_{\Theta,\alpha}(\mathbf{x}_i))$
- Algorithm : **EM** *[Dempster 1977]*

  (VLFeat's `gmm`*)*

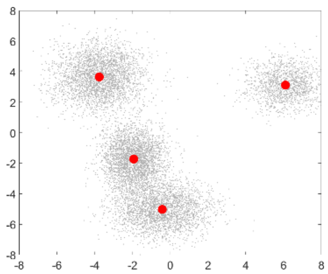
Compressive approach

- Model : $p_\theta = \delta_\theta \qquad \theta \in \mathbb{R}^n$

(clustered distribution = noisy mixture of Diracs)

Application : K-means, GMM

K-means



Classic approach

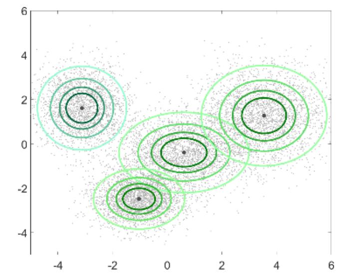
- Goal : $\min_\Theta \sum_{i=1}^N \left(\min_{1 \le k \le K} \|\mathbf{x}_i - \theta_k\|_2^2 \right)$
- Algorithm : **Lloyd-Max** *[Lloyd 1982]* *(Matlab's `kmeans`)*

Compressive approach

- Model : $p_\theta = \delta_\theta \qquad \theta \in \mathbb{R}^n$

(clustered distribution = noisy mixture of Diracs)

GMM diagonal cov.



Classic approach

- Goal : $\min_{\Theta,\alpha} \sum_{i=1}^N (-\log p_{\Theta,\alpha}(\mathbf{x}_i))$
- Algorithm : **EM** *[Dempster 1977]* *(VLFeat's `gmm`)*

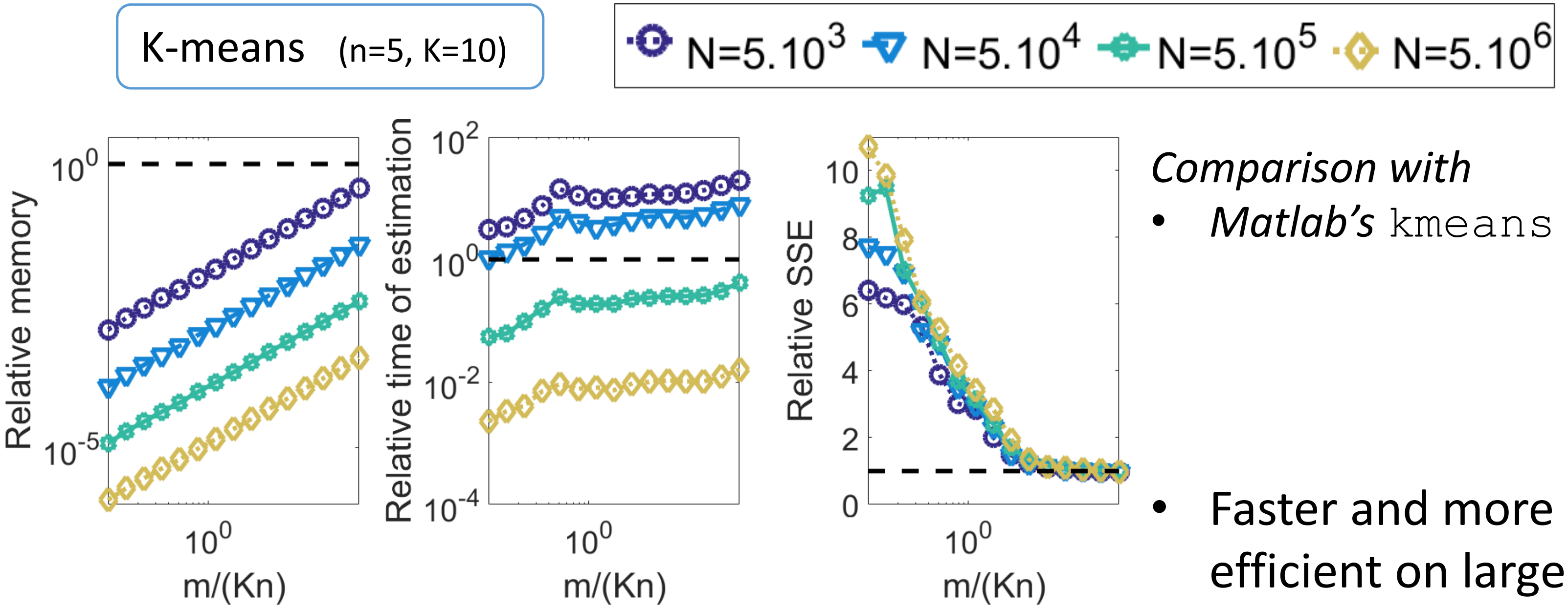
Compressive approach

- Model : $\theta = (\mu, \sigma) \in \mathbb{R}^{2n}$

  $p_\theta = \mathcal{N}(\mu, diag(\sigma))$

Large-scale result

K-means (n=5, K=10)



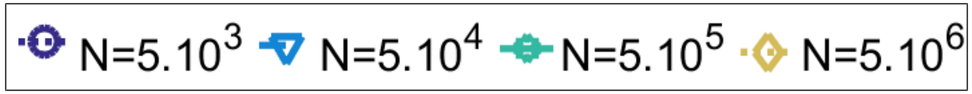
Comparison with

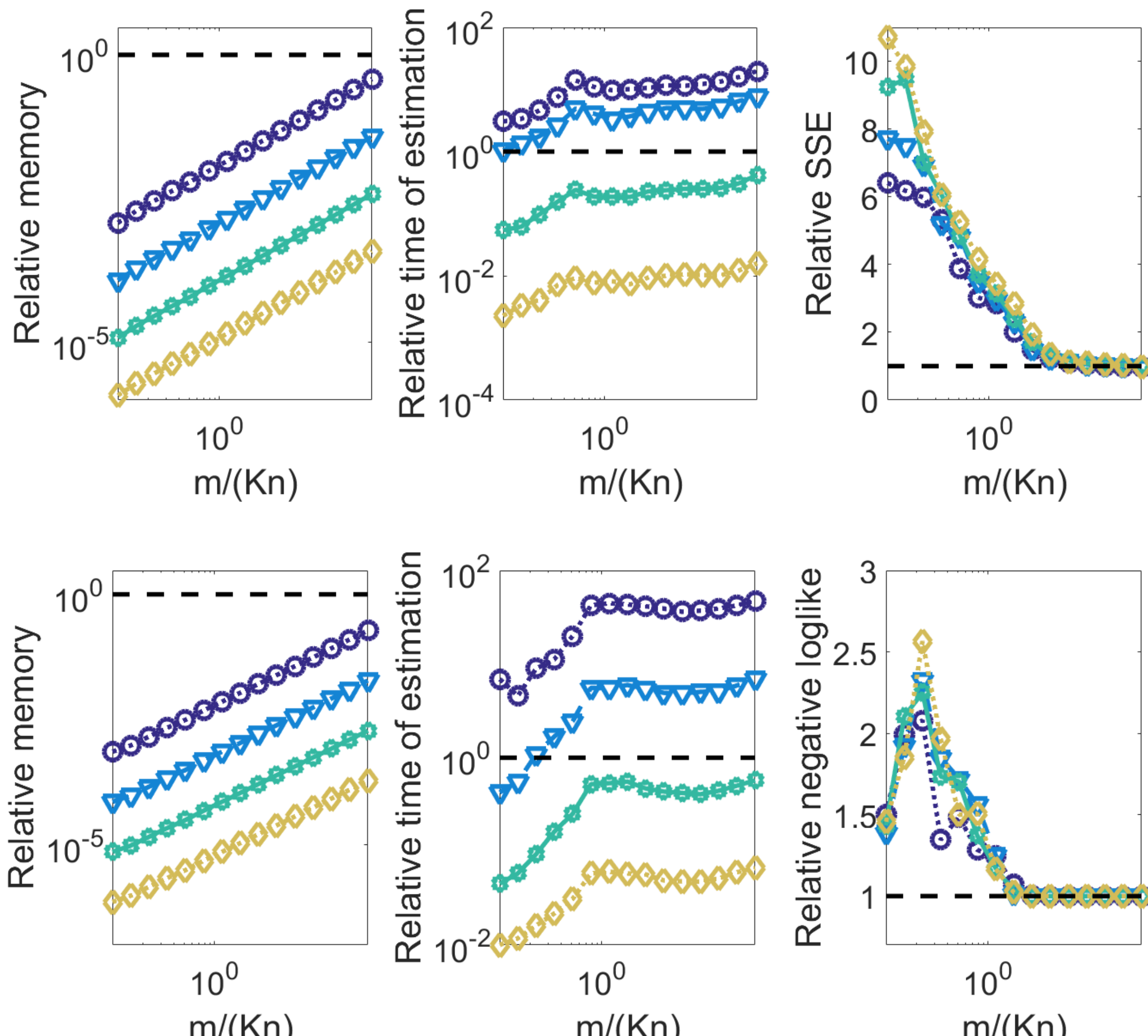
- *Matlab's* `kmeans`

- Faster and more memory efficient on large databases
- Number of measurements does not depend on N

Large-scale result

K-means (n=5, K=10)





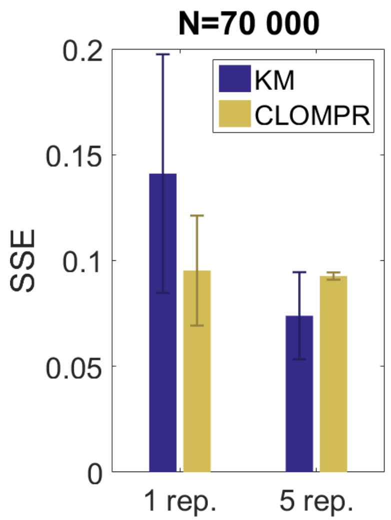
Comparison with

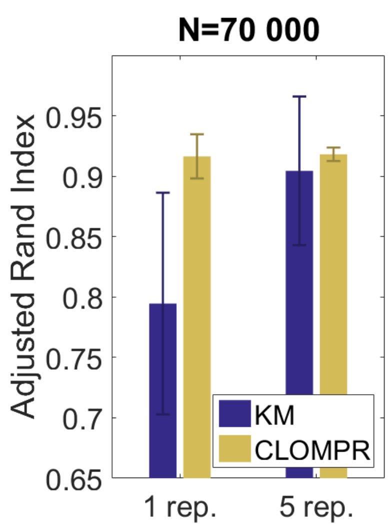
- *Matlab's* `kmeans`
- *VLFeat's* `gmm`

- Faster and more memory efficient on large databases
- Number of measurements does not depend on N

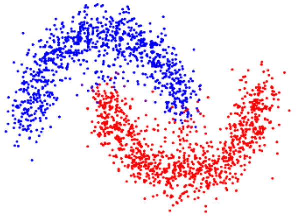
GMMs, diagonal cov. (n=5, K=5)

Application : spectral clustering



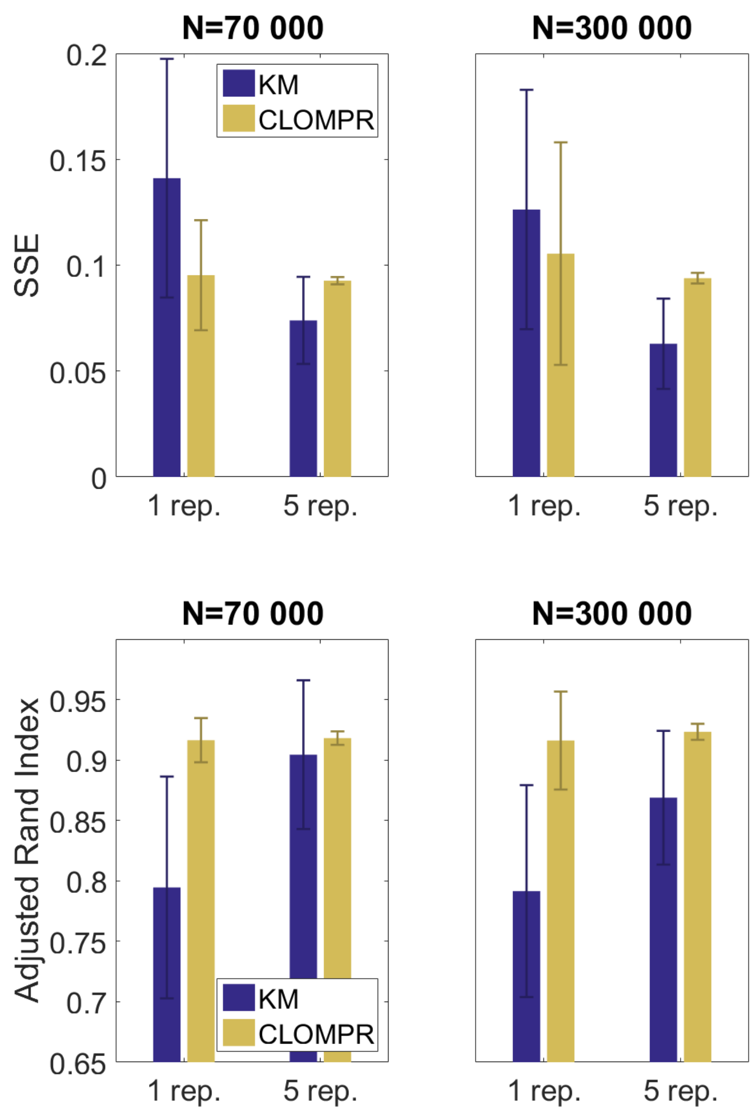


K-means (n=10, K=10, m=1000)
Mean and var. over 50 exp.

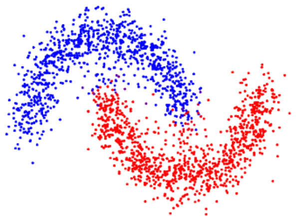


Spectral clustering for classification [Uw 2001], augmented MNIST database [Loosli 2007].

Application : spectral clustering

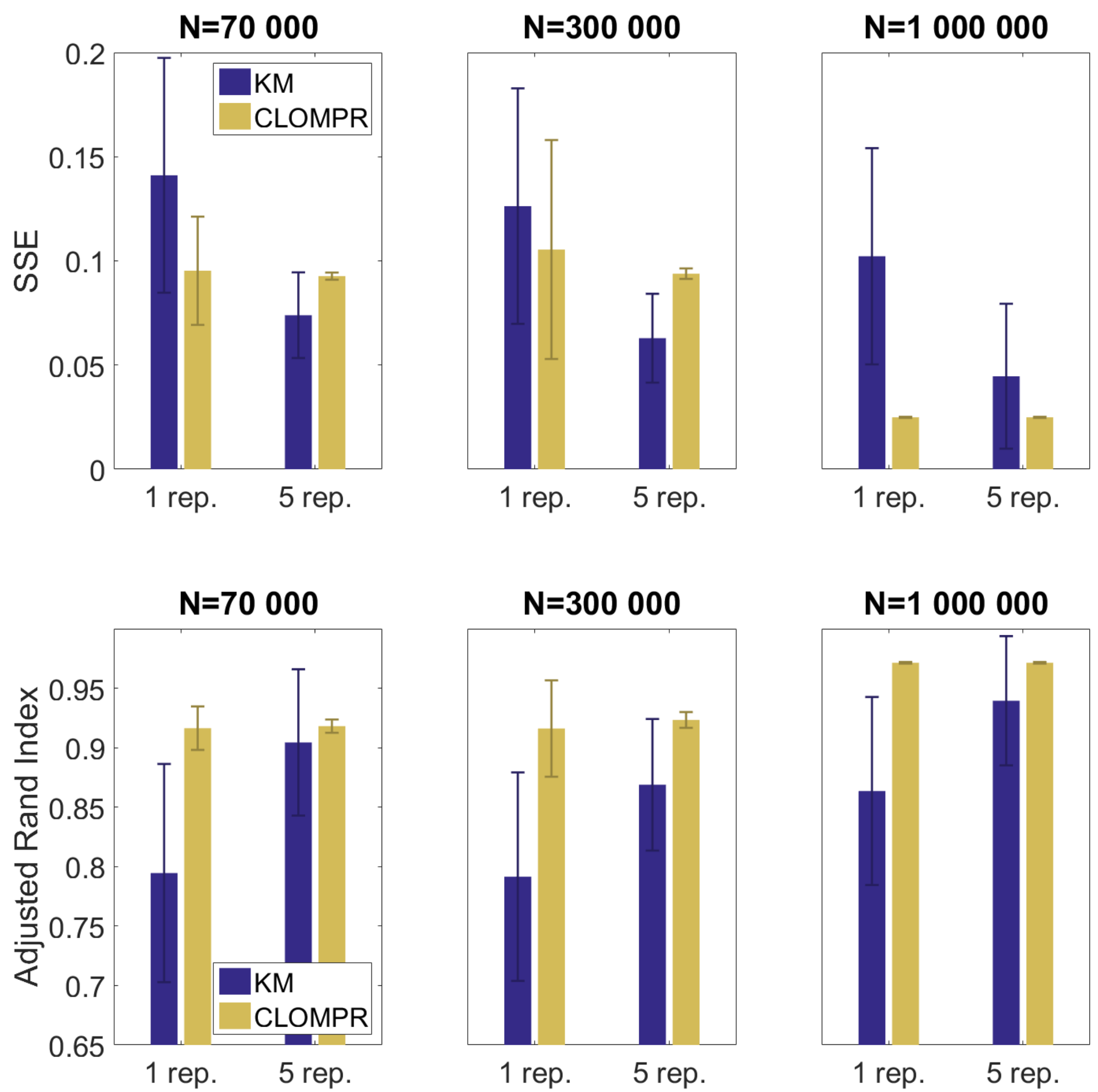


K-means (n=10, K=10, m=1000)
Mean and var. over 50 exp.

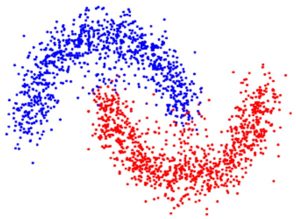


Spectral clustering for classification [Uw 2001], augmented MNIST database [Loosli 2007].

Application : spectral clustering



K-means (n=10, K=10, m=1000)
Mean and var. over 50 exp.



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

- CLOMPR **performs better** and is **more stable** with a large database

Application : speaker recognition

Variant of CLOMPR, faster at large K

GMM (n=12, K=64)

	(Hierarchical) CLOMPR			EM
	$m = 10^3$	$m = 10^4$	$m = 10^5$	
$N = 3.10^5$	37.15	30.24	29.77	29.53
$\mathbf{N = 2.10^8}$	36.57	**28.96**	**28.59**	**N/A**

Classical method for speaker recognition [Reynolds 2000] (for proof of concept) NIST 2005 database, MFCCs.

- Also performs better on a large database.

Outline

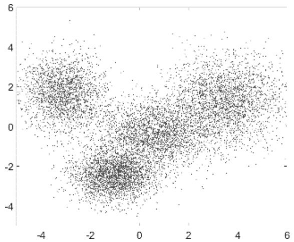
1. Introduction
2. Practical Approach
3. Results
4. **Theoretical analysis**
5. Conclusion and outlooks

Information-preservation guarantees

Guarantee for CLOMPR ? Difficult ! (non-convex, random…)

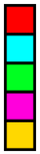
Information-preservation guarantees

Guarantee for CLOMPR ? Difficult ! (non-convex, random…)



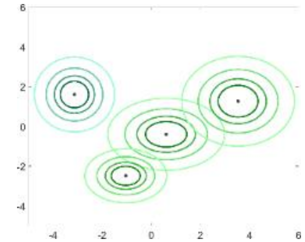
$\hat{p}$

$\mathcal{A}$



$\hat{\mathbf{z}} = \mathcal{A}\hat{p}$

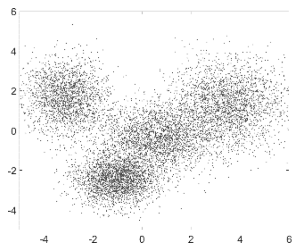
Solve cost func.



$$p_{\hat{\Theta},\hat{\alpha}} \in \arg\min_{\Theta,\alpha} \|\hat{\mathbf{z}} - \mathcal{A}p_{\Theta,\alpha}\|_2$$

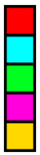
Information-preservation guarantees

Guarantee for CLOMPR ? Difficult ! (non-convex, random…)



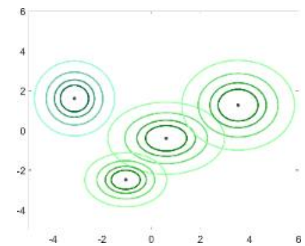
$\hat{p}$

$\xrightarrow{\mathcal{A}}$



$\hat{\mathbf{z}} = \mathcal{A}\hat{p}$

Solve cost func. $\longrightarrow$



$p_{\hat{\Theta},\hat{\alpha}} \in \arg\min_{\Theta,\alpha} \|\hat{\mathbf{z}} - \mathcal{A}p_{\Theta,\alpha}\|_2$

- Robustness to using $\hat{\mathbf{z}} = \mathcal{A}\hat{p}$ instead of $\mathbf{z} = \mathcal{A}p$?
- Robustness to p not being **exactly** a mixture model ?
- Guarantees in terms of usual learning cost functions ?
  - K-means : sum of distances to closest centroid
  - GMMs : negative log-likelihood

K-means : result

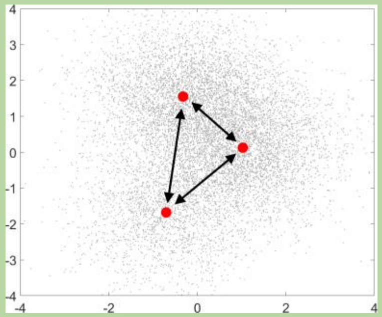
Goal minimize $R(\Theta) = \mathbb{E}_{\mathbf{x}\sim p^*}\left[\min_k \|\mathbf{x} - \theta_k\|_2^2\right]$ *(expected risk)*

K-means : result

Goal minimize $R(\Theta) = \mathbb{E}_{\mathbf{x}\sim p^*}\left[\min_k \|\mathbf{x}-\theta_k\|_2^2\right]$ *(expected risk)*

Hyp.

- ε - separation



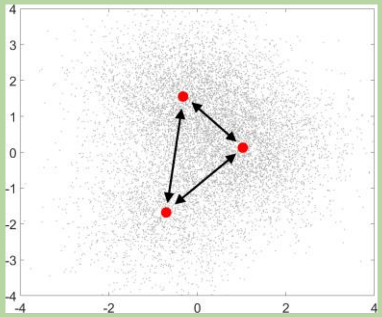
- M - bounded domain
- **Reweighted** Fourier features
  (needed for theory, no effect in practice)

K-means : result

Goal minimize $R(\Theta) = \mathbb{E}_{\mathbf{x}\sim p^*}\left[\min_k \|\mathbf{x}-\theta_k\|_2^2\right]$ *(expected risk)*

Hyp.

- ε - separation



- M - bounded domain
- **Reweighted** Fourier features
  (needed for theory, no effect in practice)

If $m \geq \mathcal{O}\left(K^2 n^3 \mathtt{polylog}(K,n)\log(M/\varepsilon)\right)$

w.h.p. $R(\hat{\Theta}) \lesssim R(\Theta^*) + \mathcal{O}\left(\sqrt{n^2 K/N}\right)$

GMMs with known covariance : result

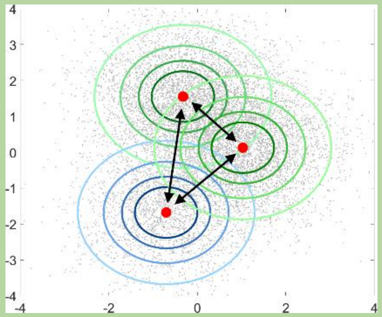
Goal minimize $R(\Theta,\alpha)=\mathbb{E}_{\mathbf{x}\sim p^*}\Big[-\log p_{\Theta,\alpha}(\mathbf{x})\Big]$ $p_{\Theta,\alpha}=\sum_k \alpha_k \mathcal{N}(\theta_k,\Sigma)$ *(expected risk)*

GMMs with known covariance : result

Goal minimize $R(\Theta,\alpha) = \mathbb{E}_{\mathbf{x}\sim p^*}\left[-\log p_{\Theta,\alpha}(\mathbf{x})\right]$ $p_{\Theta,\alpha} = \sum_k \alpha_k \mathcal{N}(\theta_k, \Sigma)$ *(expected risk)*

Hyp.

- Large enough separation



- M - bounded domain
- Fourier features

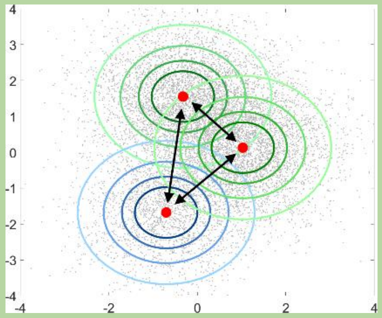
GMMs with known covariance : result

Goal minimize $R(\Theta, \alpha) = \mathbb{E}_{\mathbf{x} \sim p^*}\left[-\log p_{\Theta,\alpha}(\mathbf{x}) \right]$ $p_{\Theta,\alpha} = \sum_k \alpha_k \mathcal{N}(\theta_k, \Sigma)$

(expected risk)

Hyp.

- Large enough separation



- M - bounded domain
- Fourier features

If m large enough w.h.p.

$$R(\hat{\Theta}, \hat{\alpha}) - R(\Theta^*, \alpha^*) \lesssim \inf_{\Theta,\alpha} \|p^* - p_{\Theta,\alpha}\|_{L^1} + \mathcal{O}\left(1/\sqrt{N}\right)$$

L1 distance from p* to the set of (separated) GMMs

GMM trade-off

Separation of means

Size of sketch

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

GMM with **unknown diagonal** covariance

$$\underbrace{\|p^* - p_{\hat{\Theta},\hat{\alpha}}\|}_{} \lesssim \inf_{\Theta,\alpha} \|p^* - p_{\Theta,\alpha}\|_{L^1} + \mathcal{O}\left(1/\sqrt{N}\right) + \underbrace{\mathcal{O}\left(1/\sqrt{m}\right)}_{}$$

Related to learning cost function ?

Efficiency of « compressive » approach ?

GMM with **unknown diagonal** covariance

$$\underbrace{\|p^* - p_{\hat{\Theta},\hat{\alpha}}\|}_{} \lesssim \inf_{\Theta,\alpha} \|p^* - p_{\Theta,\alpha}\|_{L^1} + \mathcal{O}\left(1/\sqrt{N}\right) + \underbrace{\mathcal{O}\left(1/\sqrt{m}\right)}_{}$$

Related to learning cost function ?

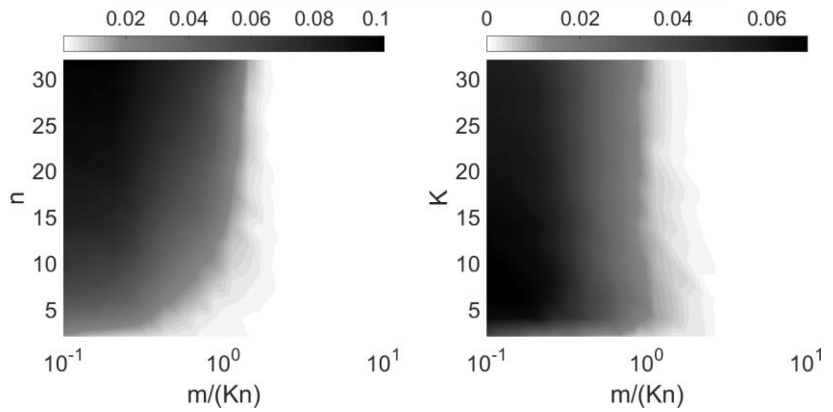
Efficiency of « compressive » approach ?

Nevertheless : $p_{\hat{\Theta},\hat{\alpha}} \xrightarrow{N,m\to\infty} p^*$ when p^* is exactly a GMM

Number of measurements

Relative SSE

K-means



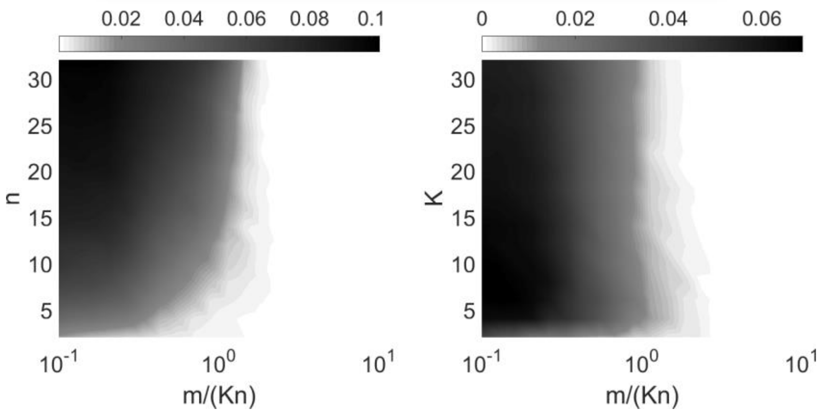
In theory, at least

$$m \geq \mathcal{O}(K^2 n^2)$$

Number of measurements

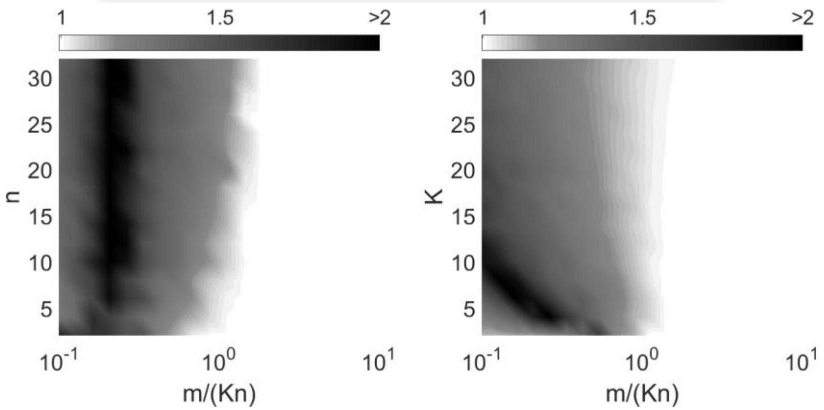
Relative SSE

K-means



GMMs, known cov.

Relative loglike



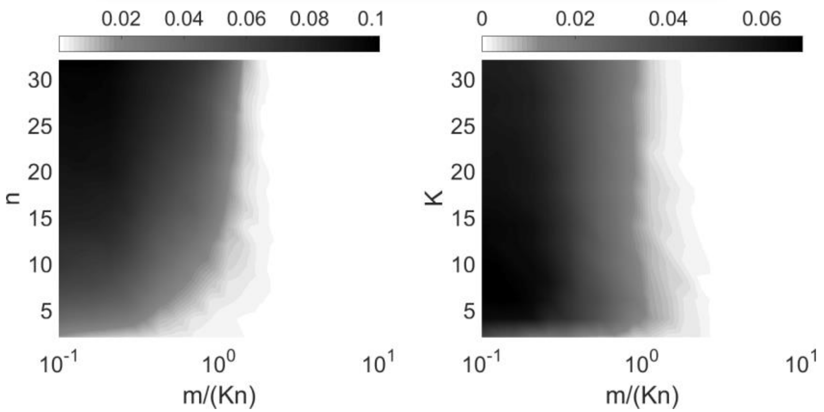
In theory, at least

$$m \geq \mathcal{O}(K^2 n^2)$$

Number of measurements

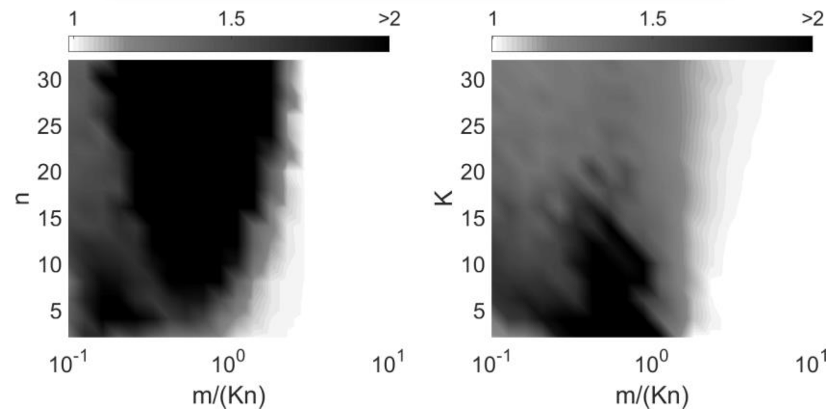
K-means

Relative SSE



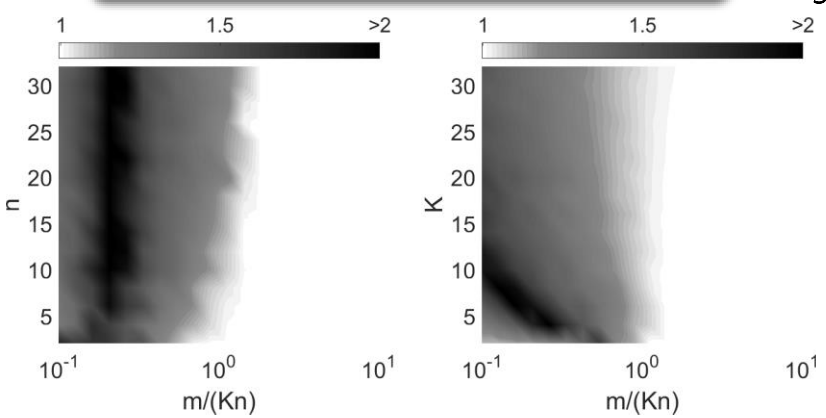
GMMs, diagonal cov.

Relative loglike



GMMs, known cov.

Relative loglike



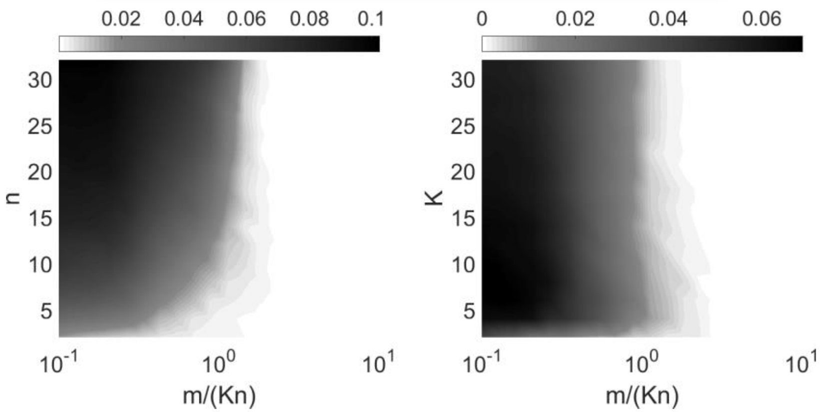
In theory, at least

$$m \geq \mathcal{O}(K^2 n^2)$$

Number of measurements

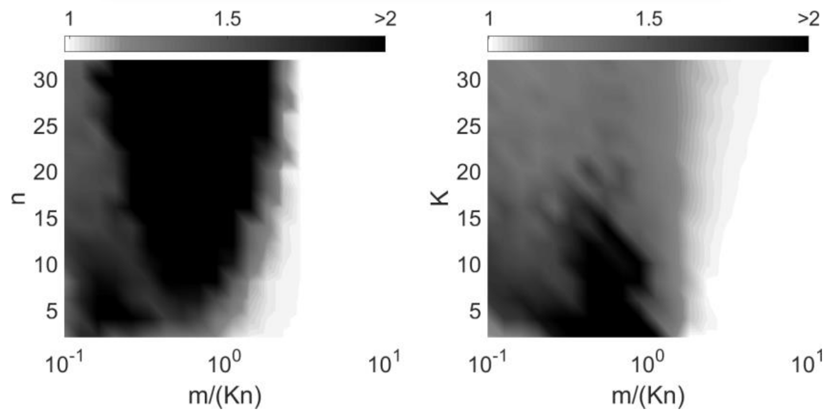
Relative SSE

K-means



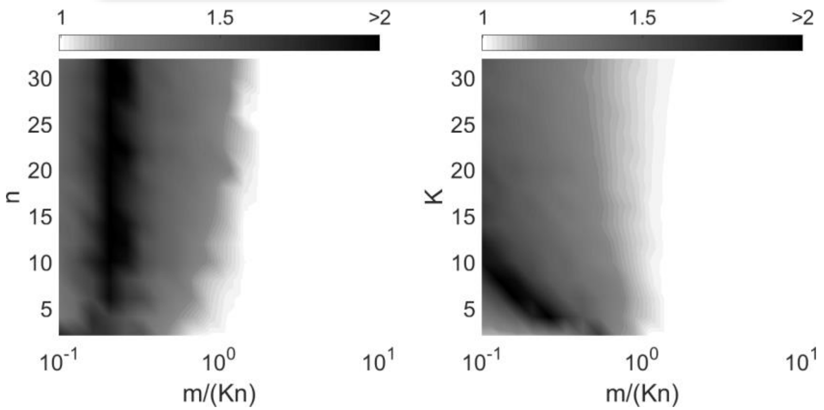
GMMs, diagonal cov.

Relative loglike



GMMs, known cov.

Relative loglike



In theory, at least

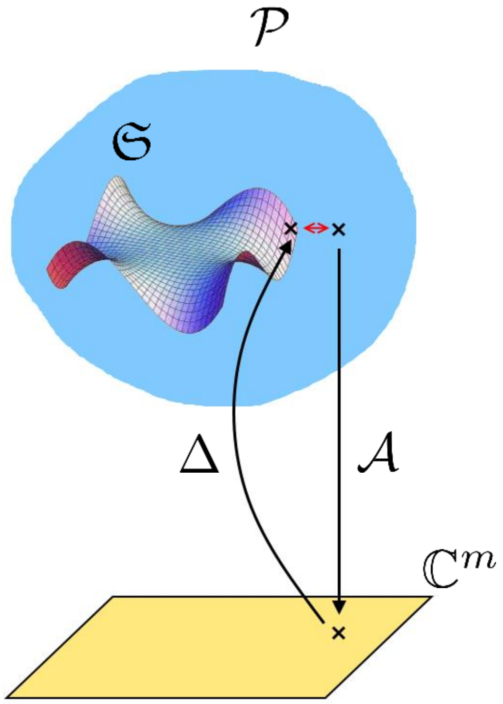
$$m \geq \mathcal{O}(K^2 n^2)$$

Empirically ?

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

Sketch of proof : principle

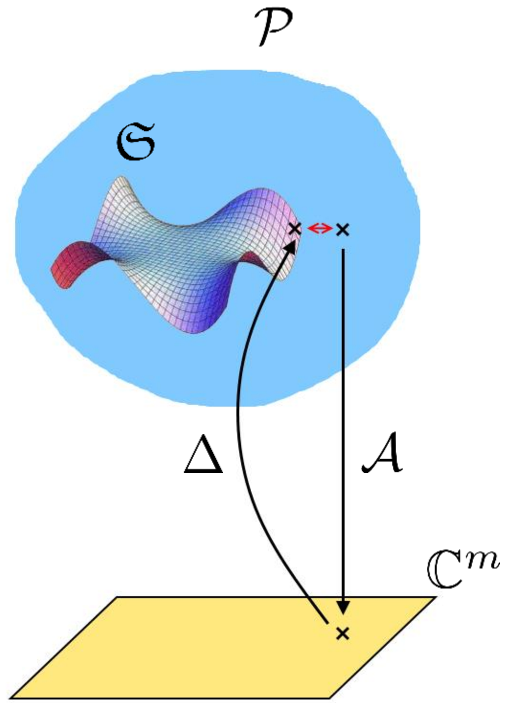
Goal : Existence of instance Optimal Decoder



$$\|p^* - \Delta(\hat{\mathbf{z}})\| \lesssim d(p^*, \mathfrak{S}) + \underbrace{\|\mathcal{A}(p^* - \hat{p})\|}_{\mathcal{O}(1/\sqrt{N})}$$

Sketch of proof : principle

Goal : Existence of instance Optimal Decoder



$$\|p^* - \Delta(\hat{\mathbf{z}})\| \lesssim d(p^*, \mathfrak{S}) + \underbrace{\|\mathcal{A}(p^* - \hat{p})\|}_{\mathcal{O}(1/\sqrt{N})}$$

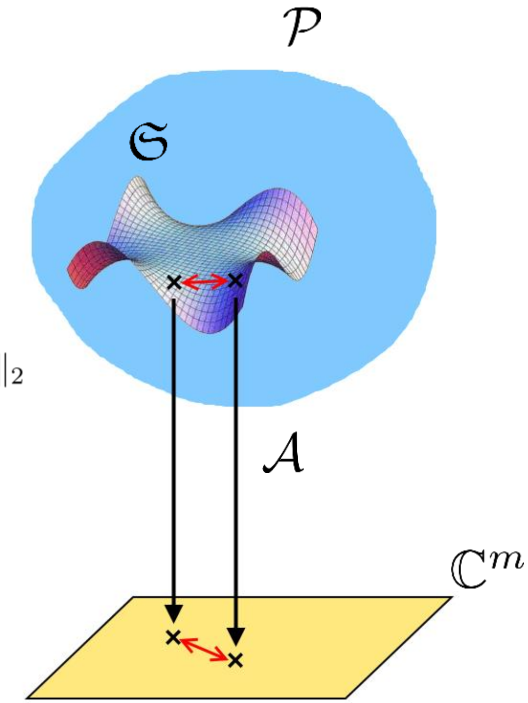
[Bourrier 2014]

$$\Longrightarrow$$

$$\Longleftarrow$$

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

Lower Restricted Isometry Property (LRIP)



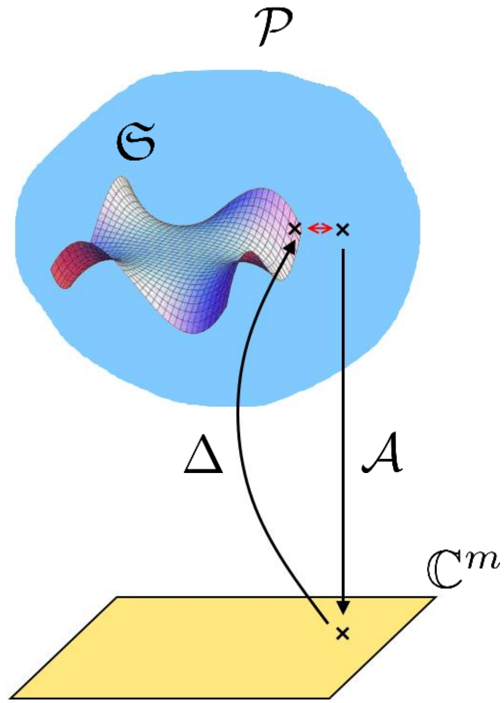
$\forall q, q' \in \mathfrak{S}$

$$\|q - q'\| \lesssim \|\mathcal{A}(q - q')\|_2$$

Sketch of proof : principle

Goal : Existence of instance Optimal Decoder

Lower Restricted Isometry Property (LRIP)

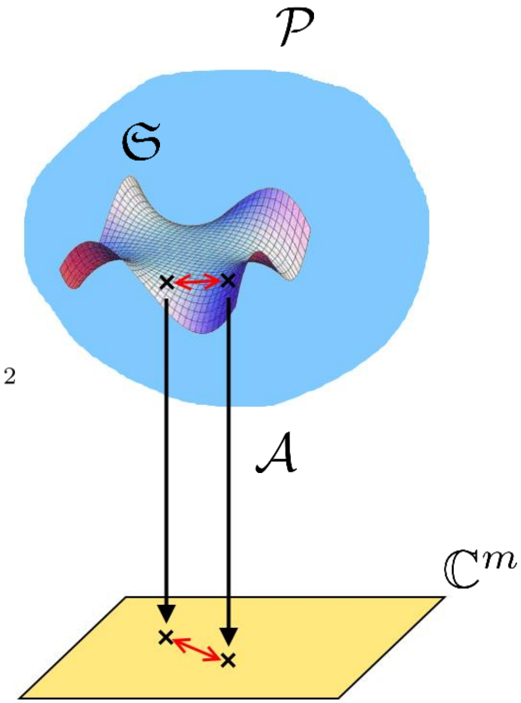


[Bourrier 2014]

$\Longrightarrow$

$\Longleftarrow$

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



$$\|p^* - \Delta(\hat{\mathbf{z}})\| \lesssim d(p^*, \mathfrak{S}) + \underbrace{\|\mathcal{A}(p^* - \hat{p})\|}_{\mathcal{O}(1/\sqrt{N})}$$

$+\eta$

$\forall q, q' \in \mathfrak{S}$

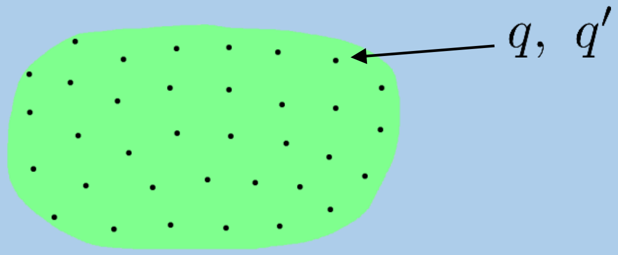
$$\|q - q'\| \lesssim \|\mathcal{A}(q - q')\|_2$$

$+\eta$

Ex : Quantization error

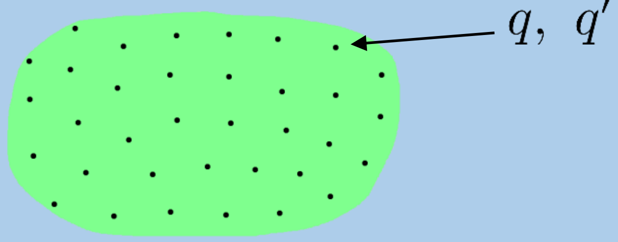
Proving the LRIP

1 : Proving non-uniform LRIP



Proving the LRIP

1 : Proving non-uniform LRIP



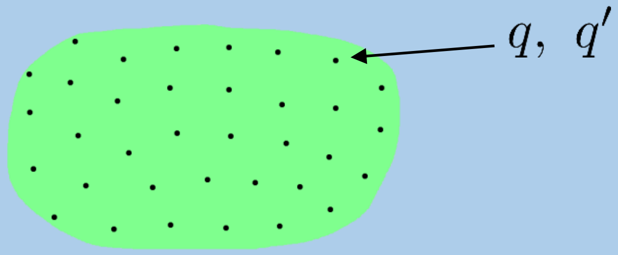
$$\left.\begin{array}{r}\text{Kernel mean embedding } \textit{[Smola 2007]} \\ \text{Random (Fourier) Features } \textit{[Rahimi 2007]}\end{array}\right\} \quad \|\mathcal{A}(q-q')\|_2^2 \approx \|q-q'\|_\kappa^2$$

$\downarrow$

Hoeffding, Bernstein, chaining…

Proving the LRIP

1 : Proving non-uniform LRIP

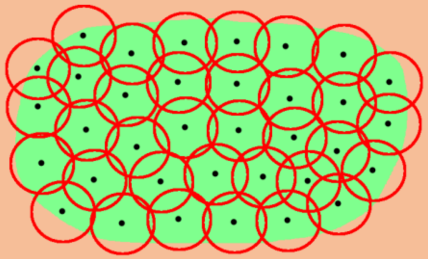


Kernel mean embedding *[Smola 2007]*
Random (Fourier) Features *[Rahimi 2007]*

$$\|\mathcal{A}(q-q')\|_2^2 \approx \|q-q'\|_\kappa^2$$

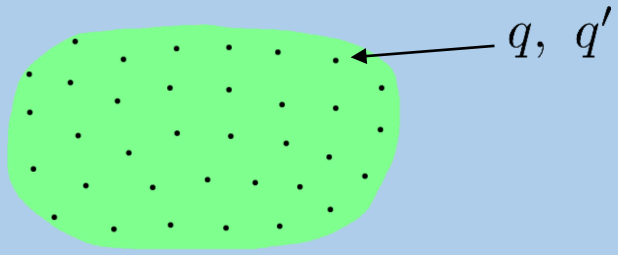
Hoeffding, Bernstein, chaining…

2 : Use ε - coverings to extend to uniform LRIP



Proving the LRIP

1 : Proving non-uniform LRIP



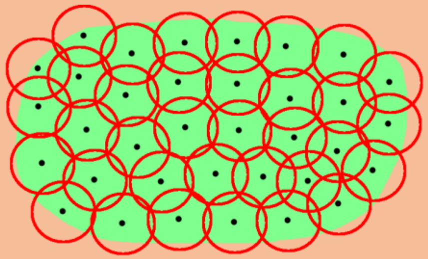
Kernel mean embedding *[Smola 2007]*
Random (Fourier) Features *[Rahimi 2007]*

$$\|\mathcal{A}(q-q')\|_2^2 \approx \|q-q'\|_\kappa^2$$

$\downarrow$

Hoeffding, Bernstein, chaining…

2 : Use ε - coverings to extend to uniform LRIP



Basic Set

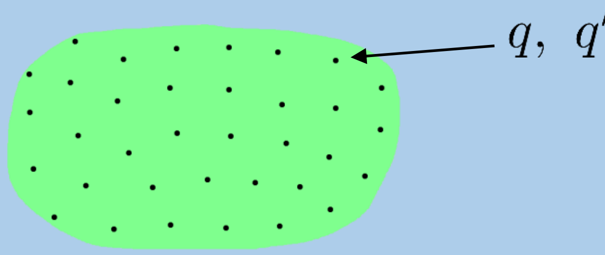
$$\mathfrak{S} \text{ for } \eta > 0$$

Easy !

Ex : Quantization error
[Boufounos 2016]

Proving the LRIP

1 : Proving non-uniform LRIP



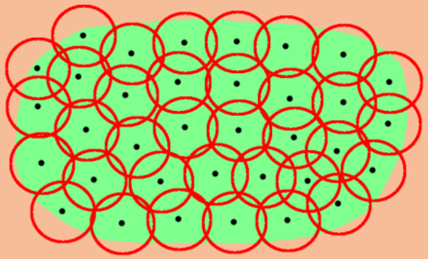
Kernel mean embedding *[Smola 2007]*
Random (Fourier) Features *[Rahimi 2007]*

$$\|\mathcal{A}(q-q')\|_2^2 \approx \|q-q'\|_\kappa^2$$

$\downarrow$

Hoeffding, Bernstein, chaining…

2 : Use ε - coverings to extend to uniform LRIP



Basic Set

$\mathfrak{S}$ for $\eta > 0$

Easy !

Ex : Quantization error
[Boufounos 2016]

Normalized Secant Set

$\left\{ \frac{q-q'}{\|q-q'\|_\kappa} \right\}$ for $\eta = 0$

- Finite-dimensional **Easy !**
- Infinite-dimensional **Difficult !**

Results

Sufficient Conditions

- $\mathfrak{S}$ has finite covering numbers

$$\eta = \mathcal{O}(1/\sqrt{m})$$

Bad !

Ex : GMMs with unknown covariance

Results

Sufficient Conditions

- $\mathfrak{S}$ has finite covering numbers
- $\mathfrak{S}$ mixtures of sufficiently separated distributions
- $\kappa(p_\theta, p_{\theta'}) = f(\|\theta - \theta'\|)$ with smooth f
- « Smooth » Random Features
- Smooth risk R

$$\eta = \mathcal{O}(1/\sqrt{m})$$

Bad !

Ex : GMMs with unknown covariance

$$\eta = \mathcal{O}(C^{-m})$$

+ guarantees w.r.t. risk

Ex : Mixture of Diracs (K-means) with
$m \geq \mathcal{O}\left(K^2 n^2 \texttt{polylog}(K, n) \log(1/\eta)\right)$

Results

Sufficient Conditions

- $\mathfrak{S}$ has finite covering numbers
- $\mathfrak{S}$ mixtures of sufficiently separated distributions
- $\kappa(p_\theta, p_{\theta'}) = f(\|\theta - \theta'\|)$ with smooth f
- « Smooth » Random Features
- Smooth risk R
- « Smoother » Random Features

$$\eta = \mathcal{O}(1/\sqrt{m})$$

Bad !

Ex : GMMs with unknown covariance

$$\eta = \mathcal{O}(C^{-m})$$

+ guarantees w.r.t. risk

Ex : Mixture of Diracs (K-means) with
$m \geq \mathcal{O}\left(K^2 n^2 \texttt{polylog}(K,n) \log(1/\eta)\right)$

$$\eta = 0$$

Ex :

- *Mixtures of Diracs (K-means) with*
  $m \geq \mathcal{O}\left(K^2 n^3 \texttt{polylog}(K,n)\right)$
- *GMMs with known covariance*

Outline

1. Introduction
2. Practical Approach
3. Results
4. Theoretical analysis
5. **Conclusion and outlooks**

Conclusions

Contributions

- **Greedy algorithm** for large-scale mixture learning from **random moments**

Conclusions

Contributions

- **Greedy algorithm** for large-scale mixture learning from **random moments**
- **Efficient heuristic** to design the sketching operator as **Fourier sampling**

Conclusions

Contributions

- **Greedy algorithm** for large-scale mixture learning from **random moments**
- **Efficient heuristic** to design the sketching operator as **Fourier sampling**
- Application to **mixtures of Diracs, GMMs**

Conclusions

Contributions

- **Greedy algorithm** for large-scale mixture learning from **random moments**
- **Efficient heuristic** to design the sketching operator as **Fourier sampling**
- Application to **mixtures of Diracs, GMMs**
- Evaluation on **synthetic** and **real** data

Conclusions

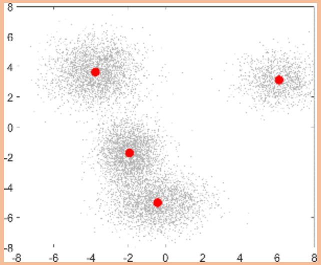
Contributions

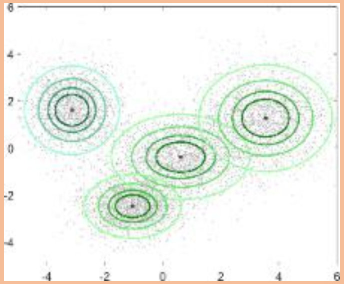
- **Greedy algorithm** for large-scale mixture learning from **random moments**
- **Efficient heuristic** to design the sketching operator as **Fourier sampling**
- Application to **mixtures of Diracs, GMMs**
- Evaluation on **synthetic** and **real** data
- **Information preservation guarantees** using infinite-dimensional Compressive Sensing

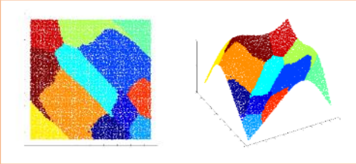
The SketchMLbox

SketchMLbox (`sketchml.gforge.inria.fr`)

- Mixture of Diracs (« K-means »)
- GMMs with known covariance
- GMMs with unknown diagonal covariance
- **Soon**:
  - Alpha-stable
  - Gaussian Locally Linear Mapping *[Deleforge 2014]*
- **Optimized for user-defined** $(\mathcal{A}p_\theta,\ \nabla_\theta \mathcal{A}p_\theta)$







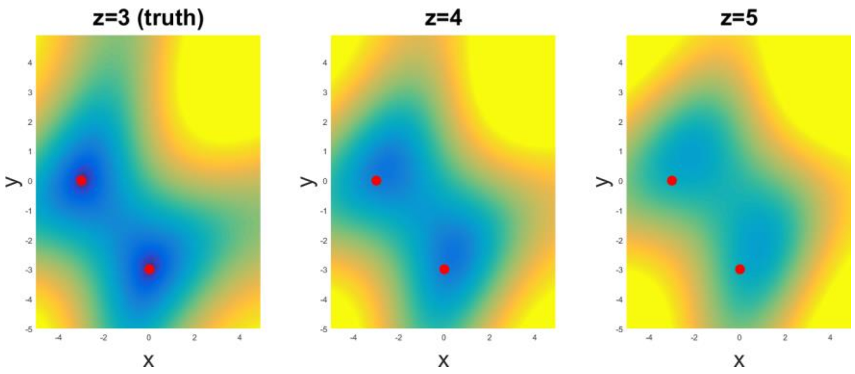
Outlooks : algorithmic guarantees

Algorithmic guarantees ? Non-convex cost function, randomized algorithm…

Outlooks : algorithmic guarantees

Algorithmic guarantees ? Non-convex cost function, randomized algorithm…

- Locally convex ?

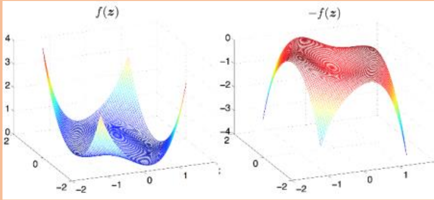


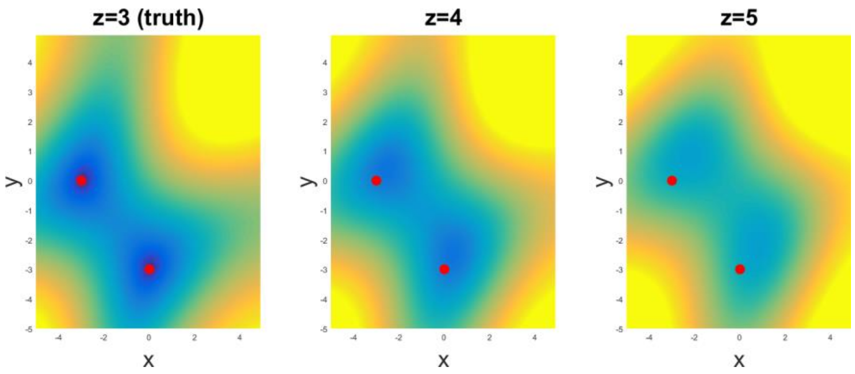
Cost function for a 3-GMM in dimension 1 at positions {x,y,z}={-3,0,3}

Outlooks : algorithmic guarantees

Algorithmic guarantees ? Non-convex cost function, randomized algorithm…

- Locally convex ?
- Basin of attraction ? *[Jacques 2016, Candes 2016…]*



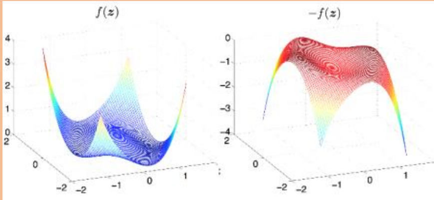


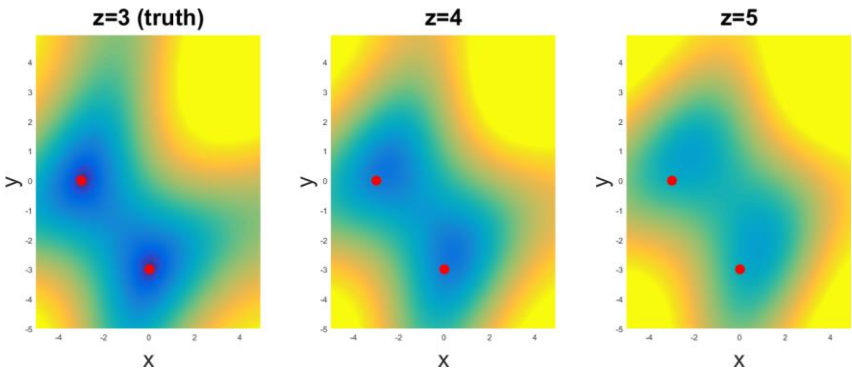
Cost function for a 3-GMM in dimension 1 at positions {x,y,z}={-3,0,3}

Outlooks : algorithmic guarantees

Algorithmic guarantees ? Non-convex cost function, randomized algorithm…

- Locally convex ?
- Basin of attraction ? *[Jacques 2016, Candes 2016…]*
- Reached by CLOMPR with reasonable hypotheses ?



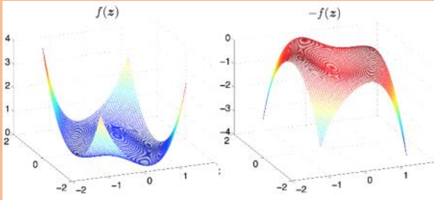


Cost function for a 3-GMM in dimension 1 at positions {x,y,z}={-3,0,3}

Outlooks : algorithmic guarantees

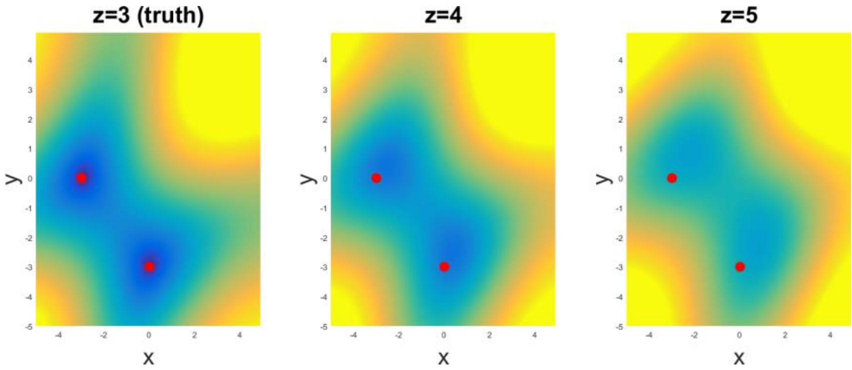
Algorithmic guarantees ? Non-convex cost function, randomized algorithm…

- Locally convex ?
- Basin of attraction ? *[Jacques 2016, Candes 2016…]*



- Reached by CLOMPR with reasonable hypotheses ?
- Stopping condition ?

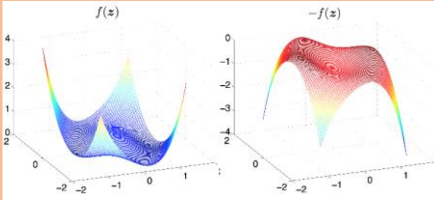
Cost function for a 3-GMM in dimension 1 at positions {x,y,z}={-3,0,3}



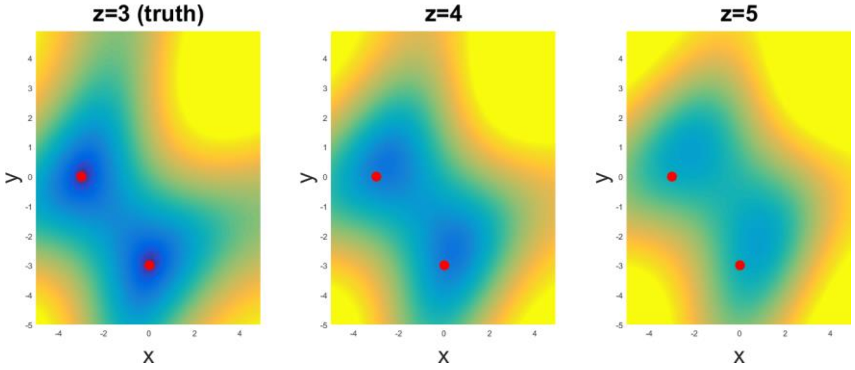
Outlooks : algorithmic guarantees

Algorithmic guarantees ? Non-convex cost function, randomized algorithm…

- Locally convex ?
- Basin of attraction ? *[Jacques 2016, Candes 2016…]*
- Reached by CLOMPR with reasonable hypotheses ?
- Stopping condition ?



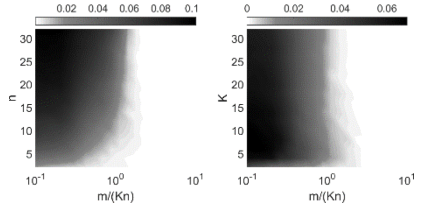
Recent result : locally block convex



Cost function for a 3-GMM in dimension 1 at positions {x,y,z}={-3,0,3}

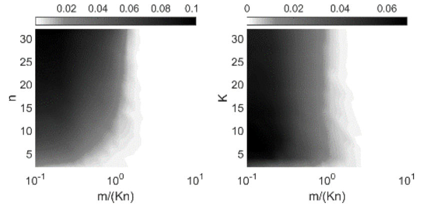
Outlooks : extension of the methods

1. Bridge observed gap between theory and practice ?



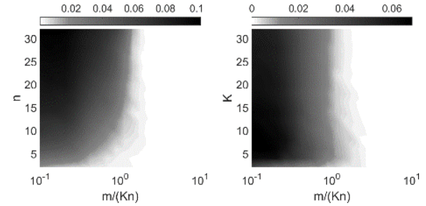
Outlooks : extension of the methods

1. Bridge observed gap between theory and practice ?
   - Does *not* come from ε - coverings



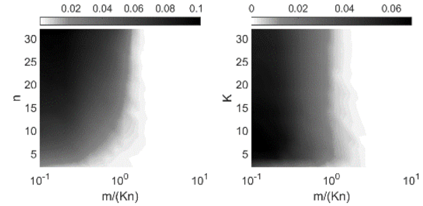
Outlooks : extension of the methods

1. Bridge observed gap between theory and practice ?
   - Does *not* come from $\mathcal{E}$ - coverings
   - Improve concentration inequalities ?



Outlooks : extension of the methods

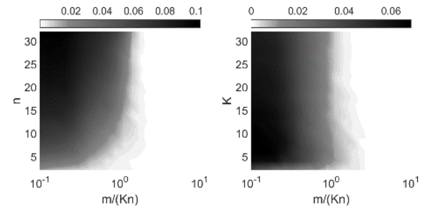
1. Bridge observed gap between theory and practice ?
   - Does *not* come from ε - coverings
   - Improve concentration inequalities ?



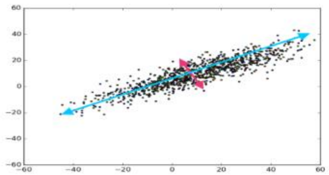
2. Extend framework to other tasks ?

Outlooks : extension of the methods

1. Bridge observed gap between theory and practice ?
   - Does *not* come from ε - coverings
   - Improve concentration inequalities ?

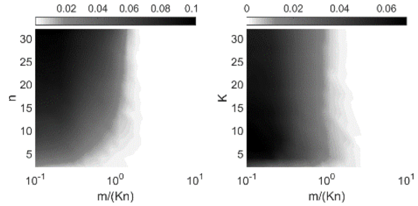


2. Extend framework to other tasks ?
   - Recent paper submitted to AISTATS : **PCA**

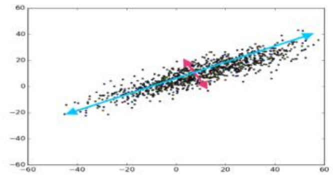


Outlooks : extension of the methods

1. Bridge observed gap between theory and practice ?
   - Does *not* come from $\mathcal{E}$ - coverings
   - Improve concentration inequalities ?

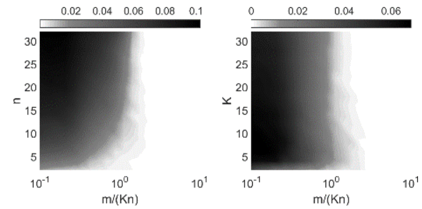


2. Extend framework to other tasks ?
   - Recent paper submitted to AISTATS : **PCA**
   - Other existing use of Fourier sketches ? : e.g. **classification** *[Sutherland 2015]*

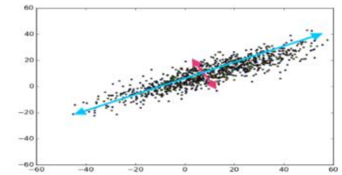


Outlooks : extension of the methods

1. Bridge observed gap between theory and practice ?
   - Does *not* come from ε - coverings
   - Improve concentration inequalities ?



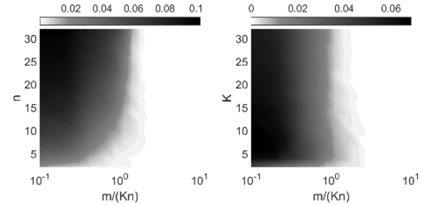
2. Extend framework to other tasks ?
   - Recent paper submitted to AISTATS : **PCA**



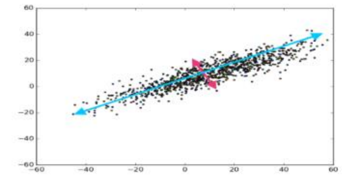
   - Other existing use of Fourier sketches ? : e.g. **classification** *[Sutherland 2015]*
   - Other kernel methods (algorithmic ? Theoretical ?)

Outlooks : extension of the methods

1. Bridge observed gap between theory and practice ?
   - Does *not* come from ε - coverings
   - Improve concentration inequalities ?



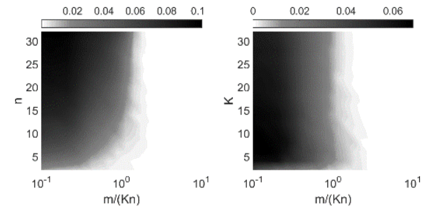
2. Extend framework to other tasks ?
   - Recent paper submitted to AISTATS : **PCA**
   - Other existing use of Fourier sketches ? : e.g. **classification** *[Sutherland 2015]*
   - Other kernel methods (algorithmic ? Theoretical ?)

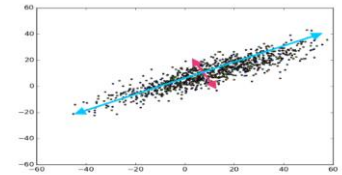


3. Extension to multi-layer sketches ? (Neural networks…)

Outlooks : extension of the methods

1. Bridge observed gap between theory and practice ?
   - Does *not* come from ε - coverings
   - Improve concentration inequalities ?



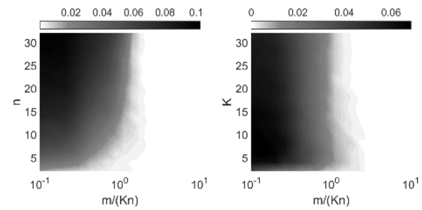


2. Extend framework to other tasks ?
   - Recent paper submitted to AISTATS : **PCA**
   - Other existing use of Fourier sketches ? : e.g. **classification** *[Sutherland 2015]*
   - Other kernel methods (algorithmic ? Theoretical ?)

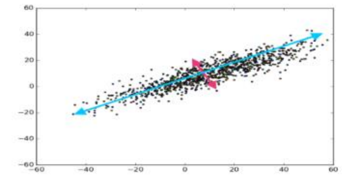
3. Extension to multi-layer sketches ? (Neural networks…)
   - May be adapted to e.g. GMMs with unknown covariance

Outlooks : extension of the methods

1. Bridge observed gap between theory and practice ?
   - Does *not* come from ε - coverings
   - Improve concentration inequalities ?



2. Extend framework to other tasks ?

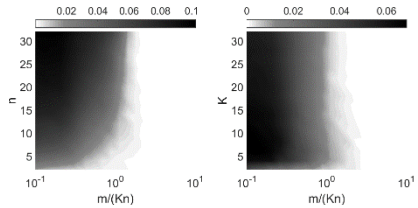


   - Recent paper submitted to AISTATS : **PCA**
   - Other existing use of Fourier sketches ? : e.g. **classification** *[Sutherland 2015]*
   - Other kernel methods (algorithmic ? Theoretical ?)

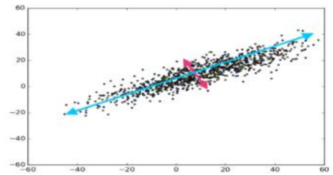
3. Extension to multi-layer sketches ? (Neural networks…)
   - May be adapted to e.g. GMMs with unknown covariance
   - Equivalence between LRIP and instance optimality still valid for **non-linear operators** !

Outlooks : extension of the methods

1. Bridge observed gap between theory and practice ?
   - Does *not* come from ε - coverings
   - Improve concentration inequalities ?



2. Extend framework to other tasks ?
   - Recent paper submitted to AISTATS : **PCA**
   - Other existing use of Fourier sketches ? : e.g. **classification** *[Sutherland 2015]*
   - Other kernel methods (algorithmic ? Theoretical ?)



3. Extension to multi-layer sketches ? (Neural networks…)
   - May be adapted to e.g. GMMs with unknown covariance
   - Equivalence between LRIP and instance optimality still valid for **non-linear operators** !
   - CLOMPR and current sufficient conditions no longer valid…

Thank you !

- K., Bourrier, Gribonval, Perez. **Sketching for Large-Scale Learning of Mixture Models** *ICASSP 2016*
- K., Bourrier, Gribonval, Perez. **Sketching for Large-Scale Learning of Mixture Models** (extended version) *submitted to Information and Inference, arXiv:1606.0238*
- K., Tremblay, Gribonval, Traonmilin. **Compressive K-means** *ICASSP 2017*
- Gribonval, Blanchard, K., Traonmilin. **Random moments for Sketched Statistical Learning** *submitted to AISTATS 2017, extended version soon*

Appendix : CLOMPR

Algorithm 2: Compressive mixture learning *à la* OMP: CLOMP ($T = K$) and CLOMPR ($T = 2K$)

Data: Empirical sketch $\hat{\mathbf{z}}$, sketching operator $\mathcal{A}$, sparsity K, number of iterations $T \geq K$

Result: Support Θ, weights $\boldsymbol{\alpha}$

$\hat{\mathbf{r}} \leftarrow \hat{\mathbf{z}}$; $\Theta \leftarrow \emptyset$;

for $t \leftarrow 1$ **to** T **do**

- **Step 1**: Find a normalized atom highly correlated with the residual with a gradient descent
  - $\boldsymbol{\theta} \leftarrow \texttt{maximize}_{\boldsymbol{\theta}} \left(\mathrm{Re} \left\langle \frac{\mathcal{A}P_{\boldsymbol{\theta}}}{\|\mathcal{A}P_{\boldsymbol{\theta}}\|_2}, \hat{\mathbf{r}} \right\rangle_2, \texttt{init} = \texttt{rand} \right)$;
- **end**
- **Step 2**: Expand support
  - $\Theta \leftarrow \Theta \cup \{\boldsymbol{\theta}\}$;
- **end**
- **Step 3**: Enforce sparsity by Hard Thresholding if needed
  - **if** $|\Theta| > K$ **then**
    - $\boldsymbol{\beta} \leftarrow \arg\min_{\boldsymbol{\beta} \geq 0} \left\| \hat{\mathbf{z}} - \sum_{k=1}^{|\Theta|} \beta_k \frac{\mathcal{A}P_{\boldsymbol{\theta}_k}}{\|\mathcal{A}P_{\boldsymbol{\theta}_k}\|_2} \right\|_2$ Select K largest entries $\beta_{i_1}, ..., \beta_{i_K}$;
    - Reduce the support $\Theta \leftarrow \{\boldsymbol{\theta}_{i_1}, ..., \boldsymbol{\theta}_{i_K}\}$;
  - **end**
- **end**
- **Step 4**: Project to find weights
  - $\boldsymbol{\alpha} \leftarrow \arg\min_{\boldsymbol{\alpha} \geq 0} \left\| \hat{\mathbf{z}} - \sum_{k=1}^{|\Theta|} \alpha_k \mathcal{A}P_{\boldsymbol{\theta}_k} \right\|_2$;
- **end**
- **Step 5**: Perform a gradient descent *initialized with current parameters*
  - $\Theta, \boldsymbol{\alpha} \leftarrow \texttt{minimize}_{\Theta, \boldsymbol{\alpha}} \left(\left\| \hat{\mathbf{z}} - \sum_{k=1}^{|\Theta|} \alpha_k \mathcal{A}P_{\boldsymbol{\theta}_k} \right\|_2, \texttt{init} = (\Theta, \boldsymbol{\alpha}), \texttt{constraint} = \{\boldsymbol{\alpha} \geq 0\} \right)$;
- **end**
- Update residual: $\hat{\mathbf{r}} \leftarrow \hat{\mathbf{z}} - \sum_{k=1}^{|\Theta|} \alpha_k \mathcal{A}P_{\boldsymbol{\theta}_k}$

end

Normalize $\boldsymbol{\alpha}$ such that $\sum_{k=1}^{K} \alpha_k = 1$