Sketching for Large-Scale Learning of Mixture Models

N. Keriven[⋆§] A. Bourrier[†] R. Gribonval[§] P. Pérez[‡]

[⋆] Université Rennes 1, France
[§] INRIA Rennes-Bretagne Atlantique, France
[†] Gipsa-Lab, St-Martin-d'Hères, France
[‡] Technicolor, Cesson Sévigné, France

GdR ISIS, 9 Juin 2016

INVENTEURS DU MONDE NUMÉRIQUE

Outline

1. Introduction
2. Proposed Algorithm
3. Sketching GMM
4. Results
5. Theoretical guarantees ?
6. Conclusion

Paths to Compressive Learning

Objective

Learn parameters Θ from a **large** database $(\mathbf{x}_1, ..., \mathbf{x}_N) \in \mathbb{R}^n$.

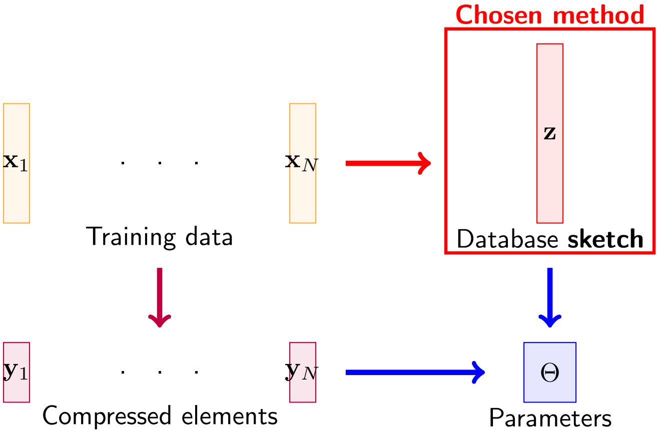
Examples:

- Learn subspace V_Θ of principal components
- Learn parameters of a classifier f_Θ
- Fit a probability distribution p_Θ
- ...

Paths to Compressive Learning

Objective

Learn parameters Θ from a **large** database $(\mathbf{x}_1, ..., \mathbf{x}_N) \in \mathbb{R}^n$.



In this talk

In this talk

Efficient method for Gaussian Mixture Model (GMM) estimation from a sketch.

Example :

Estimation of a 20-GMM from a database of $N = 10^6$ vectors in $\mathbb{R}^{10}$

- 5000-fold compression of the database
  - Can be performed efficiently on GPU / clusters
- Estimation process $70\times$ faster than EM
- Same precision than EM in the result

Approach : Generalized Compressive Sensing

Traditional Compressive Sensing (CS)

From $\mathbf{y} \approx \mathbf{M}\mathbf{x} \in \mathbb{R}^m$ recover vector $\mathbf{x} \in \mathbb{R}^n$

- Linear $\mathbf{M} \in \mathbb{R}^{m \times n}$ with $m < n$
- Typical assumption: $\mathbf{x}$ sparse, etc.

Generalized Compressive Sensing

From $\mathbf{z} \approx \mathcal{A}p \in \mathbb{C}^m$ recover probability distribution $p \in L^1(\mathbb{R}^n)$

Must define:

- Linear operator $\mathcal{A} : L^1(\mathbb{R}^n) \mapsto \mathbb{C}^m$
- Generalized "sparsity" in $L^1(\mathbb{R}^n)$

Sparse probability distributions: Mixture Models

- K-sparse vectors: combination of K "basic" elements
- "K-sparse" probability distributions :

$$p_{\Theta,\boldsymbol{\alpha}} = \sum_{k=1}^{K} \alpha_k p_{\boldsymbol{\theta}_k}$$

  - with $\boldsymbol{\alpha} \geq 0$; $\sum_k \alpha_k = 1$; $p_{\boldsymbol{\theta}_k} \in \{p_{\boldsymbol{\theta}}; \boldsymbol{\theta} \in \mathcal{T}\}$
- Sketch $\mathbf{z} = \sum_{k=1}^{K} \alpha_k \mathcal{A} p_{\boldsymbol{\theta}_k}$ as a combination of atoms in the dictionary:

$$\mathcal{D} = \{\mathcal{A} p_{\boldsymbol{\theta}};\ \boldsymbol{\theta} \in \mathcal{T}\}$$

Challenge

Possibly infinite / continuous dictionary $\mathcal{D}$.

Application to Compressive Learning

From theoretical Generalized CS...

$$p \xrightarrow{\mathcal{A}} \mathbf{z} = \mathcal{A}p \xrightarrow{Alg.} p_{\Theta,\boldsymbol{\alpha}}$$

...to practical Compressive Learning:

$$\hat{p} = \frac{1}{N}\sum_i \delta_{\mathbf{x}_i} \xrightarrow{\mathcal{A}} \hat{\mathbf{z}} = \mathcal{A}\hat{p} \xrightarrow{Alg.} p_{\hat{\Theta},\hat{\boldsymbol{\alpha}}}$$

where $(\mathbf{x}_1, ..., \mathbf{x}_N) \overset{i.i.d.}{\sim} p$.

Questions:

- Reconstruction algorithm ? (Part 2)
- Choice of sketching operator $\mathcal{A}$? (Part 3)
- Empirically/theoretically valid ? (Parts 4 and 5)

Outline

1. Introduction
2. Proposed Algorithm
3. Sketching GMM
4. Results
5. Theoretical guarantees ?
6. Conclusion

Approach

$$p \xrightarrow{\mathcal{A}} \mathbf{z} = \mathcal{A}p \xrightarrow{\text{Alg.}} p_{\Theta,\boldsymbol{\alpha}}$$

Cost function

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

- Similar to $\min_{\mathbf{x}:\|\mathbf{x}\|_0 \leq s} \|\mathbf{y} - \mathbf{M}\mathbf{x}\|_2$ in CS.

- **Pros:** Under some hypothesis on $\mathcal{G}$ and $\mathcal{A}$, yields provably good solutions with high probability (Section 5)
- **Cons:** Generally highly non-convex / intractable
  - Convex relaxation (Bunea 2010): seems difficult because of infinite / continuous dictionary
  - Greedy approaches: approach retained here

Orthogonal Matching Pursuit with Replacement

- OMP: add an atom to the support by maximizing its correlation to the residual, update the residual, repeat.

Orthogonal Matching Pursuit with Replacement

- OMP
- **OMP with Replacement** (Jain 2011)
  - More iterations than OMP, Hard Thresholding step.

Similar to CoSAMP or Subspace Pursuit.

Orthogonal Matching Pursuit with Replacement

- OMP
- **OMP with Replacement** (Jain 2011)
  - More iterations than OMP, Hard Thresholding step.
- **Compressive Learning OMPR** *(proposed)*
  - Non-negativity on weights α
  - Continuous dictionary $\longrightarrow$ gradient descents
  - Add a global optimization step.

Orthogonal Matching Pursuit with Replacement

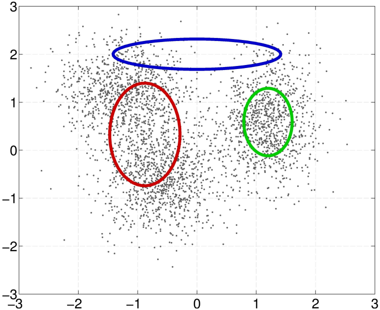
- OMP
- **OMP with Replacement** (Jain 2011)
  - More iterations than OMP, Hard Thresholding step.
- **Compressive Learning OMPR** *(proposed)*
  - Non-negativity on weights α
  - Continuous dictionary $\longrightarrow$ gradient descents
  - Add a global optimization step.

Number of iterations	Compressive Sensing	Compressive Learning
K	OMP	CLOMP
$2K$	OMPR	CLOMPR

Compressive Learning OMPR

Example : iteration 4 of CLOMPR, searching for a 3-GMM

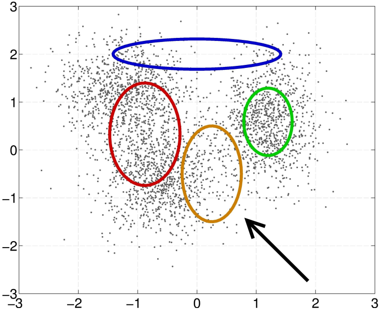
- Current support



Compressive Learning OMPR

Example : iteration 4 of CLOMPR, searching for a 3-GMM

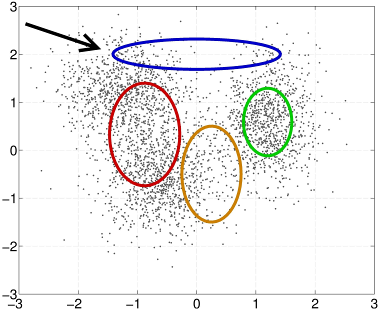
- Add an atom to the support with a gradient descent: $\arg\max_{\boldsymbol{\theta}} Re \left\langle \mathbf{r}, \frac{\mathcal{A}p_{\boldsymbol{\theta}}}{\|\mathcal{A}p_{\boldsymbol{\theta}}\|_2} \right\rangle$



Compressive Learning OMPR

Example : iteration 4 of CLOMPR, searching for a 3-GMM

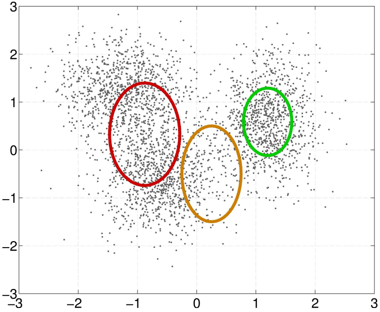
- **Hard Thresholding** to reduce the support



Compressive Learning OMPR

Example : iteration 4 of CLOMPR, searching for a 3-GMM

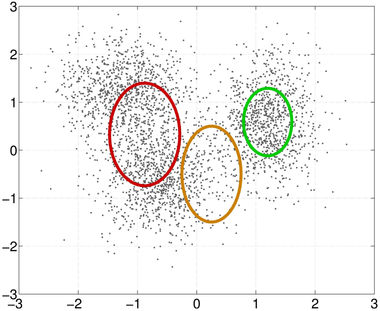
- **Hard Thresholding** to reduce the support



Compressive Learning OMPR

Example : iteration 4 of CLOMPR, searching for a 3-GMM

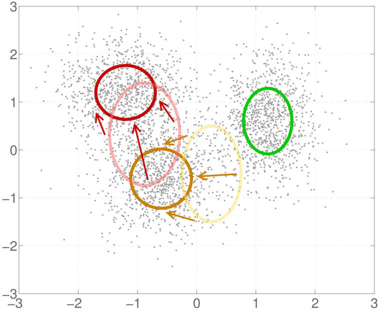
- **Hard Thresholding** to reduce the support
- Solve a **Non-negative** Least Squares to find the weights $\boldsymbol{\alpha}$.



Compressive Learning OMPR

Example : iteration 4 of CLOMPR, searching for a 3-GMM

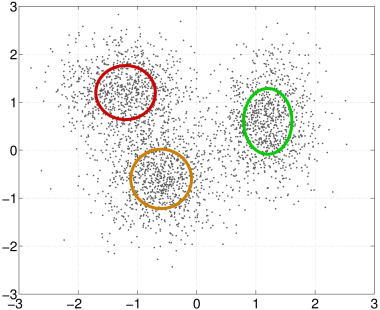
- **New step:** **global gradient descent** initialized with the current parameters to further reduce $\|\mathbf{z} - \mathcal{A}p_{\Theta,\boldsymbol{\alpha}}\|_2$



Compressive Learning OMPR

Example : iteration 4 of CLOMPR, searching for a 3-GMM

- **New step:** global gradient descent initialized with the current parameters to further reduce $\|\mathbf{z} - \mathcal{A}p_{\Theta,\boldsymbol{\alpha}}\|_2$
- Update residual.



What is left ?

$$p \xrightarrow{\mathcal{A}} \mathbf{z} = \mathcal{A}p \xrightarrow{CLOMP(R)} p_{\Theta,\boldsymbol{\alpha}} = \sum_k \alpha_k p_{\boldsymbol{\theta}_k}$$

To perform CLOMP(R), $\mathcal{A}p_{\boldsymbol{\theta}}$ and $\nabla_{\boldsymbol{\theta}}\mathcal{A}p_{\boldsymbol{\theta}}$ must have a closed-form expression.

- Here:
  - GMMs with diagonal covariance
- Soon-to-be-released toolbox:
  - K-means
  - full GMMs
  - Gaussian regression
  - α-stable (in progress)
  - User-defined !

Outline

1. Introduction
2. Proposed Algorithm
3. Sketching GMM
4. Results
5. Theoretical guarantees ?
6. Conclusion

Model: Gaussian mixture

$$p \xrightarrow{\mathcal{A}} \mathbf{z} = \mathcal{A}p \xrightarrow{CLOMP(R)} p_{\Theta,\boldsymbol{\alpha}} = \sum_k \alpha_k p_{\boldsymbol{\theta}_k}$$

Gaussian Mixture Model

$$p_{\boldsymbol{\theta}} = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \text{ with diagonal } \boldsymbol{\Sigma}$$

Sketching operator

$$p \xrightarrow{\mathcal{A}} \mathbf{z} = \mathcal{A}p \xrightarrow{CLOMP(R)} p_{\Theta,\boldsymbol{\alpha}}$$

Random Sampling of the characteristic function (Bourrier 2013)

Denote $\psi_p(\boldsymbol{\omega}) = \mathbb{E}_{\mathbf{x}\sim p}(e^{\mathrm{i}\boldsymbol{\omega}^T\mathbf{x}})$. Given $(\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_m) \in \mathbb{R}^n$, define

$$\mathcal{A}p = \frac{1}{\sqrt{m}}\left[\psi_p(\boldsymbol{\omega}_j)\right]_{j=1,\ldots,m}$$

- Closed-form for GMMs
- Analog to Random Fourier Sampling: $(\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_m) \overset{i.i.d.}{\sim} \Lambda$
- $\hat{\mathbf{z}} = \frac{1}{\sqrt{m}}\left[\frac{1}{N}\sum_i e^{\mathrm{i}\boldsymbol{\omega}_j^T\mathbf{x}_i}\right]_{j=1,\ldots,m}$ easily computable (distributed, GPU, streaming...)

To summarize

$$\hat{p} \xrightarrow{\mathcal{A}} \hat{\mathbf{z}} = \mathcal{A}\hat{p} \xrightarrow{CLOMP(R)} p_{\Theta,\boldsymbol{\alpha}}$$

Given a database $\mathcal{X} = (\mathbf{x}_1, ..., \mathbf{x}_N) \in \mathbb{R}^n$, m, K:

- Design $\mathcal{A}$
  - Choose the frequency distribution Λ
  - Draw m frequencies $(\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_m) \in \mathbb{R}^n$
- Compute $\hat{\mathbf{z}} = \frac{1}{\sqrt{m}} \left[\frac{1}{N} \sum_i e^{\mathrm{i}\boldsymbol{\omega}_j^T \mathbf{x}_i} \right]_{j=1,...,m}$
  - GPU, distributed computing, etc.
- Throw away $\mathcal{X}$!
  - Privacy preserving
- Estimate a K-GMM $p_{\Theta,\boldsymbol{\alpha}}$ from $\hat{\mathbf{z}}$ using CLOMP(R).

To summarize

$$\hat{p} \xrightarrow{\mathcal{A}} \hat{\mathbf{z}} = \mathcal{A}\hat{p} \xrightarrow{CLOMP(R)} p_{\Theta,\boldsymbol{\alpha}}$$

Given a database $\mathcal{X} = (\mathbf{x}_1, ..., \mathbf{x}_N) \in \mathbb{R}^n$, m, K:

- Design $\mathcal{A}$
  - Choose the frequency distribution Λ
  - Draw m frequencies $(\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_m) \in \mathbb{R}^n$
- Compute $\hat{\mathbf{z}} = \frac{1}{\sqrt{m}} \left[\frac{1}{N} \sum_i e^{\mathrm{i}\boldsymbol{\omega}_j^T \mathbf{x}_i}\right]_{j=1,...,m}$
  - GPU, distributed computing, etc.
- Throw away $\mathcal{X}$!
  - Privacy preserving
- Estimate a K-GMM $p_{\Theta,\boldsymbol{\alpha}}$ from $\hat{\mathbf{z}}$ using CLOMP(R).

Designing the frequency distribution

The frequency distribution must "scale" with (the variances of) the GMM.

Approach 1 Optimize the variance of a Gaussian frequency distribution

- Ex : cross-validation with likelihood
- Classical choice (Sutherland 2015)

Designing the frequency distribution

The frequency distribution must "scale" with (the variances of) the GMM.

Approach 1 Optimize the variance of a Gaussian frequency distribution

Approach 2 Proposed:

- Partial preprocessing to compute the appropriate "scaling"
- Distribution that aims at maximizing $\|\nabla_{\boldsymbol{\theta}} \psi_{p_{\boldsymbol{\theta}}}\|_2$

The proposed distribution

- Yields better precision in the reconstruction
- Is $20\times$ to $100\times$ faster to design

To summarize (2)

$$\hat{p} \xrightarrow{\mathcal{X}\to\Lambda\to\mathcal{A}} \hat{\mathbf{z}} = \mathcal{A}\hat{p} \xrightarrow{CLOMP(R)} p_{\Theta,\boldsymbol{\alpha}}$$

Given a database $\mathcal{X} = (\mathbf{x}_1, ..., \mathbf{x}_N) \in \mathbb{R}^n$, m, K:

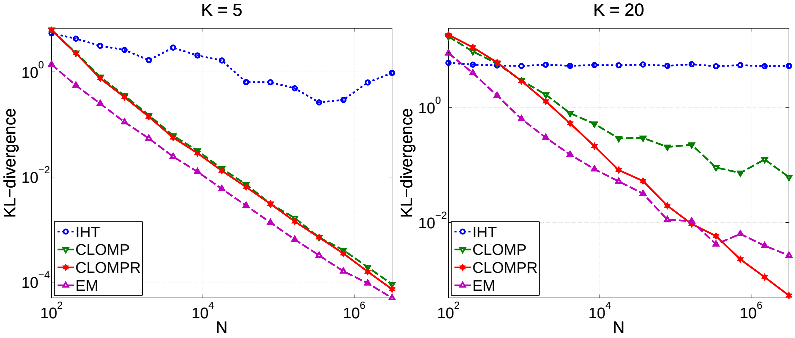
- Design $\mathcal{A}$
  - Partial preprocessing to choose the frequency distribution Λ
  - Draw m frequencies $(\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_m) \in \mathbb{R}^n$
- Compute $\hat{\mathbf{z}} = \frac{1}{\sqrt{m}} \left[\frac{1}{N} \sum_i e^{\mathrm{i}\boldsymbol{\omega}_j^T \mathbf{x}_i} \right]_{j=1,...,m}$
  - GPU, distributed computing, etc.
- Throw away $\mathcal{X}$!
  - Privacy preserving
- Estimate a K-GMM $p_{\Theta,\boldsymbol{\alpha}}$ from $\hat{\mathbf{z}}$ using CLOMP(R).

Outline

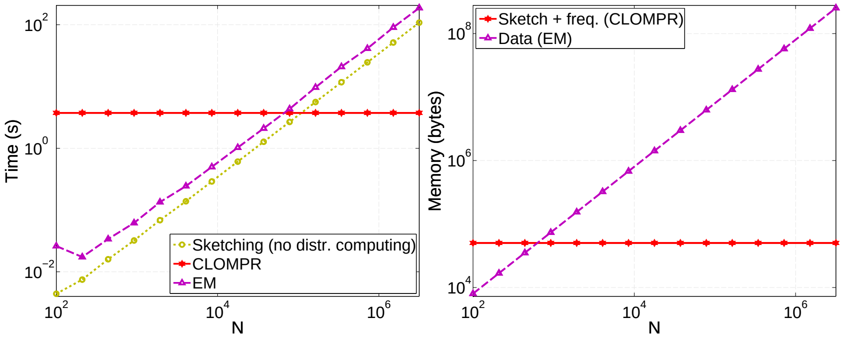
1. Introduction
2. Proposed Algorithm
3. Sketching GMM
4. Results
5. Theoretical guarantees ?
6. Conclusion

Reconstruction results

Comparison with EM (VLFeat toolbox) and previous Compressive Learning IHT (Bourrier 2013). KL-div (lower is better), $n = 10$, $m = 5(2n+1)K$.



Memory usage and computation time



- Remember : Sketching easily done on GPU/cluster

Application : speaker verification

- *NIST2005 database with MFCCs*
- *Classical method (Reynolds 2000), not state-of-the-art but serves as a proof of concept*

	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**

- A large database enhances the quality of the sketch
- Limitations are observed for large K : difficult "sparse approximation" task of a non-sparse distribution

Outline

1. Introduction
2. Proposed Algorithm
3. Sketching GMM
4. Results
5. Theoretical guarantees ?
6. Conclusion

Information preservation guarantees ?

$$p \xrightarrow{\mathcal{A}} \mathbf{z} = \mathcal{A}p \xrightarrow{CLOMP(R)} p_{\Theta,\boldsymbol{\alpha}}$$

- CLOMP(R) attempts to solve $\min_{\Theta,\boldsymbol{\alpha}} \|\mathbf{z} - \mathcal{A}p_{\Theta,\boldsymbol{\alpha}}\|_2$

Information preservation guarantees ?

$$p \xrightarrow{\mathcal{A}} \mathbf{z} = \mathcal{A}p \xrightarrow{CLOMP(R)} p_{\Theta,\boldsymbol{\alpha}}$$

- CLOMP(R) attempts to solve $\min_{\Theta,\boldsymbol{\alpha}} \|\mathbf{z} - \mathcal{A}p_{\Theta,\boldsymbol{\alpha}}\|_2$
  - Difficult to obtain guarantees for CLOMP(R): non-convex, random...

Information preservation guarantees ?

$$p \xrightarrow{\mathcal{A}} \mathbf{z} = \mathcal{A}p \xrightarrow{CLOMP(R)} p_{\Theta,\boldsymbol{\alpha}}$$

- CLOMP(R) attempts to solve $\min_{\Theta,\boldsymbol{\alpha}} \|\mathbf{z} - \mathcal{A}p_{\Theta,\boldsymbol{\alpha}}\|_2$
  - Difficult to obtain guarantees for CLOMP(R): non-convex, random...
- More fundamentally: if we **were** able to **exactly** solve

$$\min_{p \in \Sigma} \|\mathbf{z} - \mathcal{A}p\|_2,$$

with Σ "low-dimensional" set of distribution (e.g. K-sparse GMMs), do we have any guarantee ?

Information preservation guarantees ?

$$p \xrightarrow{\mathcal{A}} \mathbf{z} = \mathcal{A}p \xrightarrow{\text{Best algo. possible}} \bar{p} \in \arg\min_{p \in \Sigma} \|\mathbf{z} - \mathcal{A}p\|_2$$

Information preservation guarantees ?

$$p \xrightarrow{\mathcal{A}} \mathbf{z} = \mathcal{A}p \xrightarrow{\text{Best algo. possible}} \bar{p} \in \arg\min_{p \in \Sigma} \|\mathbf{z} - \mathcal{A}p\|_2$$

- Does $\mathbf{z}$ contains "enough" information to recover $p \in \Sigma$?

Information preservation guarantees ?

$$p \xrightarrow{\mathcal{A}} \mathbf{z} = \mathcal{A}p \xrightarrow{\text{Best algo. possible}} \bar{p} \in \arg\min_{p\in\Sigma} \|\mathbf{z} - \mathcal{A}p\|_2$$

- Does $\mathbf{z}$ contains "enough" information to recover $p \in \Sigma$?
- Is it stable if $p \notin \Sigma$?

Information preservation guarantees ?

$$p \xrightarrow{\mathcal{A}} \mathbf{z} = \mathcal{A}p \xrightarrow{\text{Best algo. possible}} \bar{p} \in \arg\min_{p\in\Sigma} \|\mathbf{z} - \mathcal{A}p\|_2$$

- Does $\mathbf{z}$ contains "enough" information to recover $p \in \Sigma$?
- Is it stable if $p \notin \Sigma$?
- Is it stable to use $\hat{\mathbf{z}}$ instead of $\mathbf{z}$?

Information preservation guarantees ? Yes !

$$\hat{p} \xrightarrow{\mathcal{A}} \hat{\mathbf{z}} = \mathcal{A}\hat{p} \xrightarrow{\text{Best algo. possible}} \bar{p} \in \arg\min_{p \in \Sigma} \|\hat{\mathbf{z}} - \mathcal{A}p\|_2$$

Main result

(under hypotheses on Σ and Λ)

- W.h.p. on $(\mathbf{x}_1, ..., \mathbf{x}_N) \overset{i.i.d.}{\sim} p^*$ and $(\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_m) \overset{i.i.d.}{\sim} \Lambda$,

$$\gamma_\Lambda(p^*, \bar{p}) \leq 5 d_{TV}(p^*, \Sigma) + \mathcal{O}\left(N^{-\frac{1}{2}}\right) + \eta,$$

  - γ_Λ "kernel" metric (Sriperumbudur 2010)
  - d_{TV} total variation distance between p^* and the model Σ
  - η additive error in m

Application to GMMs with compact set of parameters.

- $K = 1$ (toy):
  - $\eta = \mathcal{O}(\beta^{-m})$: Good !

Application to GMMs with compact set of parameters.

- $K = 1$ (toy):
  - $\eta = \mathcal{O}(\beta^{-m})$: Good !
- $K \geq 2$:
  - $\eta = \mathcal{O}\left(m^{-\frac{1}{2}}\right)$: Worst possible !

Application to GMMs with compact set of parameters.

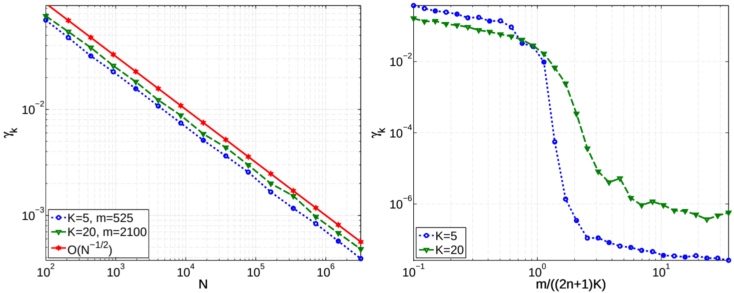
- $K = 1$ (toy):
  - $\eta = \mathcal{O}(\beta^{-m})$: Good !
- $K \geq 2$:
  - $\eta = \mathcal{O}\left(m^{-\frac{1}{2}}\right)$: Worst possible !
  - Global error in $\mathcal{O}\left(N^{-\frac{1}{2}} + m^{-\frac{1}{2}}\right)$: "compressive" approach ?

Application to GMMs with compact set of parameters.

- $K = 1$ (toy):
  - $\eta = \mathcal{O}(\beta^{-m})$: Good !
- $K \geq 2$:
  - $\eta = \mathcal{O}\left(m^{-\frac{1}{2}}\right)$: Worst possible !
  - Global error in $\mathcal{O}\left(N^{-\frac{1}{2}} + m^{-\frac{1}{2}}\right)$: "compressive" approach ?
  - **Conjecture**: it is in fact much better !

Application to GMMs with compact set of parameters.

- $K = 1$ (toy):
  - $\eta = \mathcal{O}(\beta^{-m})$: Good !
- $K \geq 2$:
  - $\eta = \mathcal{O}\left(m^{-\frac{1}{2}}\right)$: Worst possible !
  - Global error in $\mathcal{O}\left(N^{-\frac{1}{2}} + m^{-\frac{1}{2}}\right)$: "compressive" approach ?
  - **Conjecture**: it is in fact much better !



Outline

1. Introduction
2. Proposed Algorithm
3. Sketching GMM
4. Results
5. Theoretical guarantees ?
6. Conclusion

Conclusion

Summary

Effective method to learn GMMs from a sketch, using greedy algorithms and an efficient heuristic to design the sketching operator. Empirical and theoretical motivations.

In the journal paper

- Faster algorithm for GMM with large K
- More on theoretical guarantees

Future Work

- Application to other Mixture Models (K-means, α-stable...)
- Generalized theoretical guarantees
- Application to other kernel methods (Sutherland 2015) (classification...)

Questions ?

Keriven et al., **Sketching for Large-Scale Learning of Mixture Models**, *arXiv:1606.02838*