SPARS 2017

Signal Processing with Adaptive Sparse Structured Representations

Lisbon, Portugal **June 5-8, 2017**

Submission deadline: December 12, 2016

Notification of acceptance:	March 27, 2017
Summer School:	May 31-June 2, 2017 (tbc)
Workshop:	June 5-8, 2017

TÉCNICO LISBOA

instituto de telecomunicações

LISBOA UNIVERSIDADE DE LISBOA

SpaRTaN Sparse Representations and Compressed Sensing Training Network

MacSeNet Machine Sensing Training Network

spars2017.lx.it.pt

PLEASE

projection, learning and sparsity for efficient data processing

Random Moments for Compressive Learning

Rémi Gribonval
Inria Rennes - Bretagne Atlantique

remi.gribonval@inria.fr

erc

Main Contributors & Collaborators

- Anthony Bourrier

- Nicolas Keriven

- Yann Traonmilin

- Gilles Puy

- Nicolas Tremblay

- Gilles Blanchard

- Mike Davies

- Patrick Perez

Agenda

- From Compressive Sensing to Compressive Learning ?
- The Sketch Trick
- Compressive K-means
- Compressive GMM
- Conclusion

projection, learning and sparsity for efficient data processing

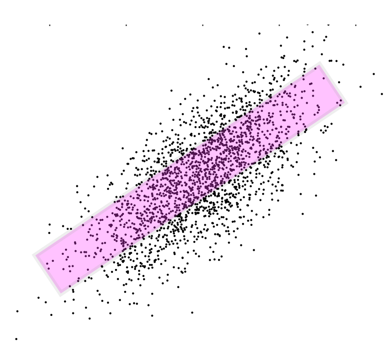
From Compressive Sensing to Compressive Learning

R. GRIBONVAL
London Workshop on Sparse Signal Processing, September 2016

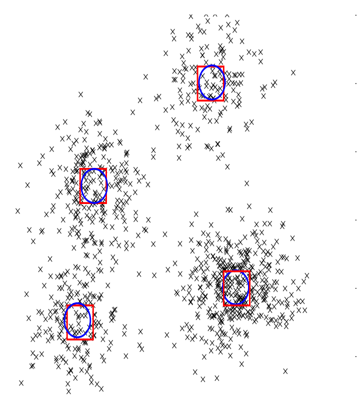
Machine Learning

- **Available data**
  - training collection of feature vectors = point cloud $\mathcal{X}$
- **Goals**
  - infer parameters to achieve a certain task
  - generalization to future samples with the same probability distribution

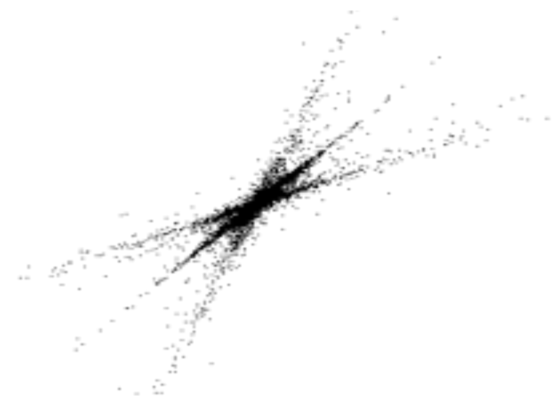
- **Examples**



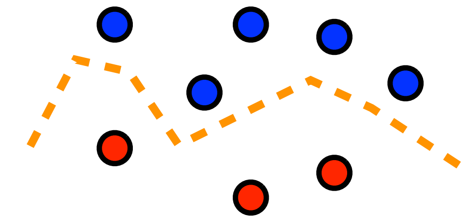
- PCA
  - principal subspace



- Clustering
  - centroids



- Dictionary learning
  - dictionary

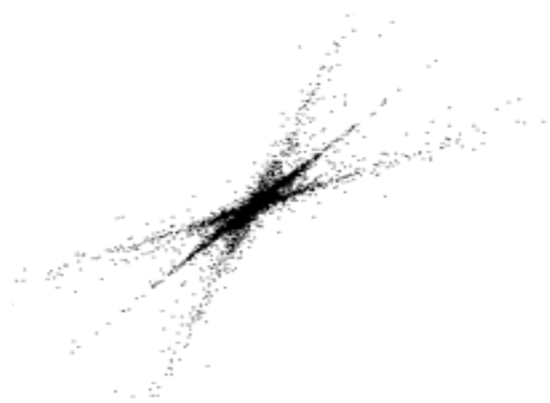


- Classification
  - classifier parameters (e.g. support vectors)

Challenging dimensions

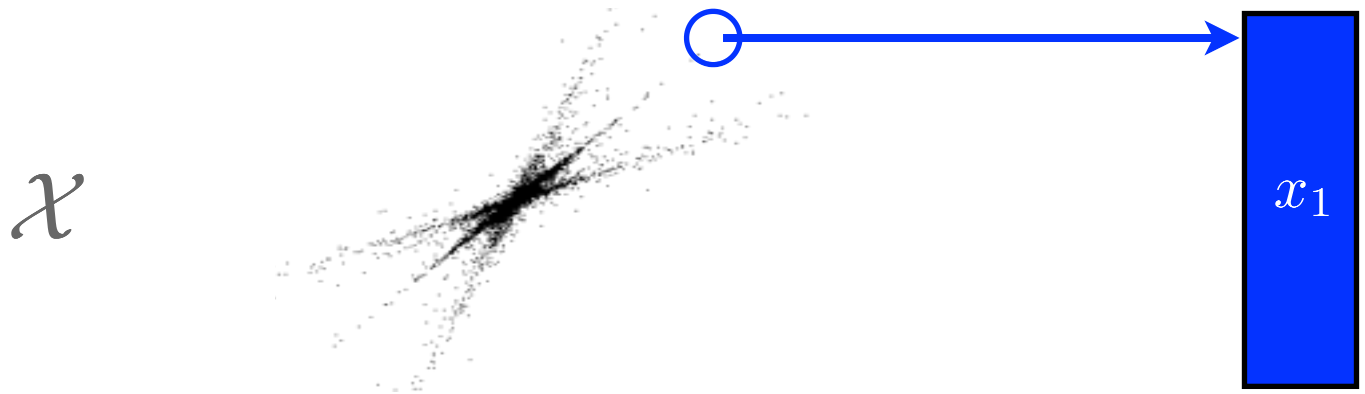
- Point cloud = large matrix of feature vectors

$\mathcal{X}$



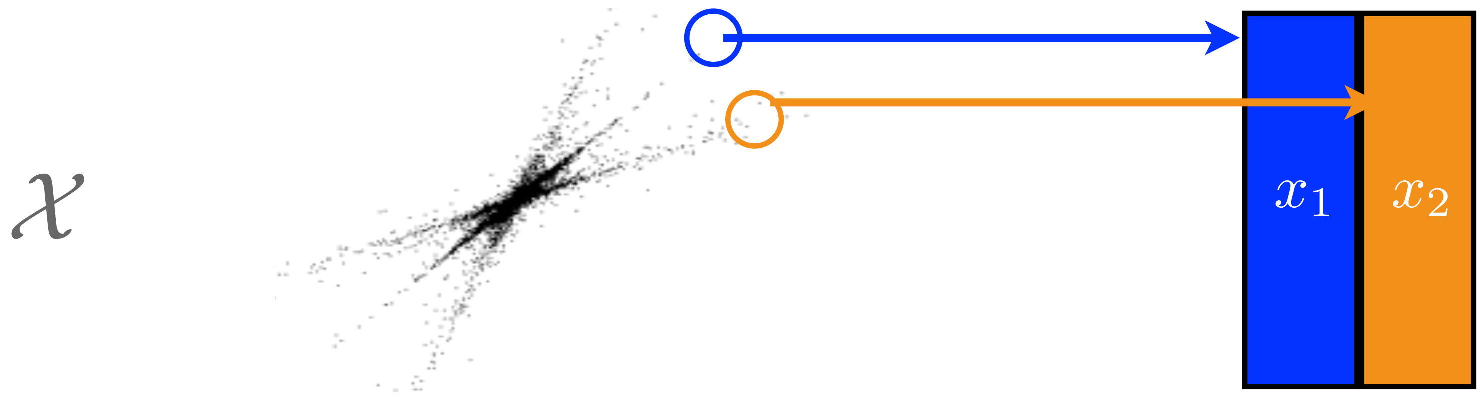
Challenging dimensions

- Point cloud = large matrix of feature vectors



Challenging dimensions

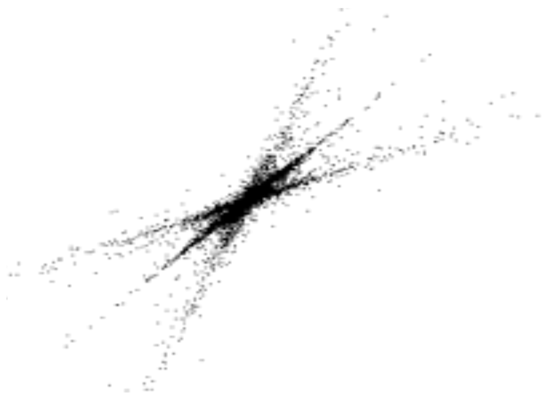
- Point cloud = large matrix of feature vectors



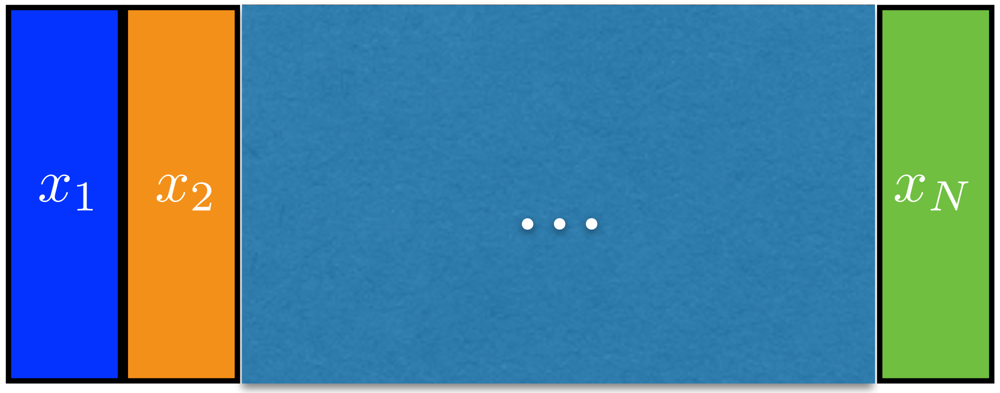
Challenging dimensions

- Point cloud = large matrix of feature vectors

$\mathcal{X}$



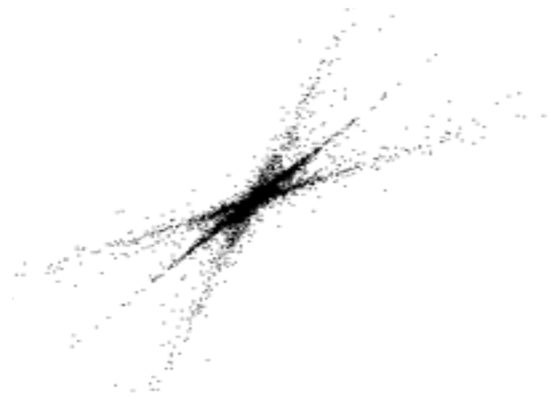
$\mathbf{X}$

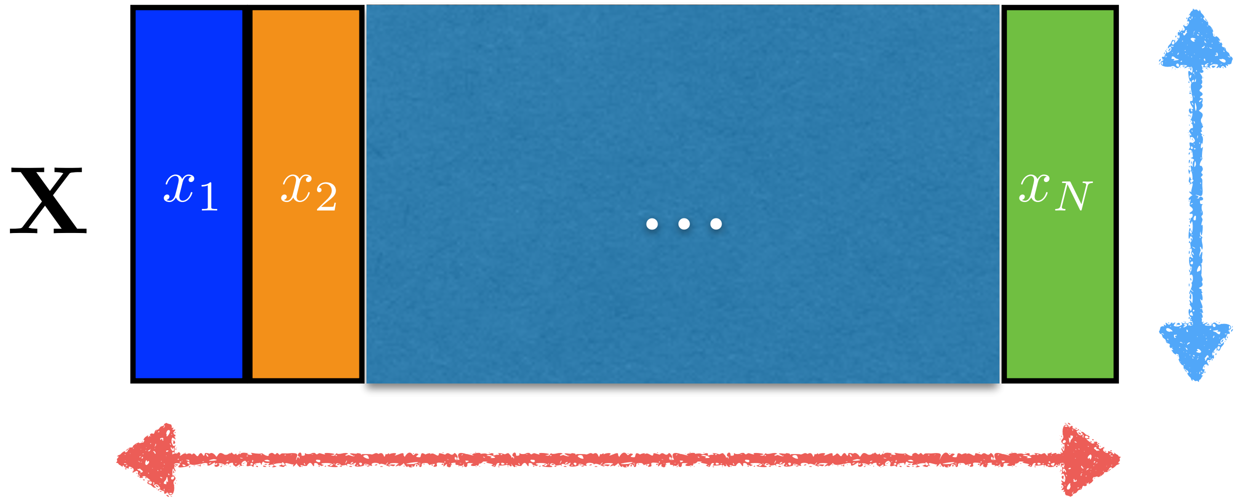


Challenging dimensions

- Point cloud = large matrix of feature vectors

$\mathcal{X}$



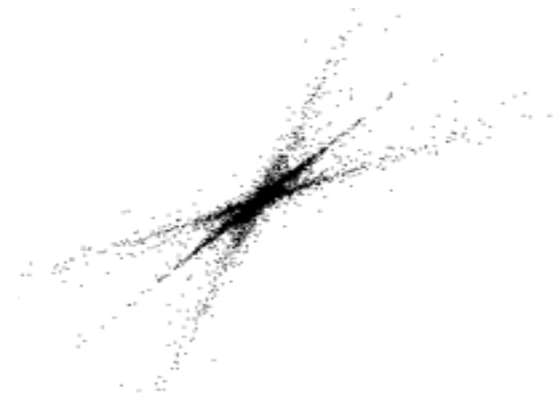


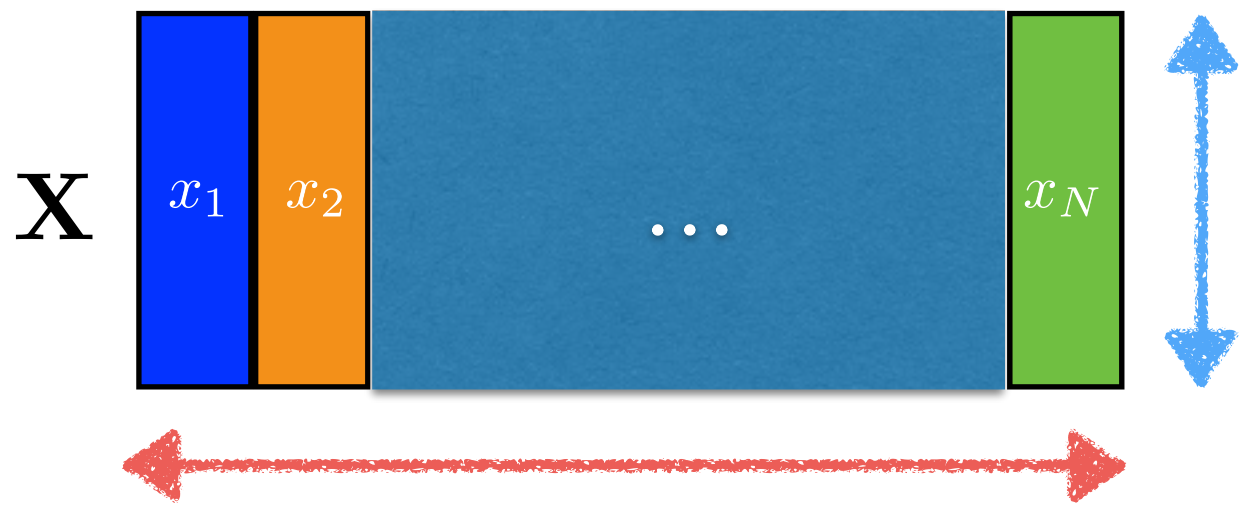
- High feature dimension *n*
- Large collection size *N*

Challenging dimensions

- Point cloud = large matrix of feature vectors

$\mathcal{X}$





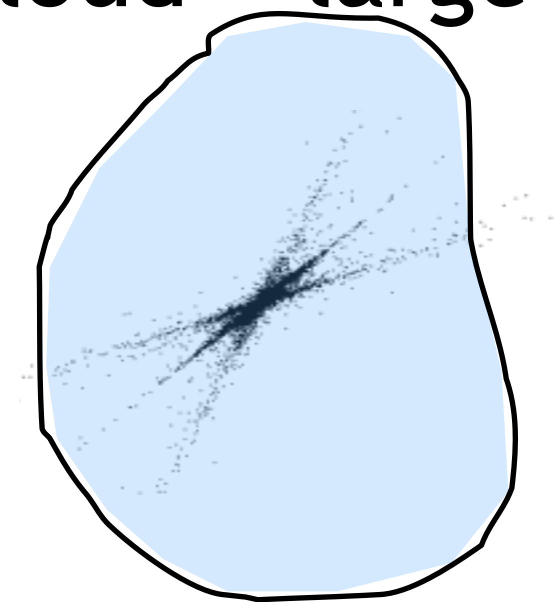
  - High feature dimension *n*
  - Large collection size *N*

Challenge: compress $\mathcal{X}$ before learning ?

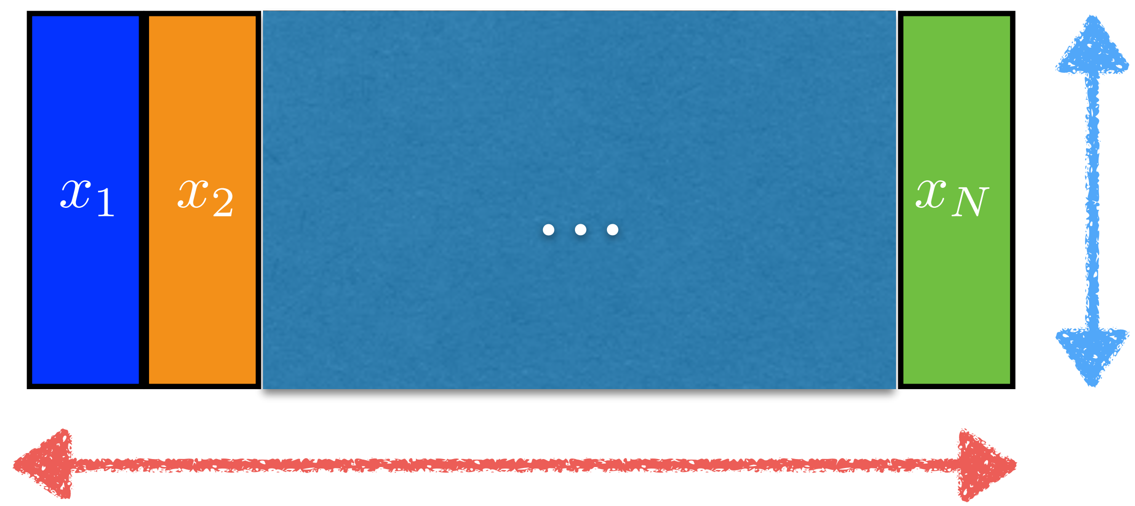
Compressive Machine Learning ?

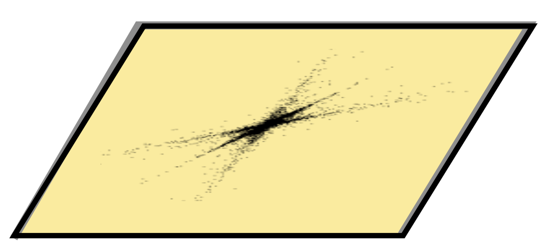
- Point cloud = large matrix of feature vectors

$\mathcal{X}$

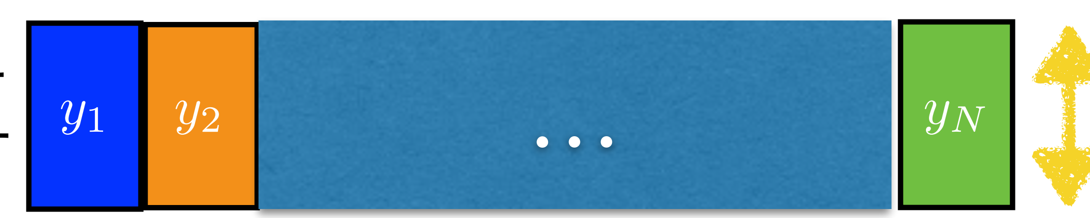


$\mathbf{X}$



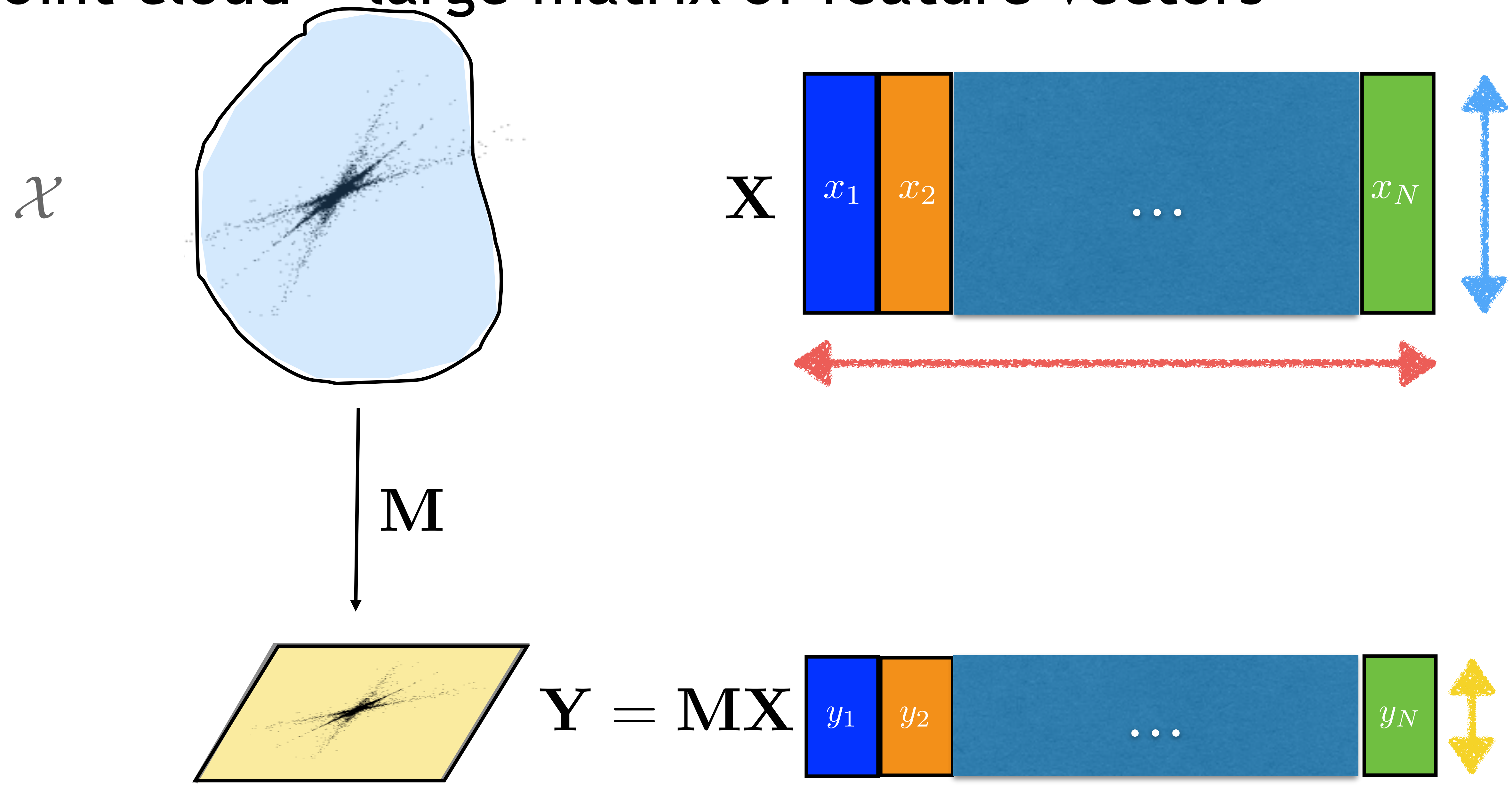


$\mathbf{Y} = \mathbf{MX}$



Compressive Machine Learning ?

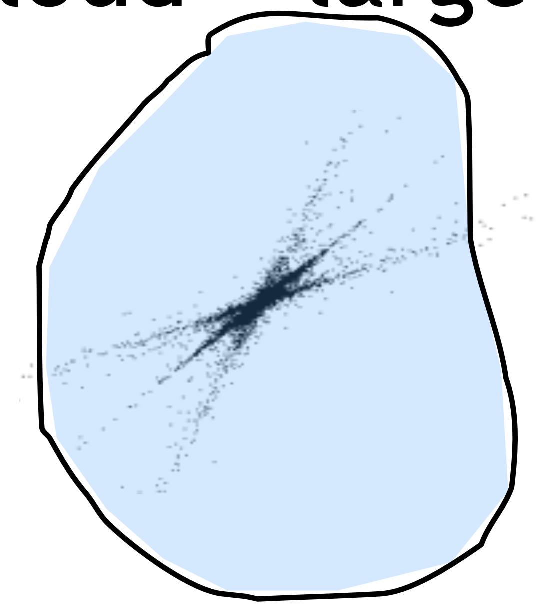
- Point cloud = large matrix of feature vectors



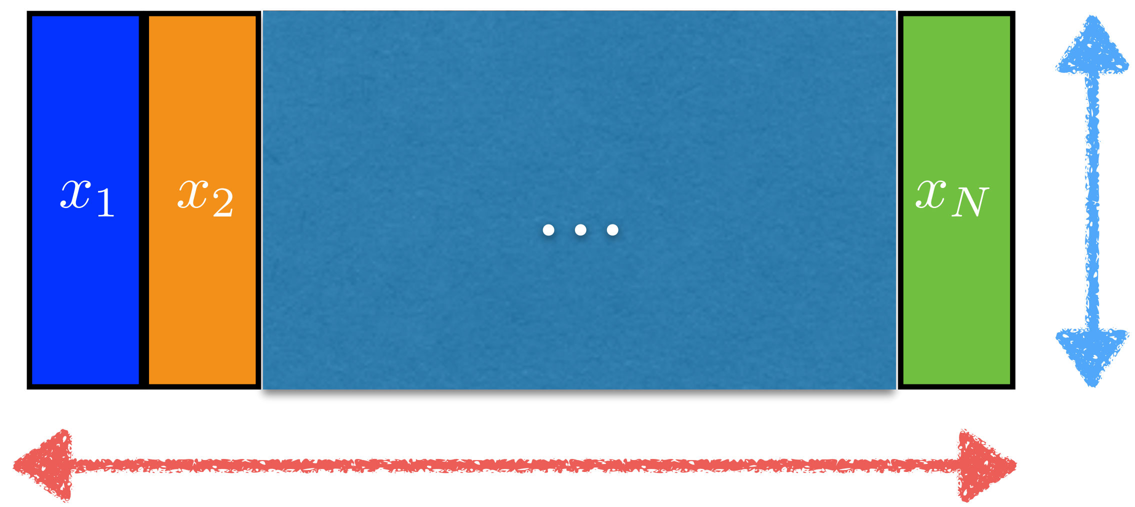
Compressive Machine Learning ?

- Point cloud = large matrix of feature vectors

$\mathcal{X}$



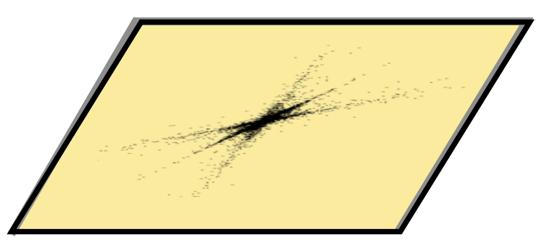
$\mathbf{X}$



- Reduce feature dimension

  [Calderbank & al 2009, Reboredo & al 2013]

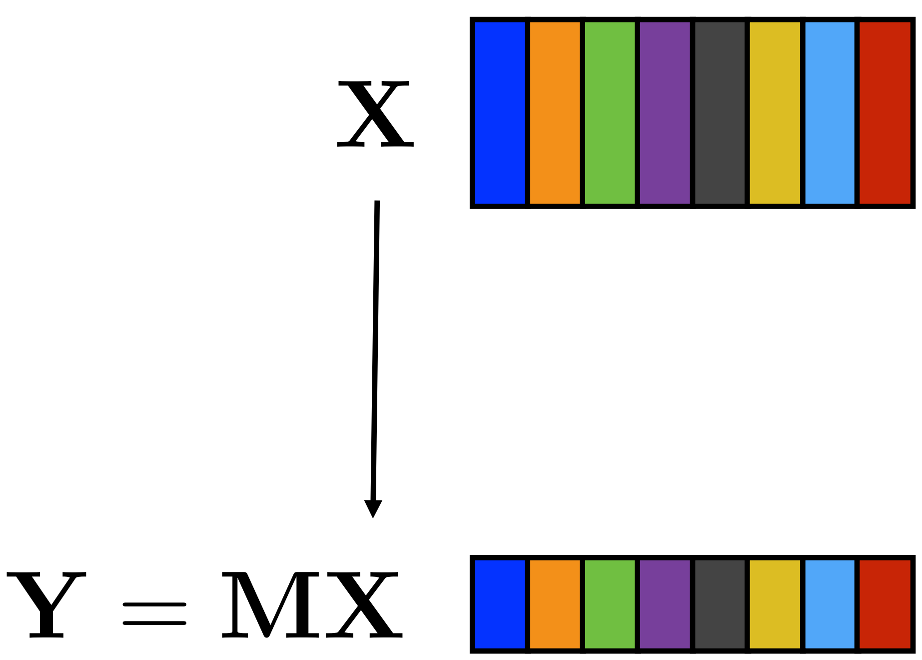
  - (Random) feature projection
  - Exploits / needs **low-dimensional *feature model***



$\mathbf{Y} = \mathbf{M}\mathbf{X}$

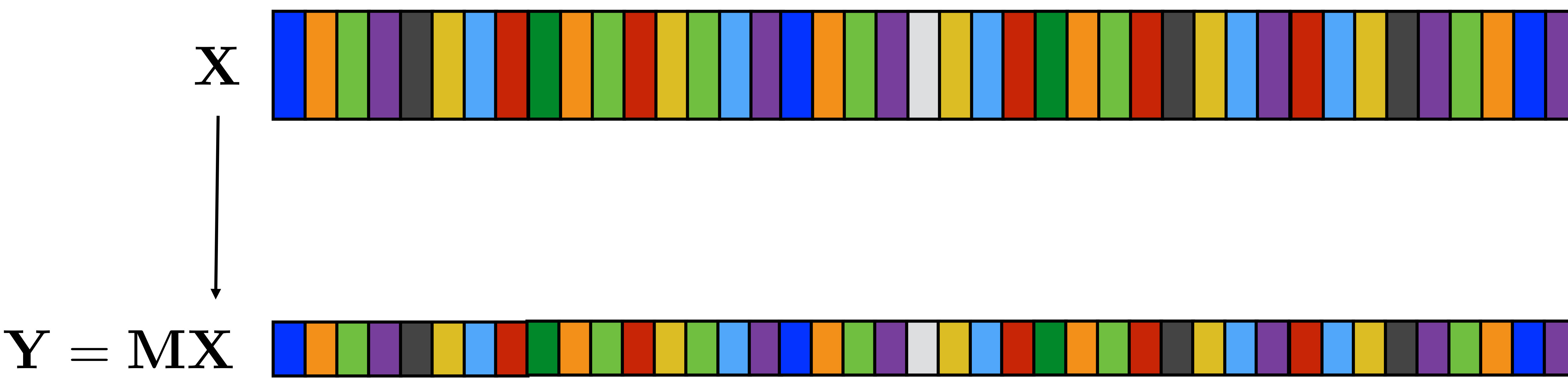
Challenges of large collections

- Feature projection: limited impact



Challenges of large collections

- Feature projection: limited impact

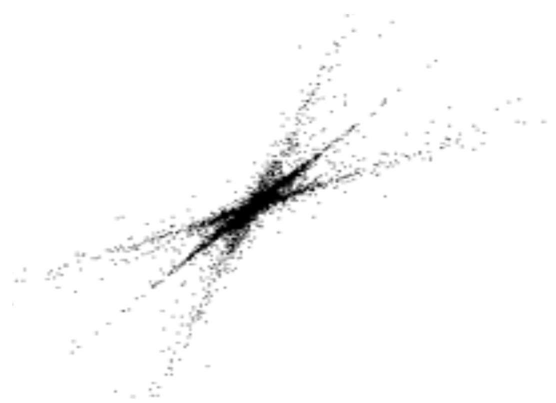


"Big Data" Challenge: compress **collection size**

Compressive Machine Learning ?

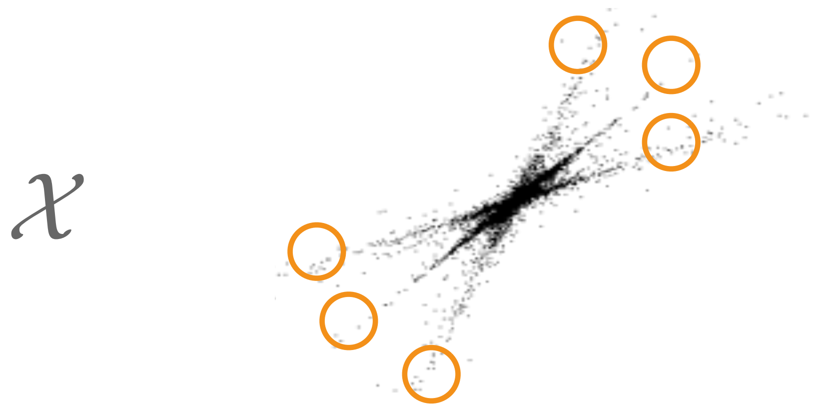
- Point cloud

$\mathcal{X}$



Compressive Machine Learning ?

- Point cloud



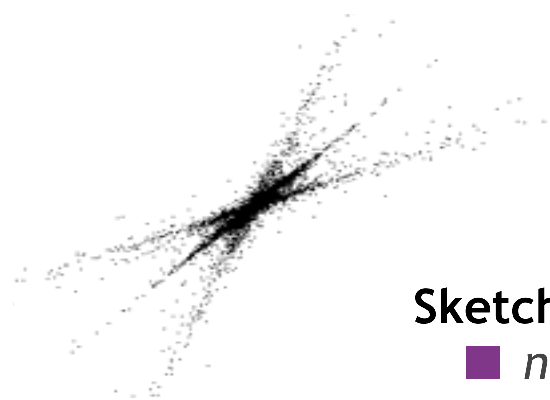
- Reduce collection dimension
  - (adaptive) column sampling / coresets

see e.g. [Agarwal & al 2003, Felman 2010]

Compressive Machine Learning ?

- **Point cloud**

$\mathcal{X}$



$\xrightarrow{\mathcal{M}}$

Sketching operator

- *nonlinear in the feature vectors*

$z \in \mathbb{R}^m$

- **Reduce collection dimension**
  - (adaptive) column sampling / coresets

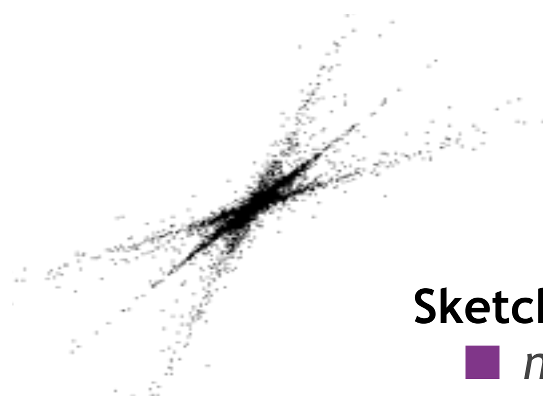
    see e.g. [Agarwal & al 2003, Felman 2010]
  - **sketching & hashing**

    see e.g. [Thaper & al 2002, Cormode & al 2005]

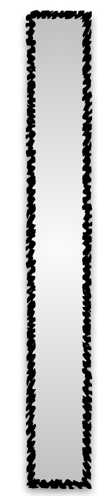
Compressive Machine Learning ?

- Point cloud = … empirical probability distribution

$\mathcal{X}$



$$\xrightarrow{\mathcal{M}}$$



$z \in \mathbb{R}^m$

Sketching operator
- *nonlinear in the feature vectors*
- ***linear** in their probability distribution*

- Reduce collection dimension
  - (adaptive) column sampling / coresets

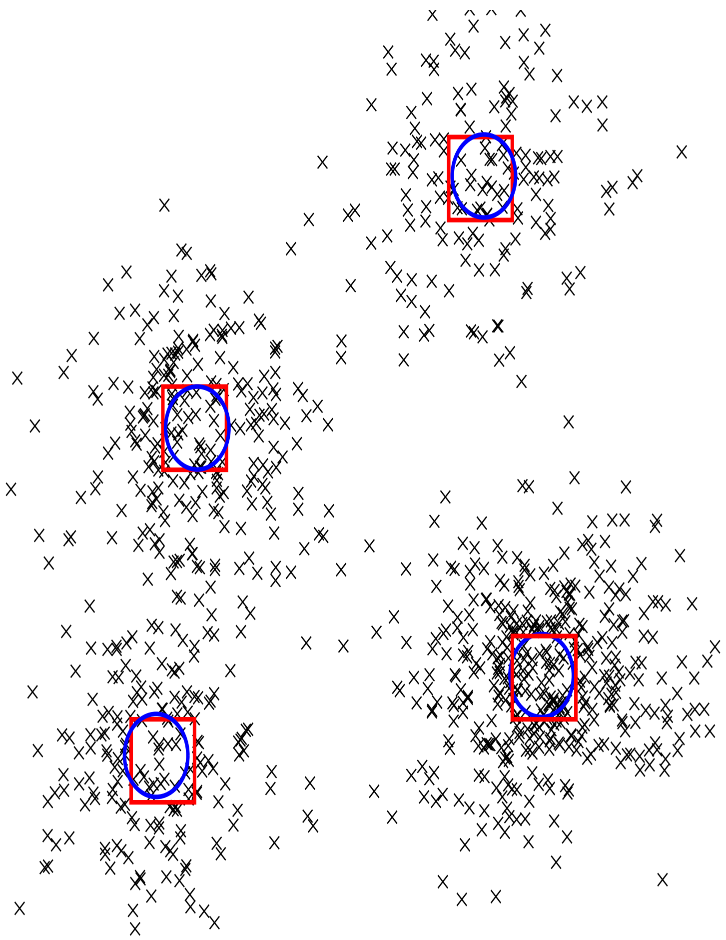
    see e.g. [Agarwal & al 2003, Felman 2010]
  - **sketching & hashing**

    see e.g. [Thaper & al 2002, Cormode & al 2005]

Example: Compressive K-means

$$\mathcal{X} \xrightarrow{\mathcal{M}} z \in \mathbb{R}^m$$

$$N = 1000; n = 2 \qquad m = 60$$



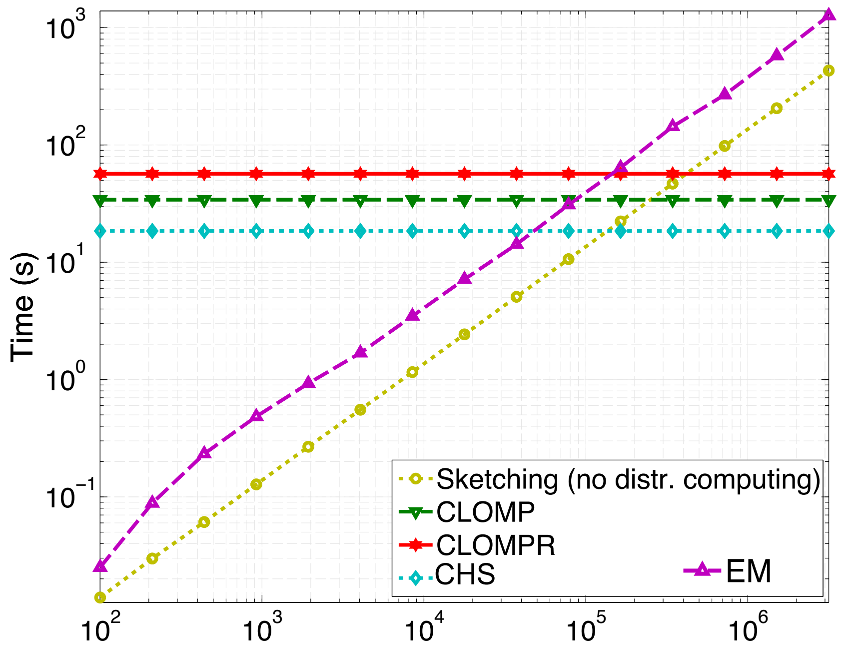
Recovery algorithm

estimated centroids

ground truth

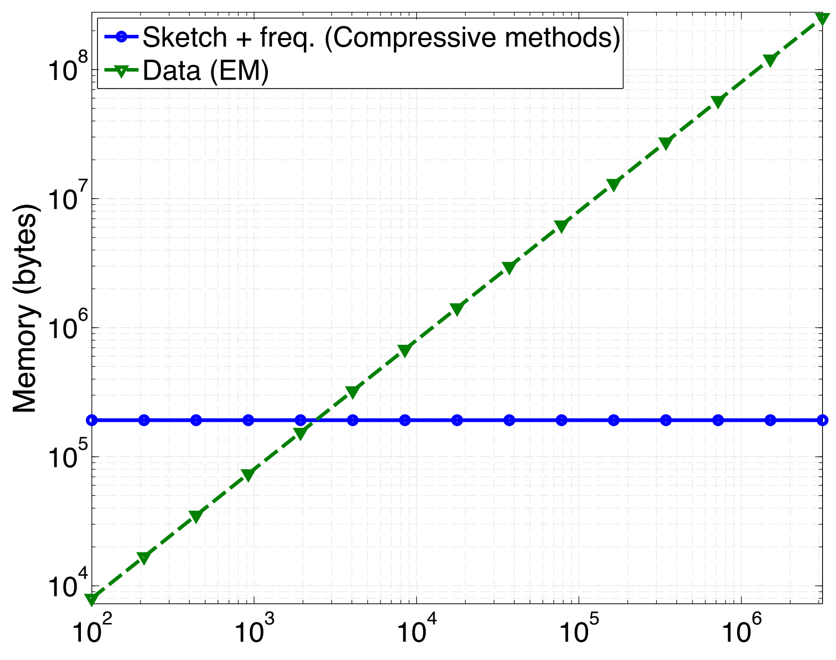
Computational impact of sketching

Computation time



Collection size N

Memory



Collection size N

Ph.D. A. Bourrier & N. Keriven

The Sketch Trick

- Data distribution

$$X \sim p(x)$$

- Sketch

The Sketch Trick

- Data distribution

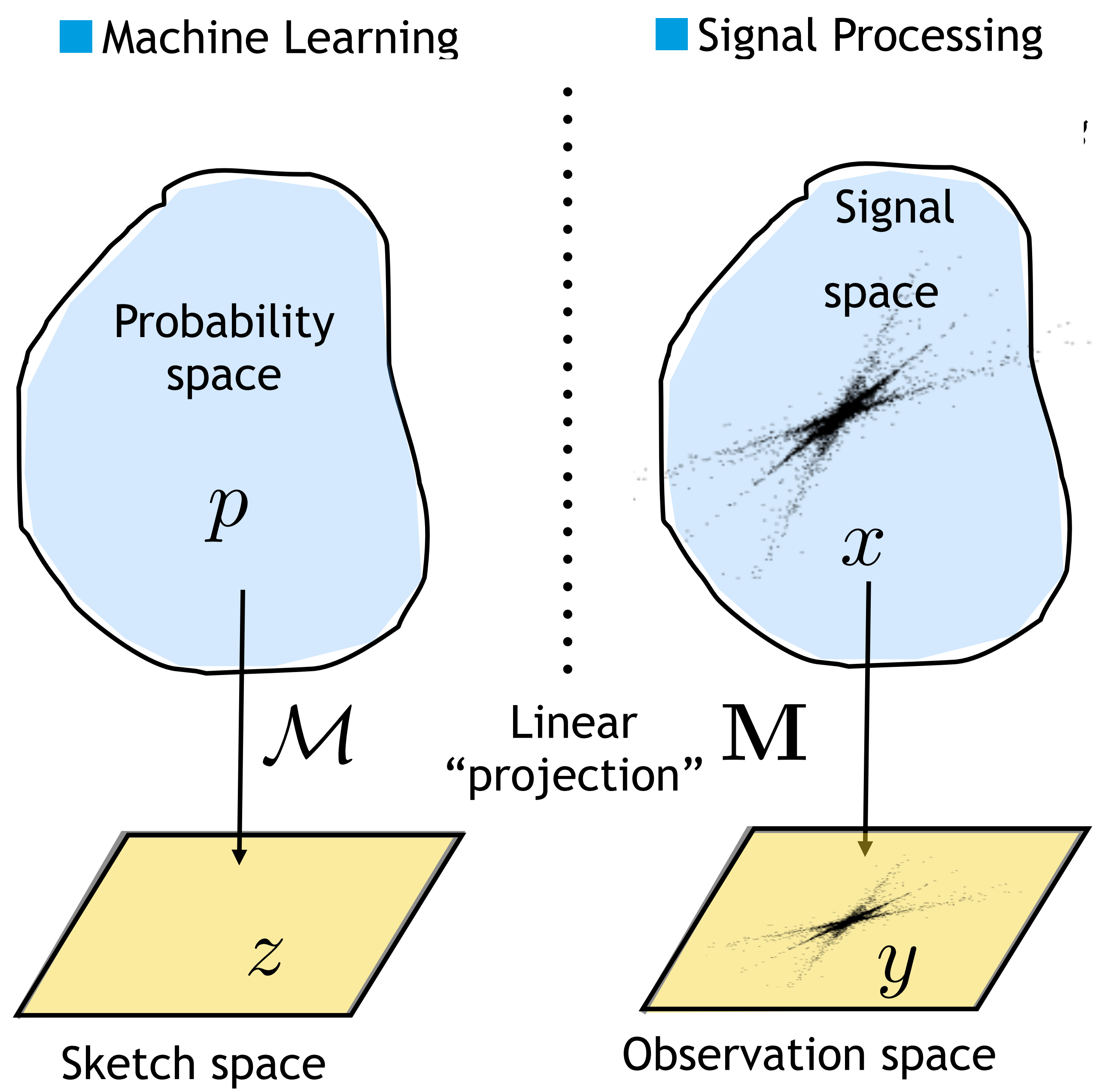
$$X \sim p(x)$$

- Sketch

$$z_\ell = \int h_\ell(x)p(x)dx$$

$$= \mathbb{E}h_\ell(X)$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} h_\ell(x_i)$$



The Sketch Trick

- Data distribution

$$X \sim p(x)$$

- Sketch

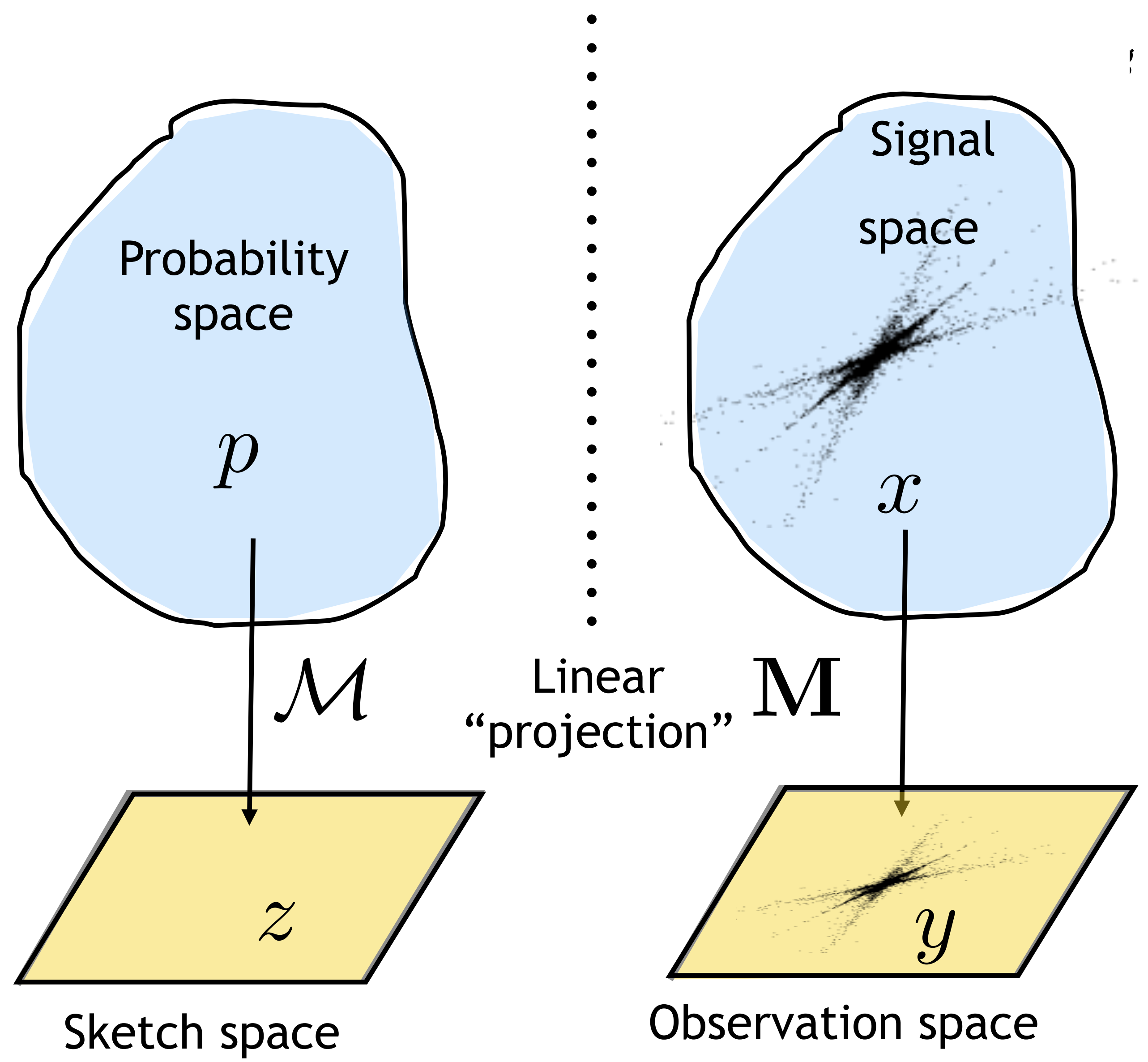
$$z_\ell = \int h_\ell(x)p(x)dx$$

$$= \mathbb{E}h_\ell(X)$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} h_\ell(x_i)$$

  - **nonlinear** in the feature vectors
  - **linear** in the distribution *p(x)*
    - finite-dimensional Mean Map Embedding, *cf Smola & al 2007, Sriperumbudur & al 2010*

- Machine Learning
- Signal Processing



The Sketch Trick

- Data distribution

$$X \sim p(x)$$

- Sketch

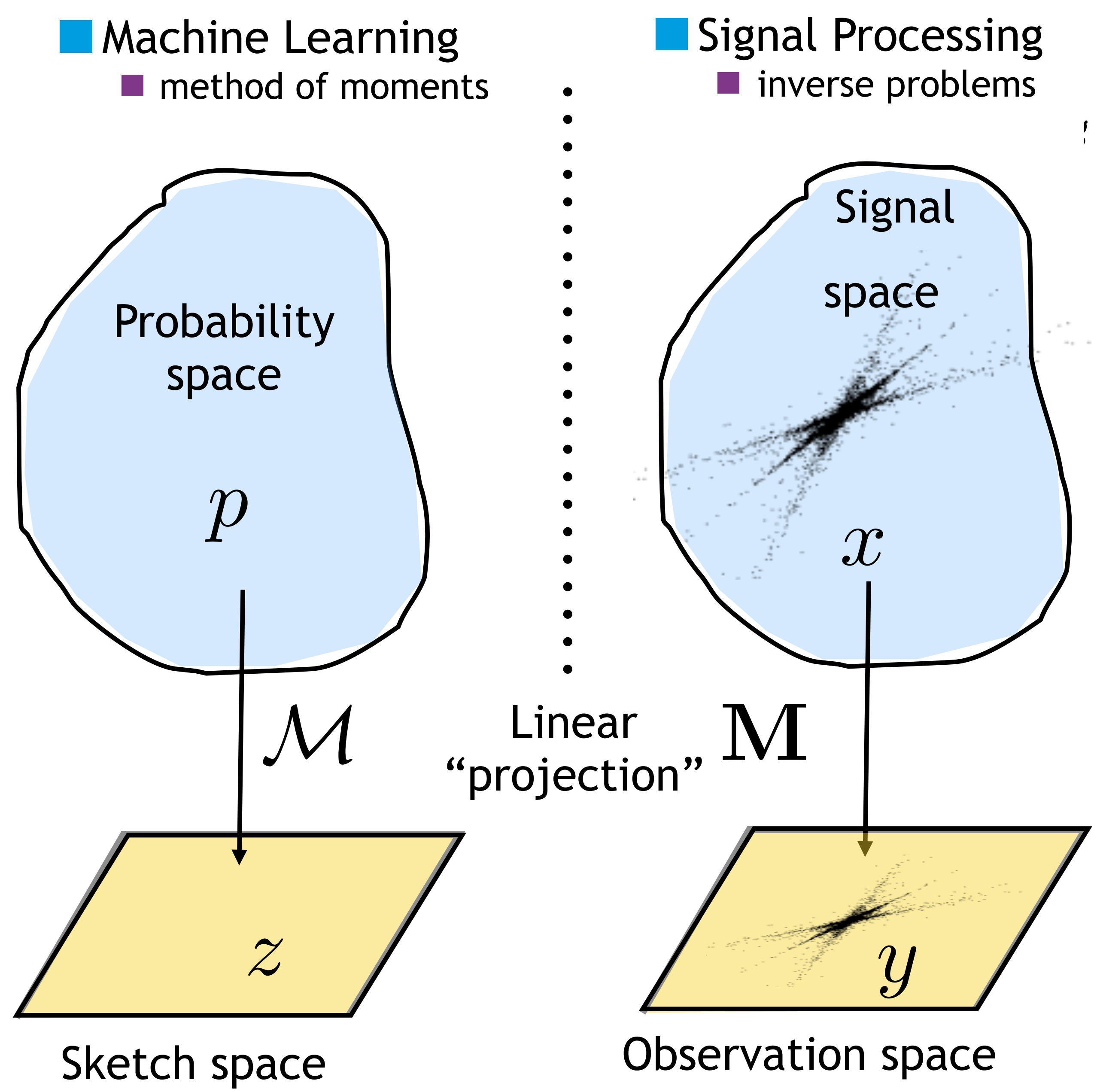
$$z_\ell = \int h_\ell(x) p(x) dx$$

$$= \mathbb{E} h_\ell(X)$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} h_\ell(x_i)$$

- **nonlinear** in the feature vectors
- **linear** in the distribution *p(x)*
  - finite-dimensional Mean Map Embedding, *cf Smola & al 2007, Sriperumbudur & al 2010*

Information preservation ?



The Sketch Trick

- **Data distribution**

$$X \sim p(x)$$

- **Sketch**

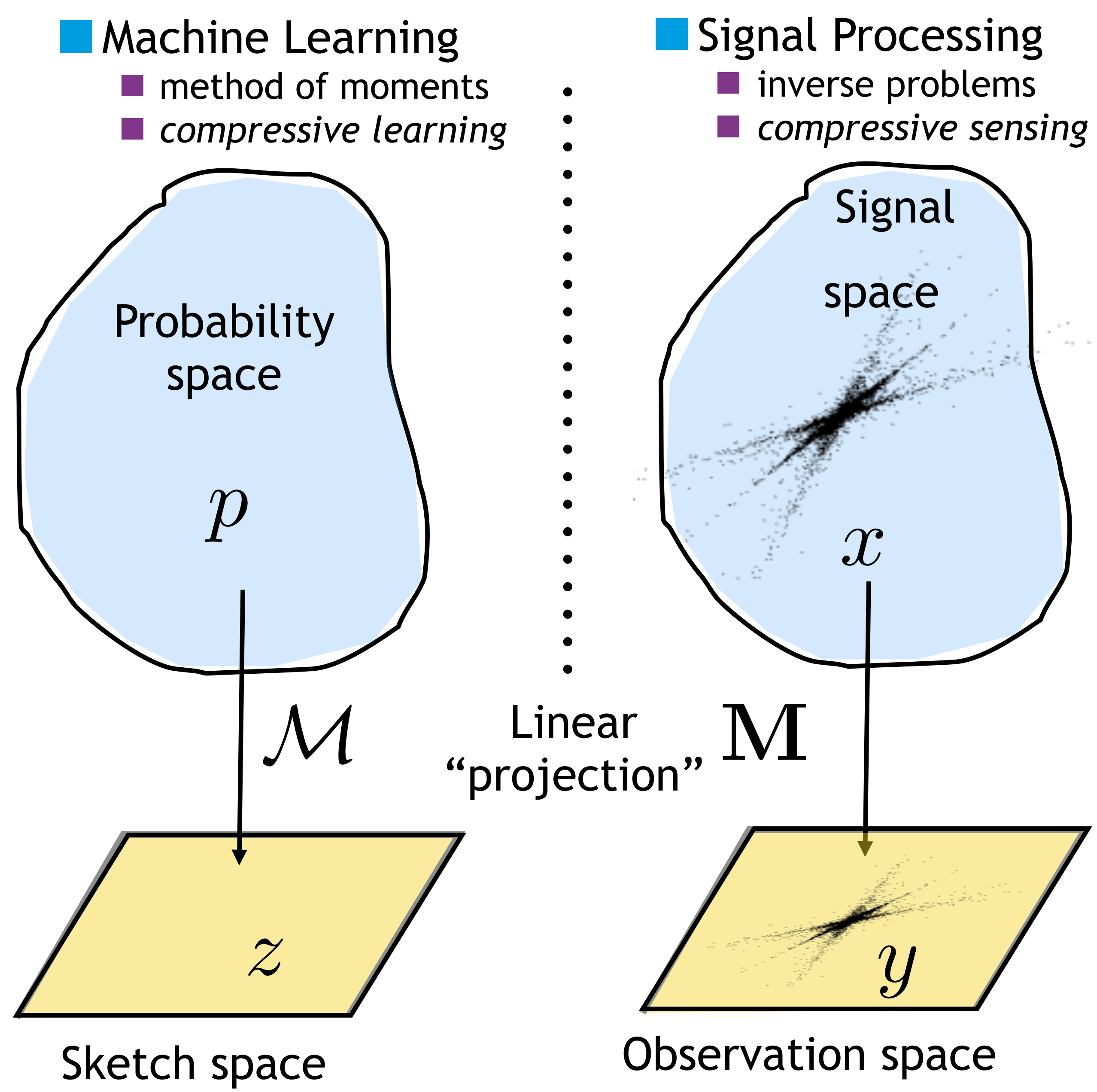
$$z_\ell = \int h_\ell(x) p(x) dx$$

$$= \mathbb{E} h_\ell(X)$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} h_\ell(x_i)$$

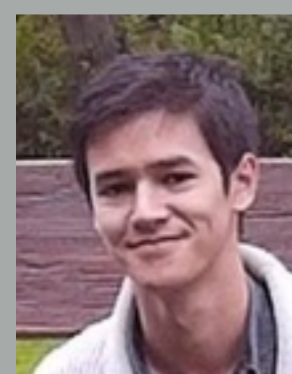
- **nonlinear** in the feature vectors
- **linear** in the distribution *p(x)*
  - finite-dimensional Mean Map Embedding, *cf Smola & al 2007, Sriperumbudur & al 2010*

Dimension reduction ?



PLEASE

projection, learning and sparsity for efficient data processing



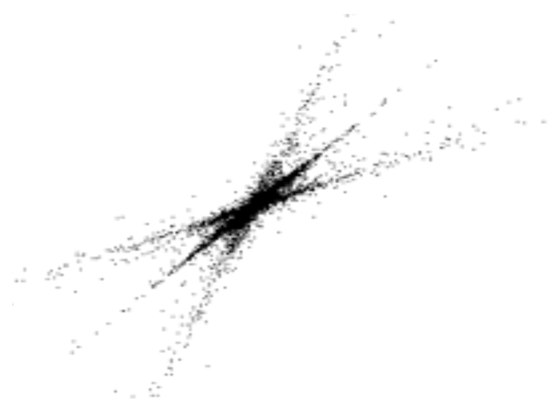
Compressive Learning (Heuristic) Examples

R. GRIBONVAL
London Workshop on Sparse Signal Processing, September 2016

Compressive Machine Learning

- Point cloud = empirical probability distribution

$\mathcal{X}$



$$\xrightarrow{\mathcal{M}} \quad z \in \mathbb{R}^m$$

Sketching operator

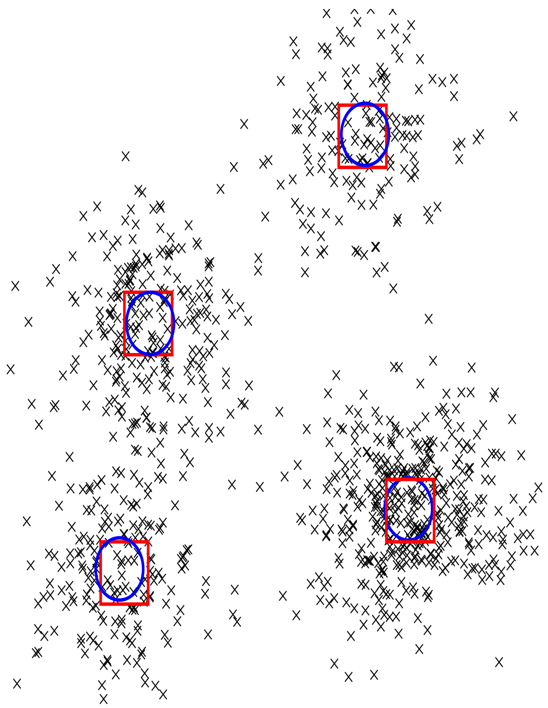
- Reduce collection dimension ~ **sketching**

$$z_\ell = \frac{1}{N}\sum_{i=1}^{N} h_\ell(x_i) \qquad 1 \leq \ell \leq m$$

Choosing information preserving sketch ?

Example: Compressive K-means

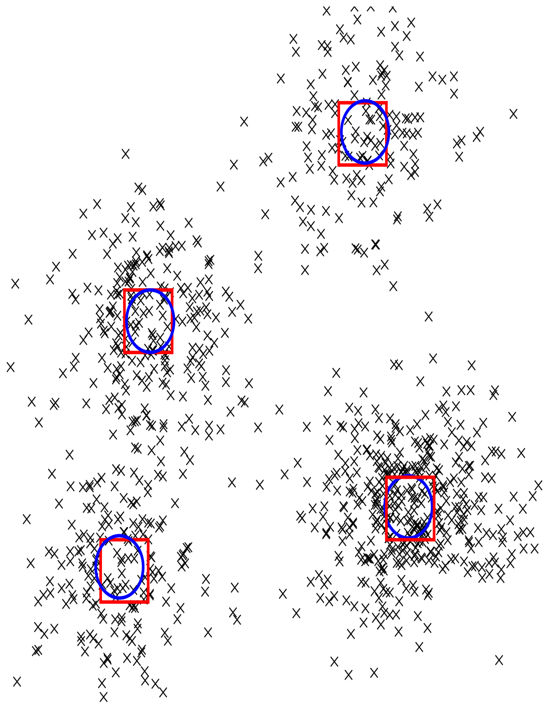
- Goal: find *k* centroids



- Standard approach = K-means

Example: Compressive K-means

- Goal: find *k* centroids

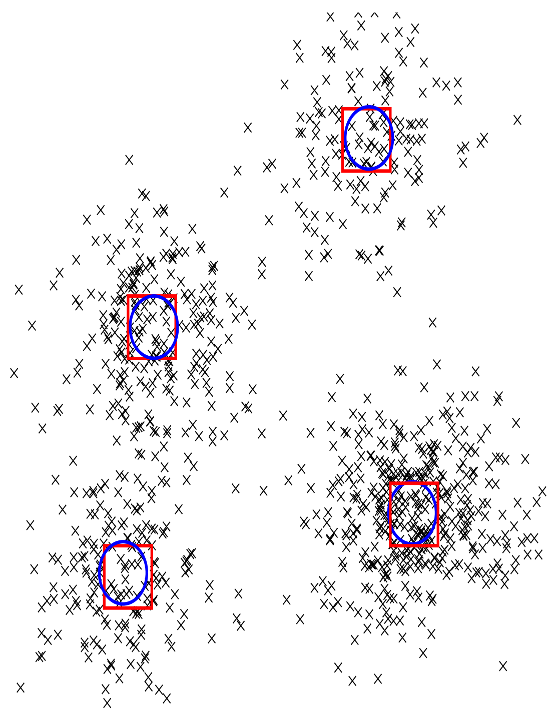


  - Standard approach = K-means

- Sketching approach
  - *p(x)* is **spatially localized**
    - need **“incoherent”** sampling
    - choose **Fourier** sampling

Example: Compressive K-means

- **Goal: find *k* centroids**



- Standard approach = K-means

- **Sketching approach**
  - *p(x)* is **spatially localized**
    - need **“incoherent”** sampling
    - choose **Fourier** sampling
  - sample **characteristic function**

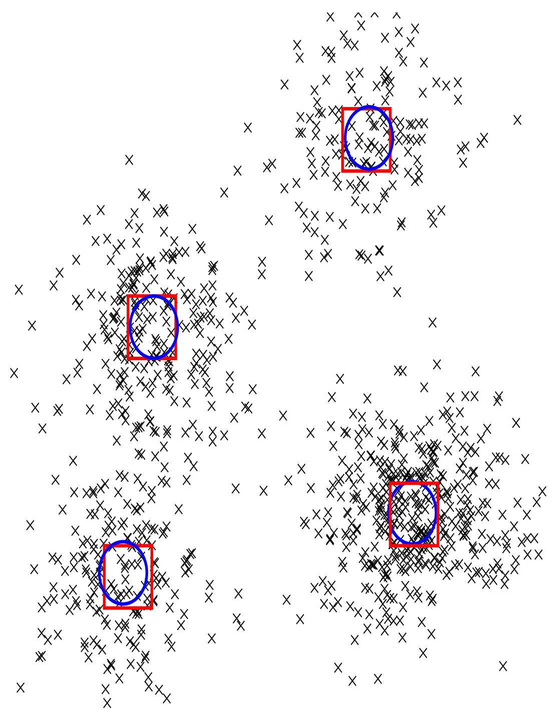
$$\boxed{z_\ell = \frac{1}{N}\sum_{i=1}^{N} e^{j w_\ell^\top x_i}}$$

- ~**pooled** Random Fourier Features, *cf Rahimi & Recht 2007*
  - choose ***sampling frequencies***

$$\omega_\ell \in \mathbb{R}^n$$

Example: Compressive K-means

- Goal: find *k* centroids



$$\mathcal{X} \xrightarrow[\text{Sampled Characteristic Function}]{\mathcal{M}} z \in \mathbb{R}^m$$

$N = 1000; n = 2$

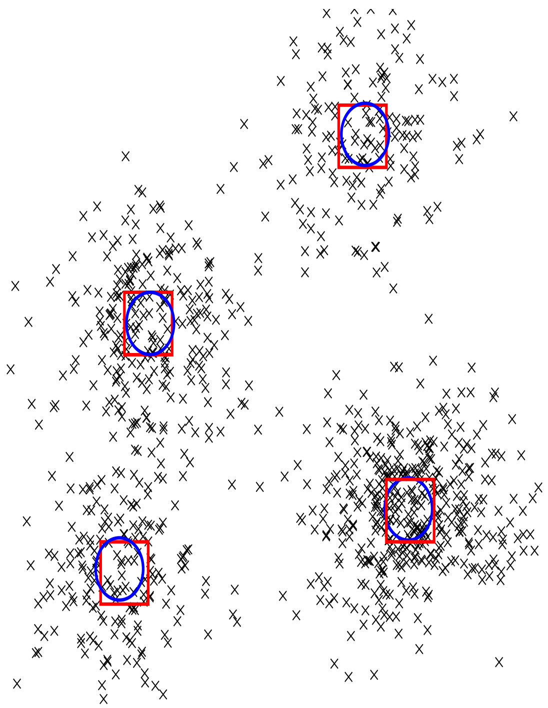
$m = 60$

Example: Compressive K-means

Density model=mixture of K Diracs

$$p \approx \sum_{k=1}^{K} \alpha_k \delta_{\theta_k}$$ ground truth

- Goal: find *k* centroids



$$\mathcal{X} \xrightarrow[\textit{Sampled Characteristic Function}]{\mathcal{M}} z \in \mathbb{R}^m$$

$$\approx \sum_{k=1}^{K} \alpha_k \mathcal{M} \delta_{\theta_k}$$

$N = 1000; n = 2$

$m = 60$

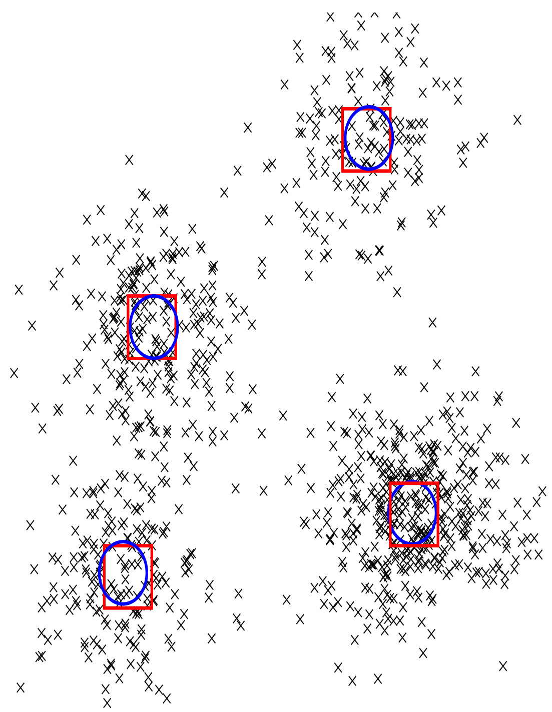
Example: Compressive K-means

Density model=mixture of K Diracs

$$p \approx \sum_{k=1}^{K} \alpha_k \delta_{\theta_k}$$ ground truth

estimated centroids

- Goal: find *k* centroids



CLOMP =***Compressive Learning OMP***
similar to: OMP with Replacement, Subspace Pursuit & CoSaMP

$$\approx \arg\min_{\alpha_k,\theta_k} \|z - \sum_{k=1}^{K} \alpha_k \mathcal{M}\delta_{\theta_k}\|_2$$

Δ Recovery algorithm = **"decoder"**

$$\mathcal{X} \xrightarrow{\mathcal{M}} z \in \mathbb{R}^m \approx \sum_{k=1}^{K} \alpha_k \mathcal{M}\delta_{\theta_k}$$

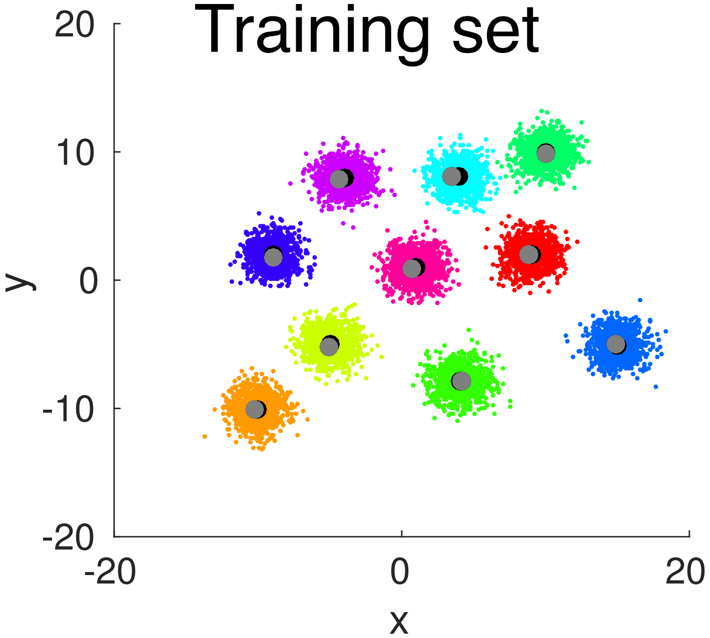
Sampled Characteristic Function

$N = 1000; n = 2$

$m = 60$

Compressive K-Means: Empirical Results

Training set



Sketch vector

Matrix of centroids

$$\xrightarrow{\mathcal{M}} \quad z \in \mathbb{R}^m \longrightarrow \mathbf{C} = \texttt{CLOMP}(z) \in \mathbb{R}^{n \times K}$$

$$\mathbf{x}_i \in \mathbb{R}^n$$

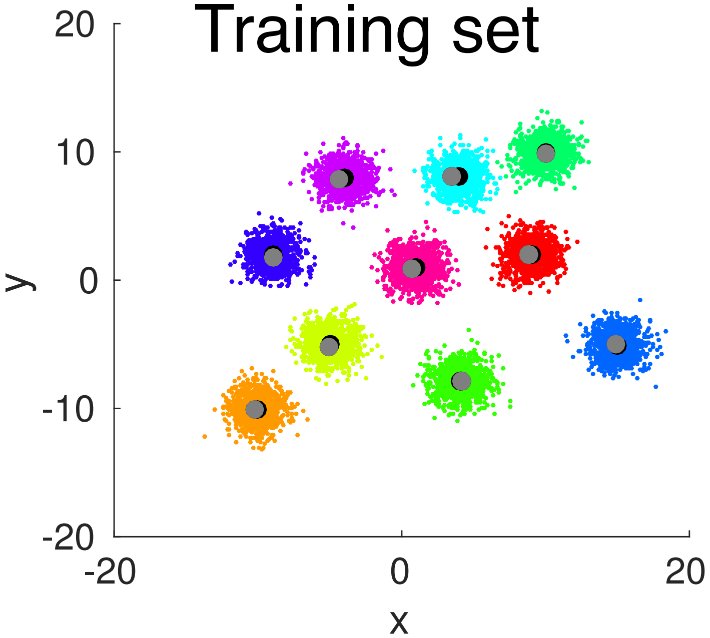
N training samples
K clusters

K-means objective

$$\mathrm{SSE}(\mathcal{X}, \mathbf{C}) = \sum_{i=1}^{N} \min_k \|\mathbf{x}_i - \mathbf{c}_k\|^2$$

Compressive K-Means: Empirical Results

Training set | Sketch vector | Matrix of centroids



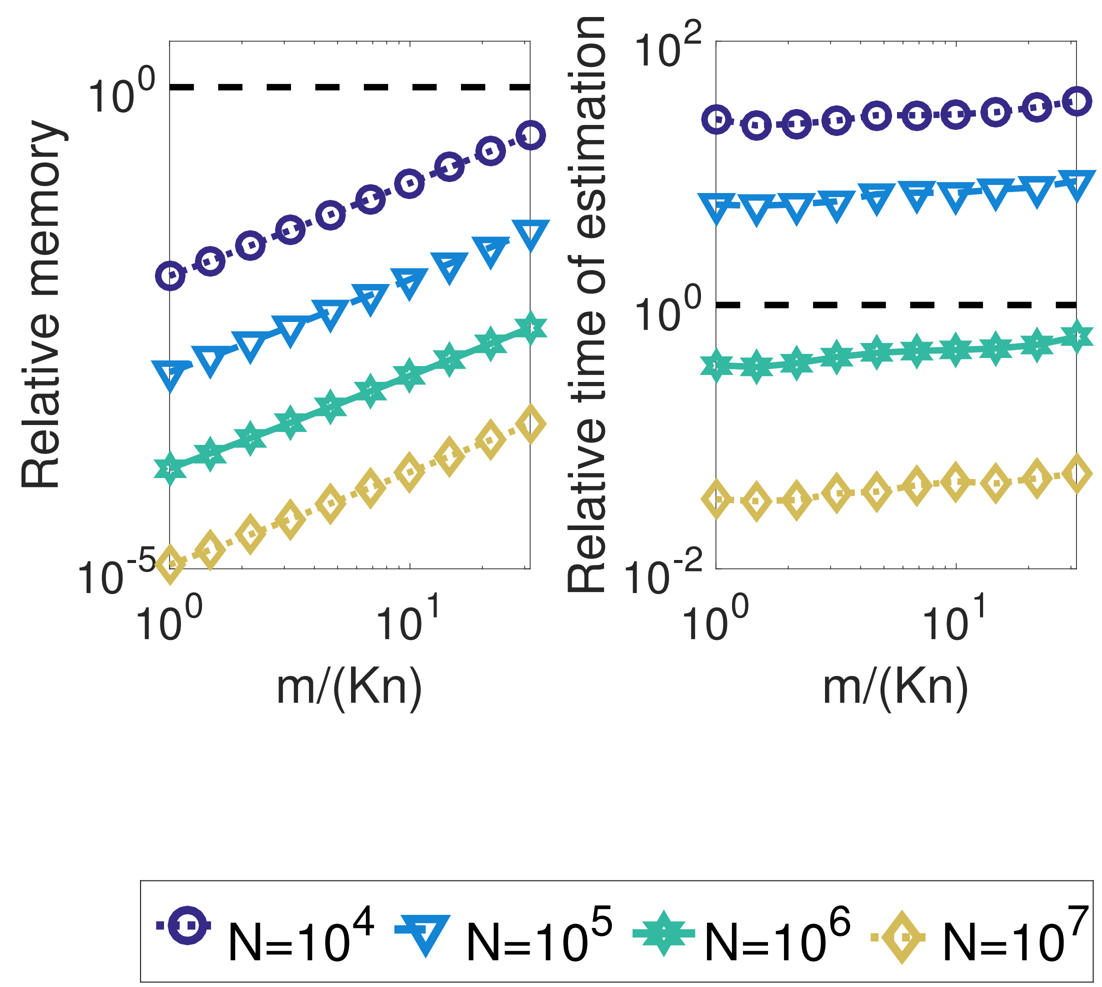
$$\xrightarrow{\mathcal{M}} \quad z \in \mathbb{R}^m \longrightarrow \mathbf{C} = \texttt{CLOMP}(z) \in \mathbb{R}^{n \times K}$$

$\mathbf{x}_i \in \mathbb{R}^n$

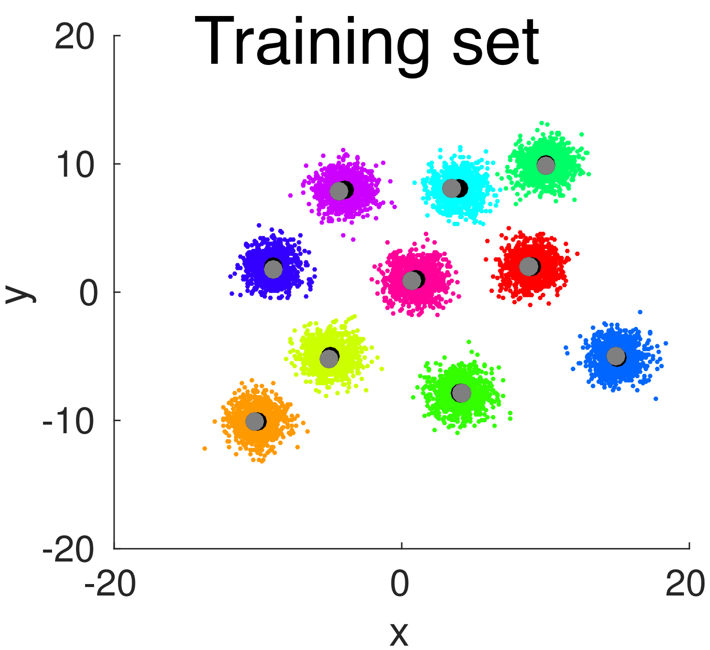
N training samples
K clusters

K-means objective

$$\mathrm{SSE}(\mathcal{X}, \mathbf{C}) = \sum_{i=1}^{N} \min_k \|\mathbf{x}_i - \mathbf{c}_k\|^2.$$



Compressive K-Means: Empirical Results



Sketch vector

Matrix of centroids

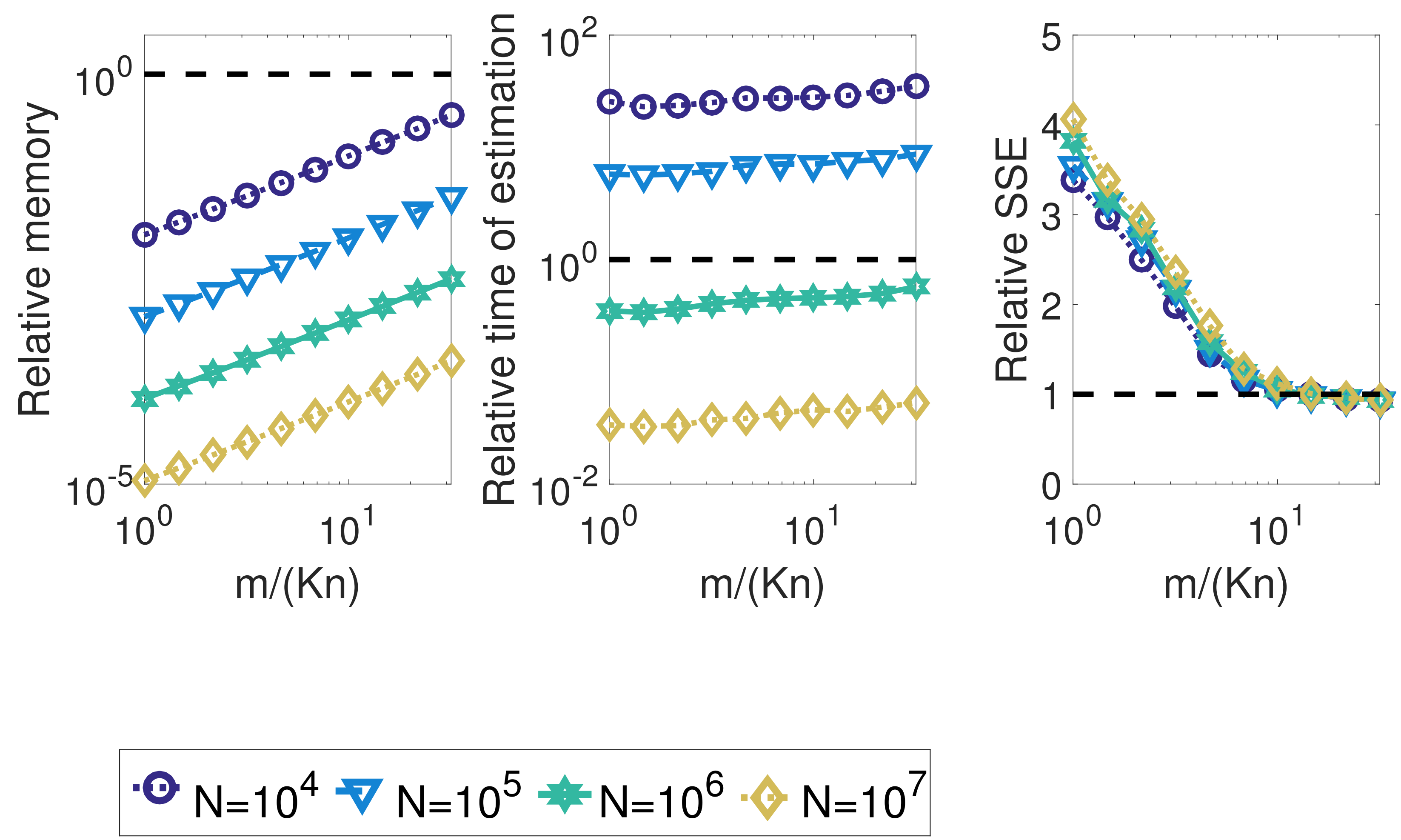
$$\xrightarrow{\mathcal{M}} z \in \mathbb{R}^m \longrightarrow \mathbf{C} = \texttt{CLOMP}(z) \in \mathbb{R}^{n \times K}$$

$\mathbf{x}_i \in \mathbb{R}^n$

N training samples
K clusters

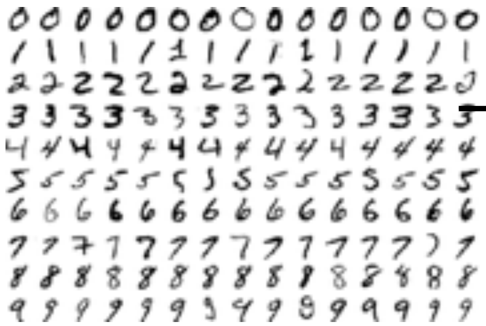
K-means objective

$$\mathrm{SSE}(\mathcal{X}, \mathbf{C}) = \sum_{i=1}^{N} \min_k \|\mathbf{x}_i - \mathbf{c}_k\|^2.$$



Compressive K-Means: Empirical Results

Training set



Spectral features

$\mathbf{x}_i \in \mathbb{R}^n$

$\xrightarrow{\mathcal{M}}$

Sketch vector

$z \in \mathbb{R}^m \longrightarrow$

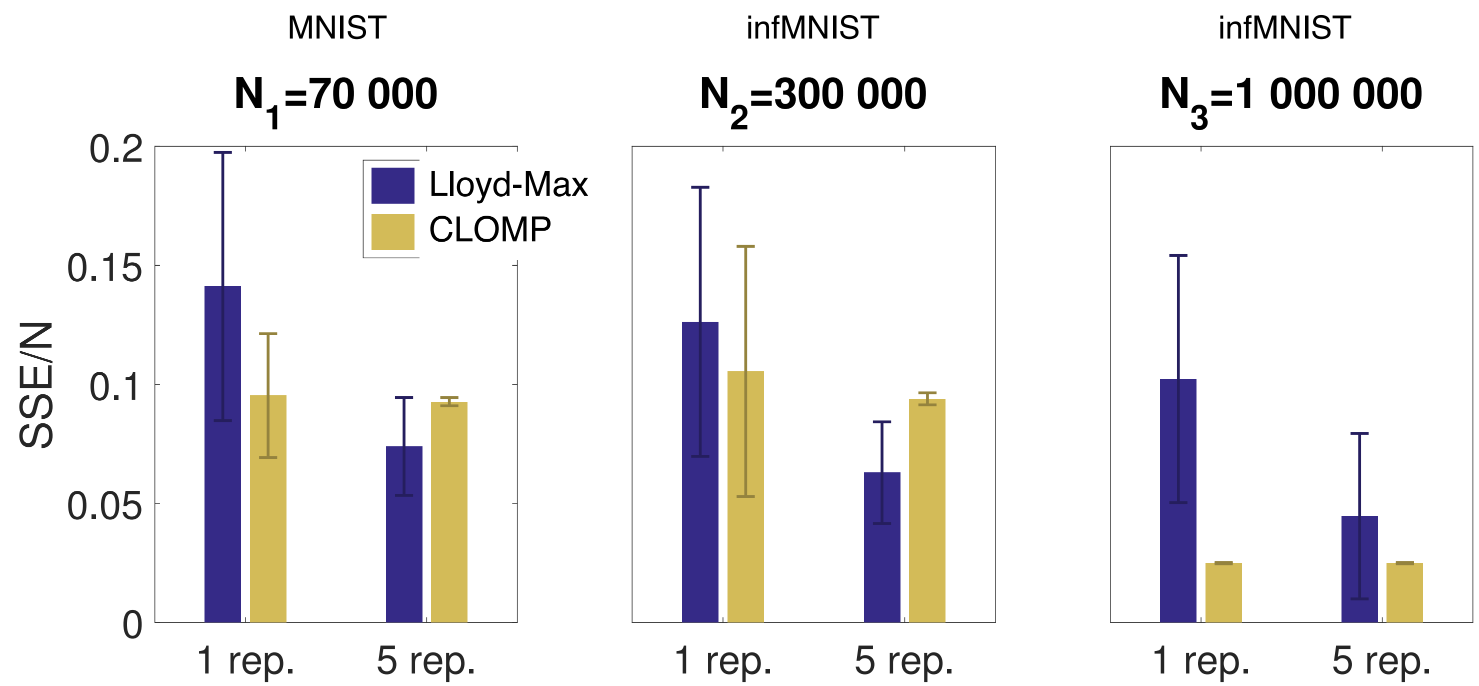
Matrix of centroids

$\mathbf{C} = \mathtt{CLOMP}(z) \in \mathbb{R}^{n \times K}$

N training samples
K=10 clusters

K-means objective

$$\mathrm{SSE}(\mathcal{X}, \mathbf{C}) = \sum_{i=1}^{N} \min_k \|\mathbf{x}_i - \mathbf{c}_k\|^2.$$



Lloyd-Max vs Sketch+CLOMP algorithm
with 1 or 5 replicates (random initialization)

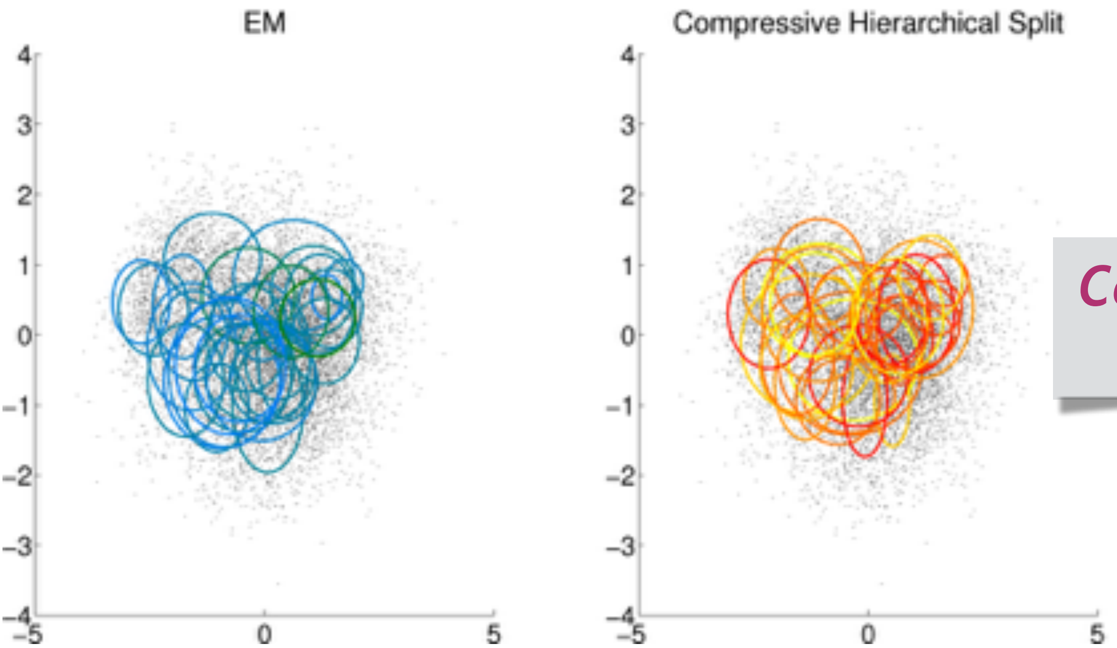
Example: Compressive GMM

- Goal: fit *k* Gaussians

Density model=GMM with diagonal covariance

$$p \approx \sum_{k=1}^{K} \alpha_k p_{\theta_k}$$

estimated GMM parameters (Θ, α)



Recovery algorithm = **"decoder"**

Δ

***Compressive Hierarchical Splitting** (CHS)*
= extension of CLOMP *to general GMM*

$$\approx \arg\min_{\alpha_k, \theta_k} \|z - \sum_{k=1}^{K} \alpha_k \mathcal{M} p_{\theta_k}\|_2$$

$$\mathcal{X} \xrightarrow[\text{Sampled Characteristic Function}]{\mathcal{M}} z \in \mathbb{R}^m \approx \sum_{k=1}^{K} \alpha_k \mathcal{M} p_{\theta_k}$$

$N = 60\ 000\ 000; n = 12$ $\quad m = 5\ 000$

Proof of Concept: Speaker Verification Results (DET-curves)

~ 50 Gbytes

~ 1000 hours of speech

- **MFCC coefficients** $x_i \in \mathbb{R}^{12}$

$$N = 300\ 000\ 000$$

Proof of Concept: Speaker Verification Results (DET-curves)

~ 50 Gbytes

~ 1000 hours of speech

- **MFCC coefficients** $x_i \in \mathbb{R}^{12}$

$$N = 300\ 000\ 000$$

- **After silence detection**

$$N = \quad 60\ 000\ 000$$

Proof of Concept: Speaker Verification Results (DET-curves)

~ 50 Gbytes
~ 1000 hours of speech

- **MFCC coefficients** $x_i \in \mathbb{R}^{12}$

  $N = 300\ 000\ 000$

- **After silence detection**

  $N = \quad 60\ 000\ 000$

- **Maximum size manageable by EM**

  $N = \qquad 300\ 000$

Proof of Concept: Speaker Verification Results (DET-curves)

~ 50 Gbytes
~ 1000 hours of speech

- **MFCC coefficients** $x_i \in \mathbb{R}^{12}$

$$N = 300\ 000\ 000$$

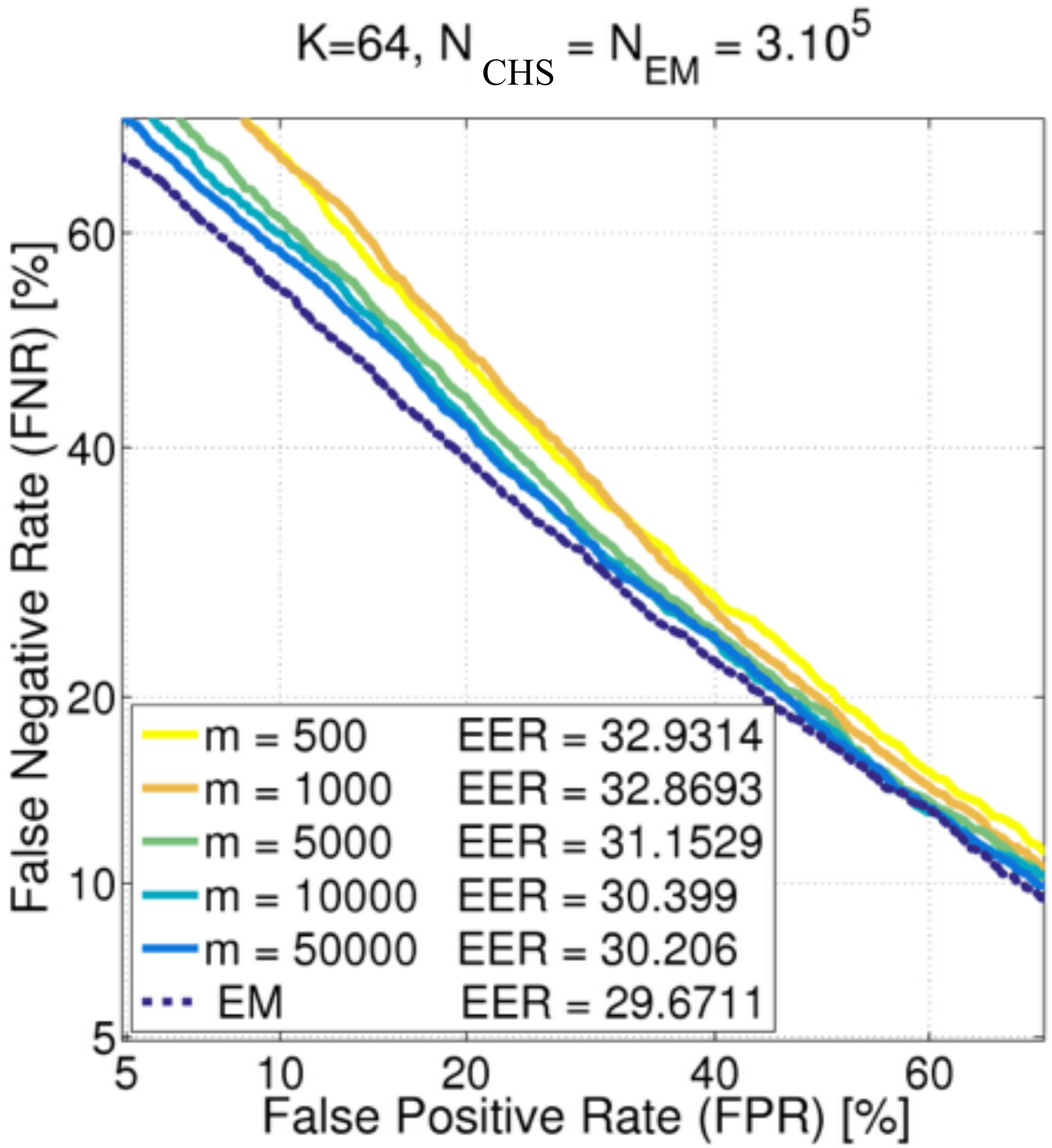
- **After silence detection**

$$N = \quad 60\ 000\ 000$$

- **Maximum size manageable by EM**

$$N = \quad 300\ 000$$

for EM
for CHS



Proof of Concept: Speaker Verification Results (DET-curves)

~ 50 Gbytes
~ 1000 hours of speech

- **MFCC coefficients** $x_i \in \mathbb{R}^{12}$

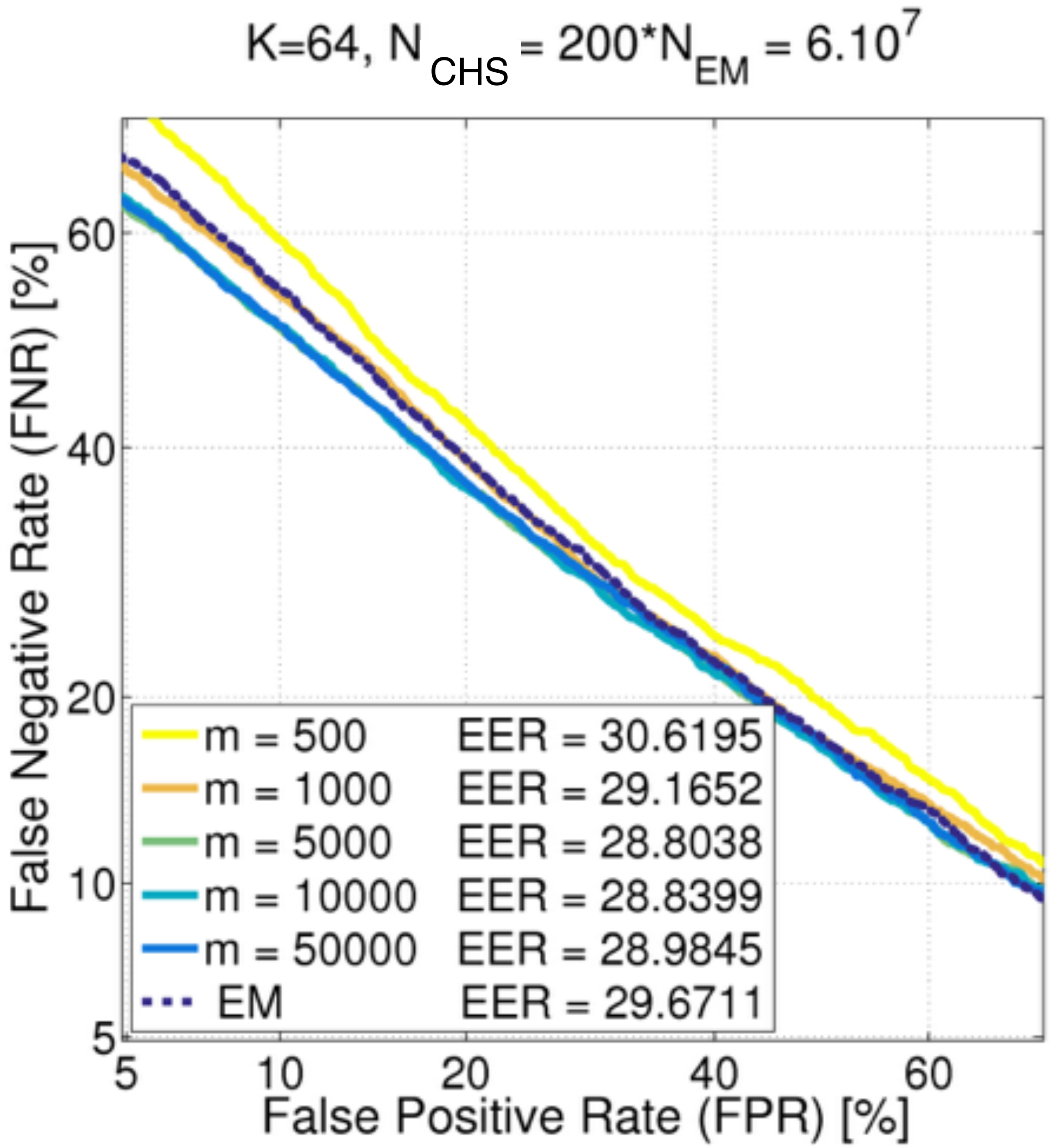
$$N = 300\ 000\ 000$$

- **After silence detection**

$$\boxed{N = \quad 60\ 000\ 000}$$ for CHS

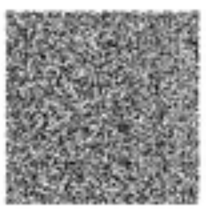
- **Maximum size manageable by EM**

$$\boxed{N = \quad 300\ 000}$$ for EM

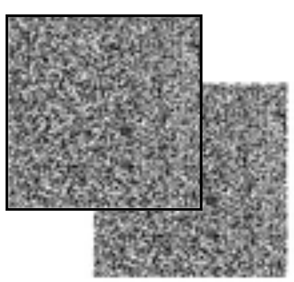


Proof of Concept: Speaker Verification Results (DET-curves)

~ 50 Gbytes
~ 1000 hours of speech

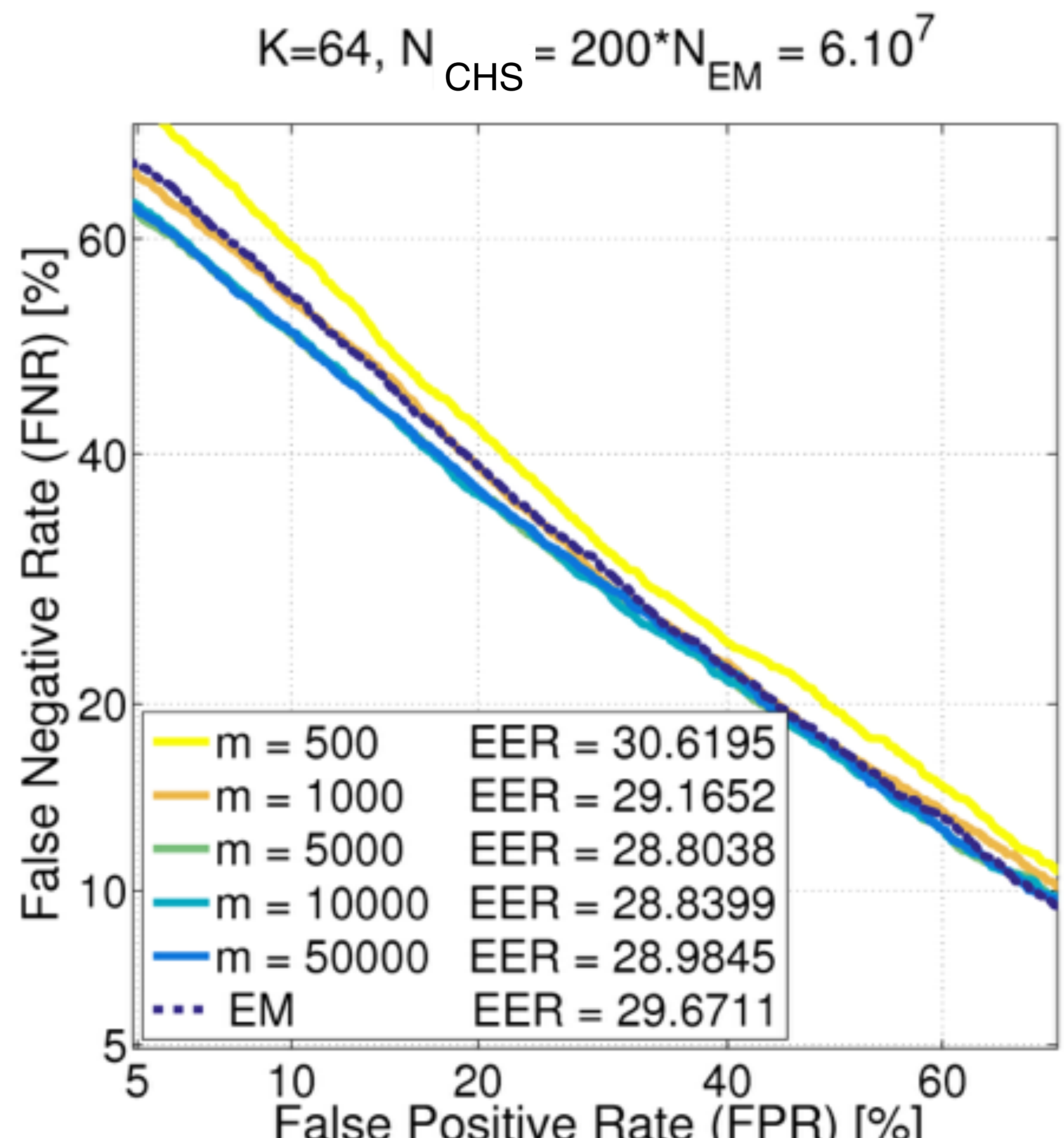


- **m= 500**
  - 7 200 000-fold compression



- **m= 1000**
  - 3 600 000-fold compression

- **m= 5 000**
  - 720 000-fold compression
    - *exploit whole collection*
    - *improved performance*



PLEASE

projection, learning and sparsity for efficient data processing

Computational Efficiency

R. GRIBONVAL

London Workshop on Sparse Signal Processing, September 2016

Computational Aspects

- **Sketching**
  - empirical characteristic function

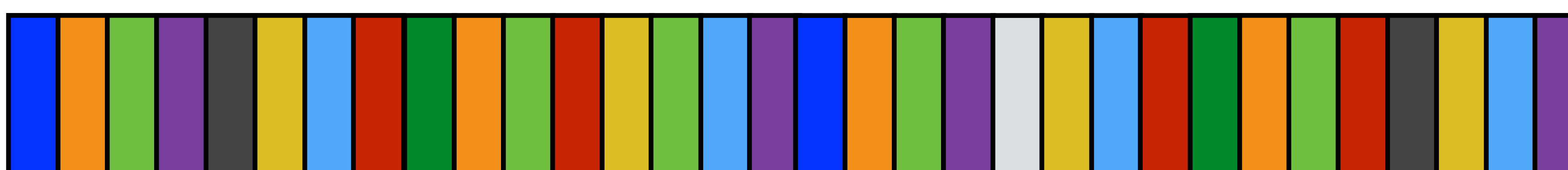
$$z_\ell = \frac{1}{N}\sum_{i=1}^{N} e^{j w_\ell^\top x_i}$$

Computational Aspects

- **Sketching**
  - empirical characteristic function

$$z_\ell = \frac{1}{N}\sum_{i=1}^{N} e^{j w_\ell^\top x_i}$$

$\mathbf{X}$

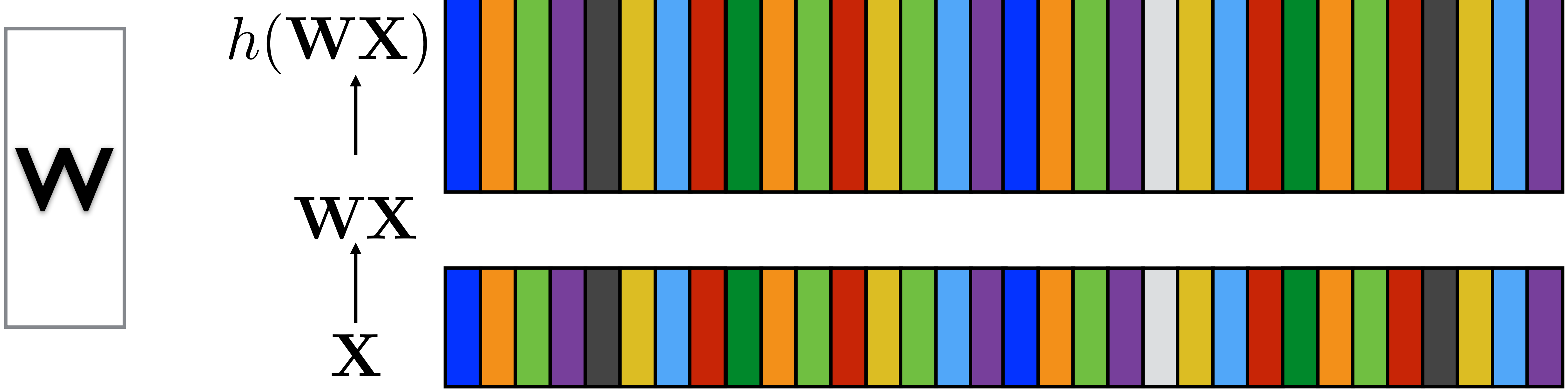


Computational Aspects

- Sketching
  - empirical characteristic function

$$z_\ell = \frac{1}{N}\sum_{i=1}^{N} e^{j w_\ell^\top x_i}$$

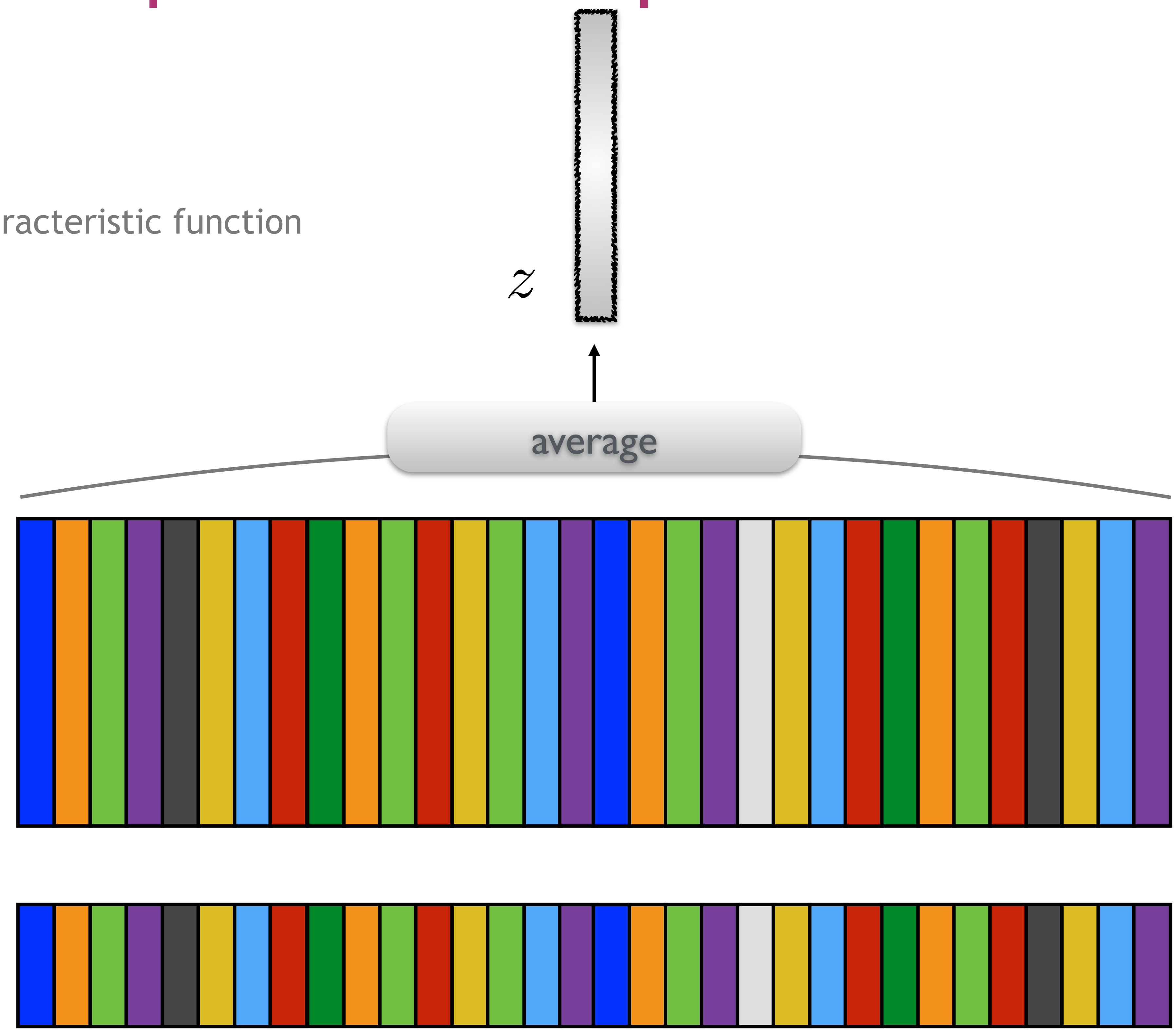
$$h(\cdot) = e^{j(\cdot)}$$



Computational Aspects

- **Sketching**
  - empirical characteristic function

$$z_\ell = \frac{1}{N}\sum_{i=1}^{N} e^{j w_\ell^\top x_i}$$



$$h(\cdot) = e^{j(\cdot)}$$

$$\mathbf{W}$$

$$h(\mathbf{WX})$$

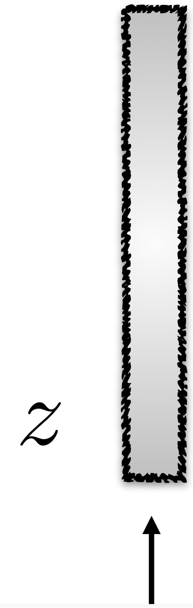
$$\mathbf{WX}$$

$$\mathbf{X}$$

Computational Aspects

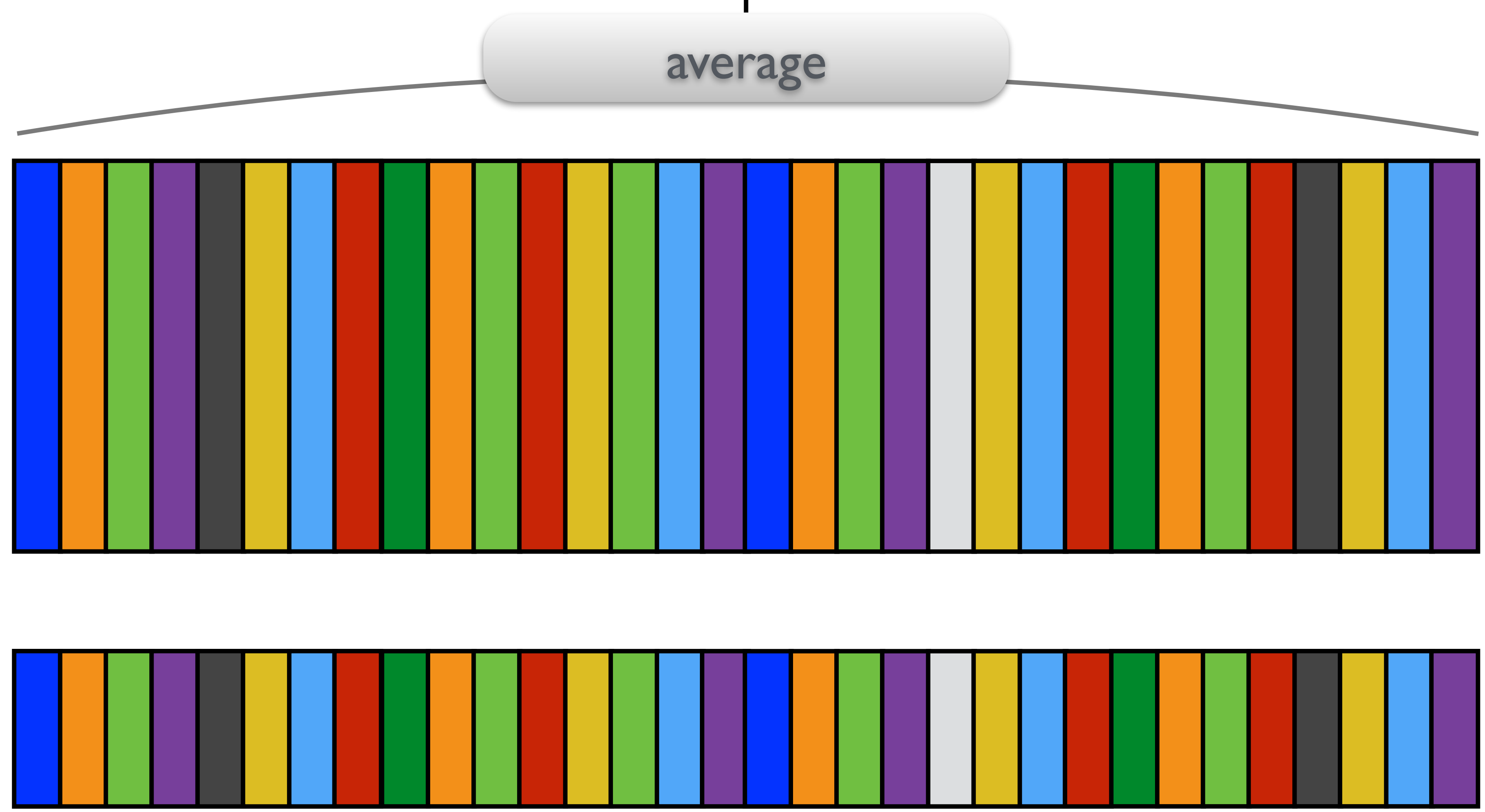
- **Sketching**
  - empirical characteristic function

$$z_\ell = \frac{1}{N}\sum_{i=1}^{N} e^{j w_\ell^\top x_i}$$

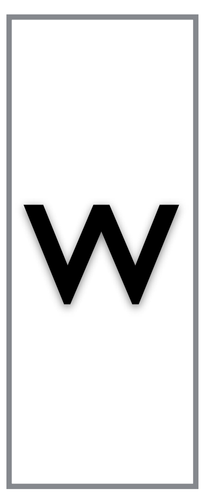


- ~ One-layer random neural net
  - Decoding = next layers
  - DNN ~ hierarchical sketching ?

see also [Bruna & al 2013, Giryes & al 2015]



$$h(\cdot) = e^{j(\cdot)}$$

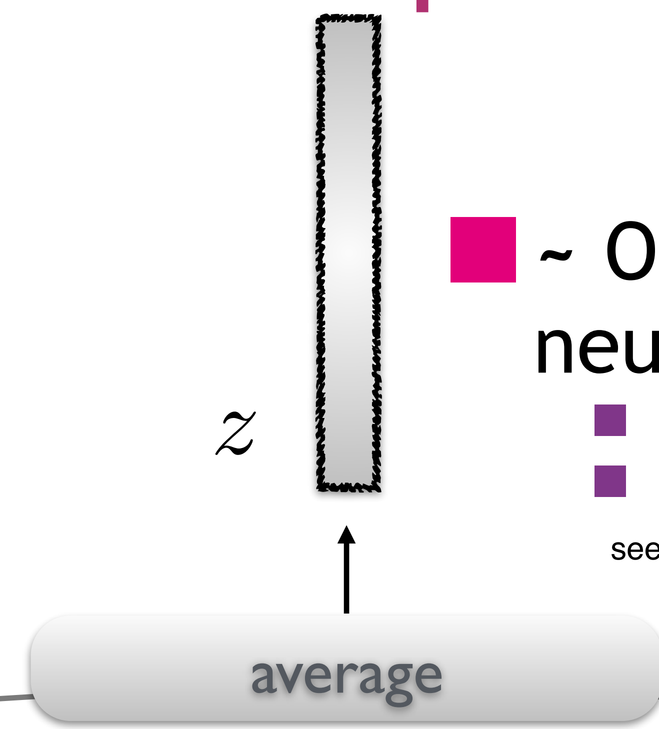


$h(\mathbf{WX})$

$\mathbf{WX}$

$\mathbf{X}$

Computational Aspects



- **Sketching**
  - empirical characteristic function

$$z_\ell = \frac{1}{N}\sum_{i=1}^{N} e^{j w_\ell^\top x_i}$$

- ~ One-layer random neural net
  - Decoding = next layers
  - DNN ~ hierarchical sketching ?

see also [Bruna & al 2013, Giryes & al 2015]

$$h(\cdot) = e^{j(\cdot)}$$

$$\boxed{\mathbf{W}}$$

$$\mathbf{X} \rightarrow \mathbf{W}\mathbf{X} \rightarrow h(\mathbf{W}\mathbf{X})$$

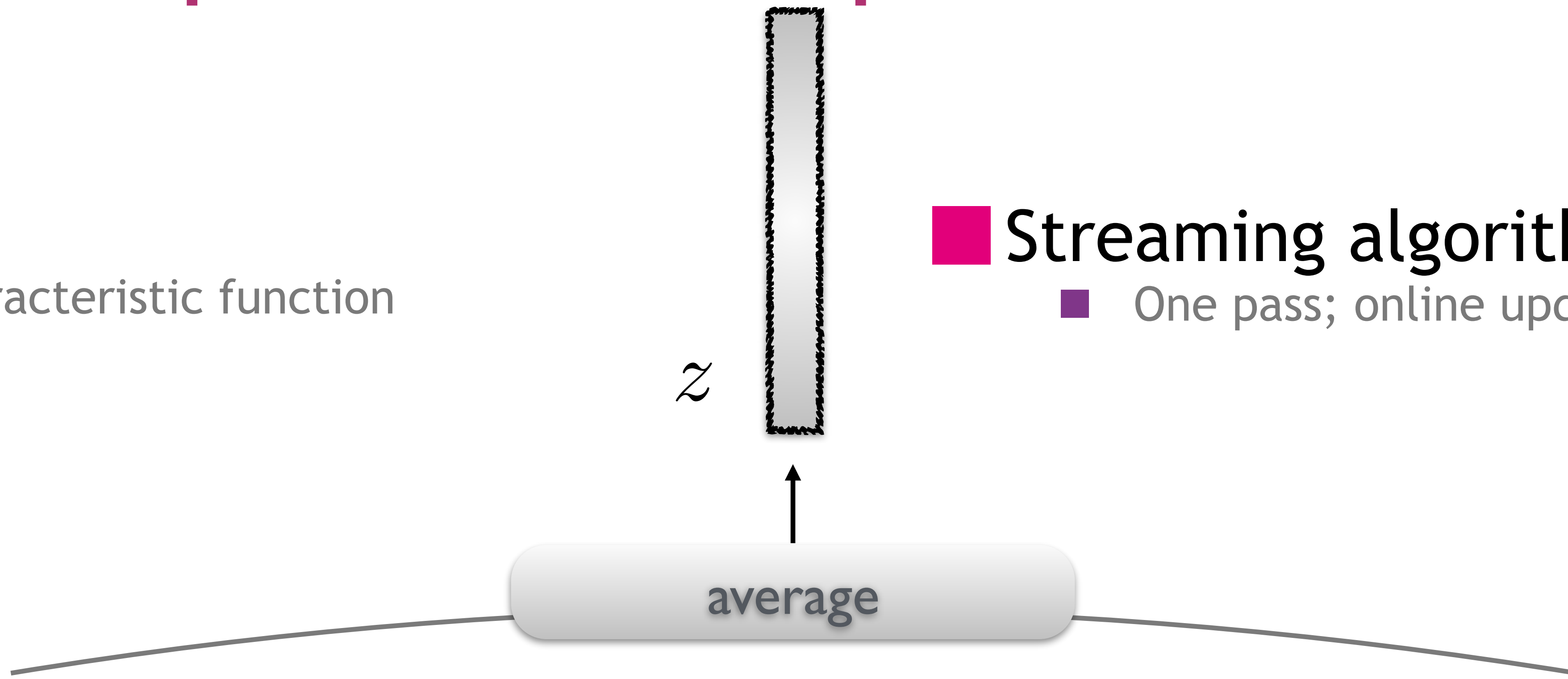
- Privacy-reserving
  - sketch and forget

Computational Aspects

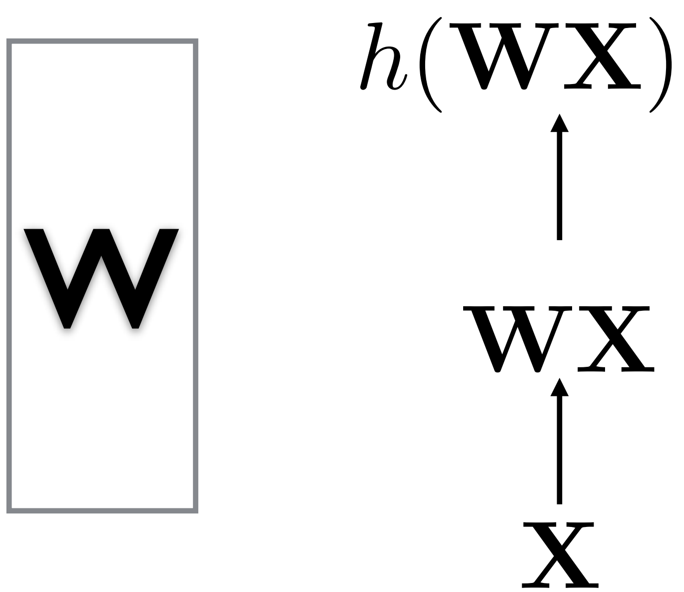
- **Sketching**
  - empirical characteristic function

$$z_\ell = \frac{1}{N}\sum_{i=1}^{N} e^{j w_\ell^\top x_i}$$

- **Streaming algorithms**
  - One pass; online update



$$h(\cdot) = e^{j(\cdot)}$$



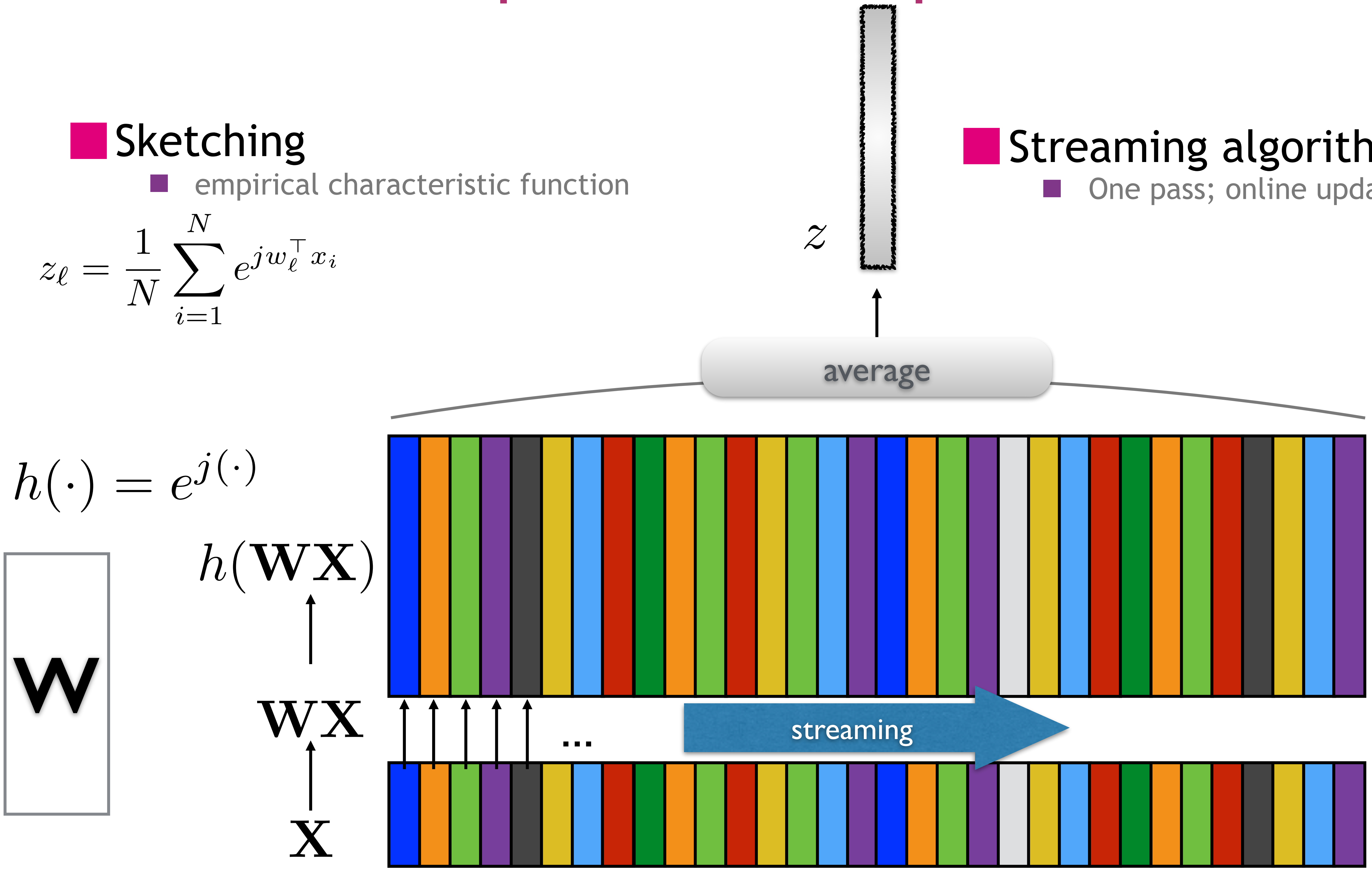
Computational Aspects

- Sketching
  - empirical characteristic function

$$z_\ell = \frac{1}{N}\sum_{i=1}^{N} e^{j w_\ell^\top x_i}$$

- Streaming algorithms
  - One pass; online update

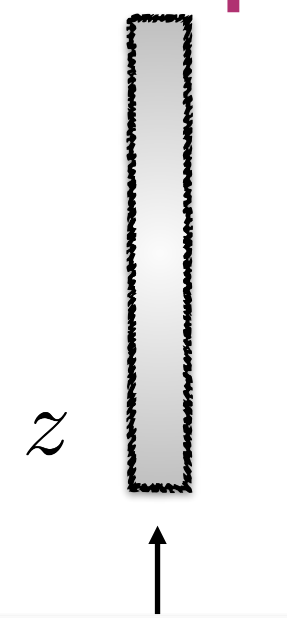
$$h(\cdot) = e^{j(\cdot)}$$



Computational Aspects

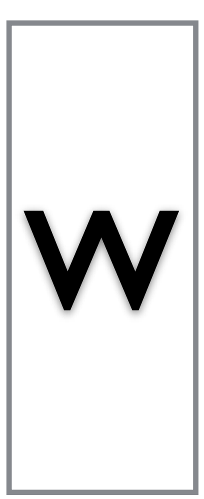
- **Sketching**
  - empirical characteristic function

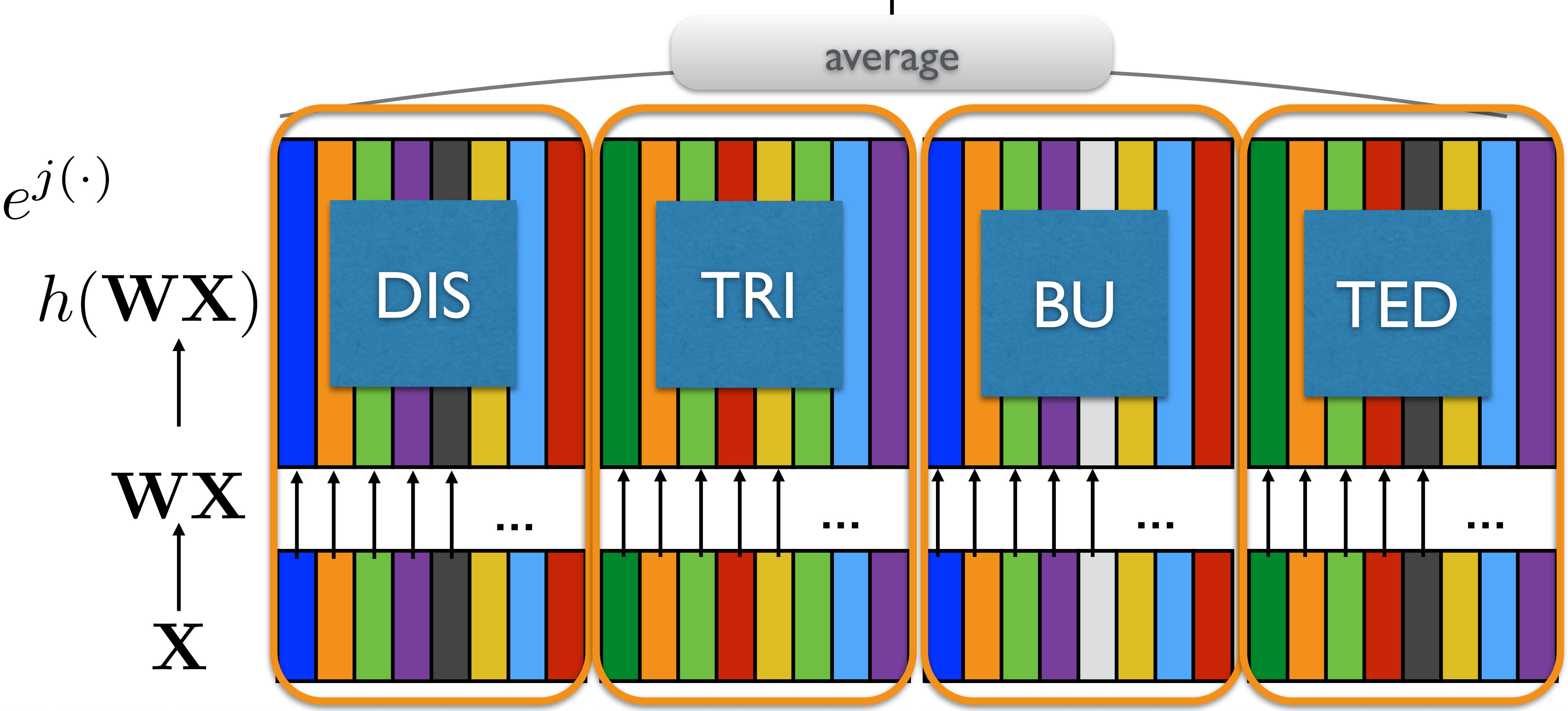
$$z_\ell = \frac{1}{N}\sum_{i=1}^{N} e^{jw_\ell^\top x_i}$$



- **Distributed computing**
  - Decentralized (HADOOP) / parallel (GPU)

$$h(\cdot) = e^{j(\cdot)}$$



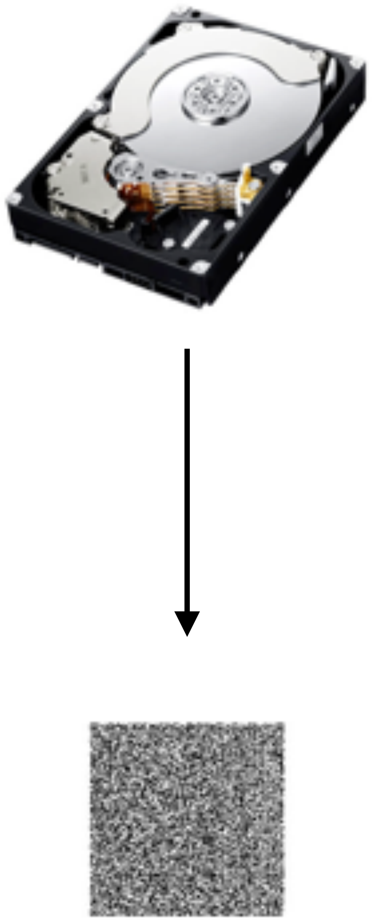


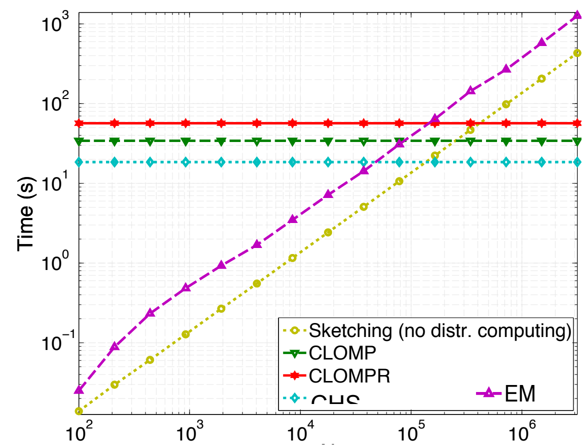
Summary: Compressive K-means / GMM

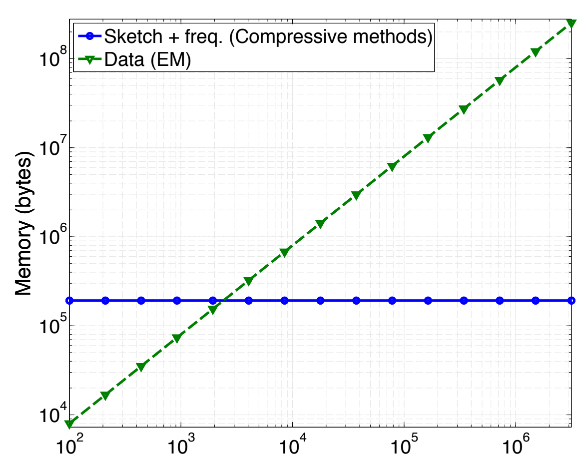
✓ Dimension reduction

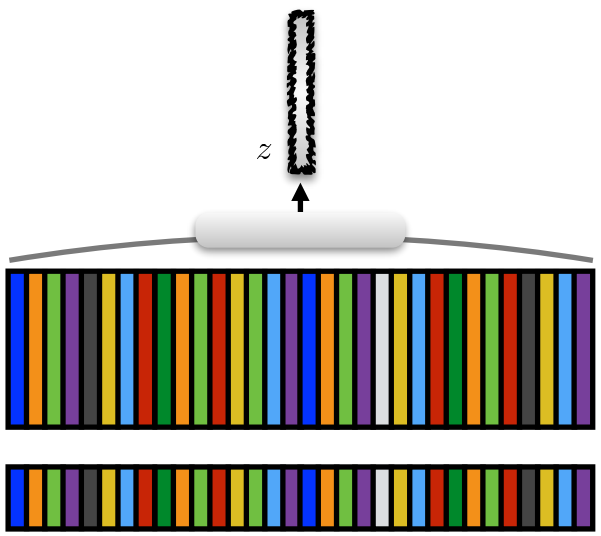
✓ Resource efficiency

✓ Neural net - like









✓ **In the pipe**: information preservation (generalized RIP, "intrinsic dimension")

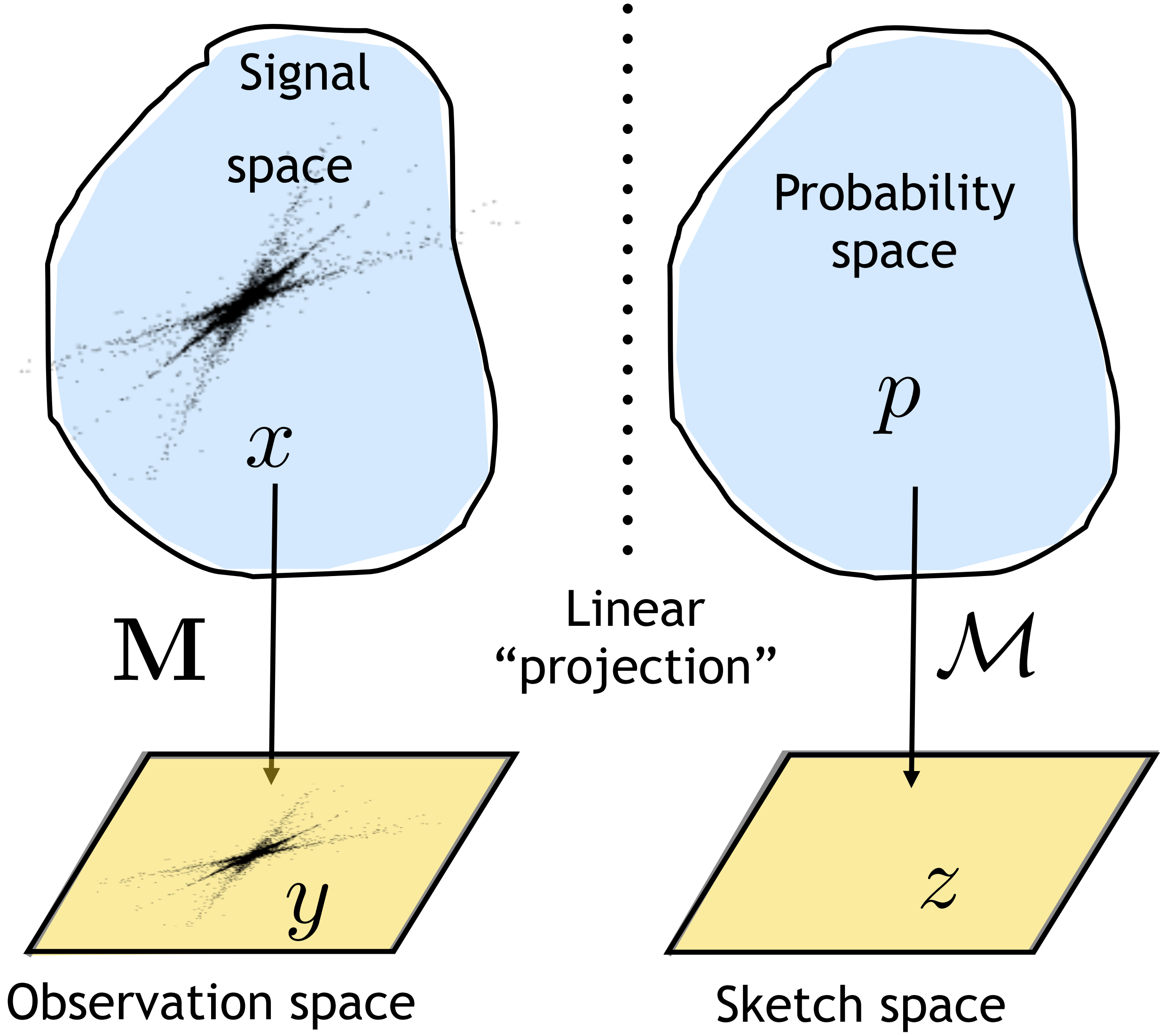
● **Challenge**: provably good recovery algorithms ?

Conclusion

R. GRIBONVAL
London Workshop on Sparse Signal Processing, September 2016

Projections & Learning

- Signal Processing
  - *compressive sensing*
- Machine Learning
  - *compressive learning*



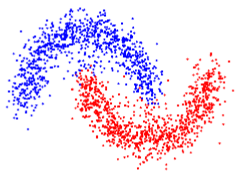
- Reduce dimension of data items
- Reduce **size of collection**

- Compressive sensing
  random projections of data ***items***
- Compressive learning with **sketches**
  random projections of ***collections***
  - *nonlinear in the feature vectors*
  - ***linear*** *in their probability distribution*

Summary

- Challenge: compress $\mathcal{X}$ before learning ?

- **Compressive GMM**
  - Bourrier, G., Perez, ***Compressive Gaussian Mixture Estimation***. ICASSP 2013
  - Keriven & al, ***Sketching for Large-Scale Learning of Mixture Models***. ICASSP 2016 & arXiv:1606.02838
- **Compressive k-means**
  - Keriven & al, ***Compressive K-Means*** *submitted to ICASSP 2017*
- **Compressive spectral clustering** (with graph signal processing)
  - Tremblay & al, ***Accelerated Spectral Clustering using Graph Filtering of Random Signals*** ICASSP 2016
  - Tremblay & al, ***Compressive Spectral Clustering*** ICML 2016 & arXiv:1602.02018
    - ***Ex: with Amazon graph (10^6 edges), 5 times speedup (3 hours instead of 15 hours for k= 500 classes)***



$$O(k^2N) \longrightarrow O(k^2 \log^2 k + N(\log N + k))$$

Recent / ongoing work / challenges

Guarantees ?

- **When is information preserved with sketches / projections ?**
  - Bourrier & al, ***Fundamental perf. limits for ideal decoders in high-dimensional linear inverse problems.*** IEEE Transactions on Information Theory, 2014
    - Notion of Instance Optimal Decoders = **Uniform guarantees**
    - Fundamental role of general **Restricted Isometry Property**
- **How to reconstruct: algorithm / decoder ?**
  - Traonmilin & G., ***Stable recovery of low-dimensional cones in Hilbert spaces - One RIP to rule them all.*** ACHA 2016
    - RIP guarantees for general (convex & nonconvex) regularizers $\Delta(y) := \arg\min_{x \in \mathcal{H}} f(x) \text{ s.t.} \|\mathbf{M}x - y\| \leq \epsilon$
- **How to (maximally) reduce dimension?**
  - ***[Dirksen 2014] :*** *given* a random sub-gaussian linear form
  - Puy & al, ***Recipes for stable linear embeddings from Hilbert spaces to*** $\mathbb{R}$***^m*** arXiv:1509.06947
    - Role of covering dimension / Gaussian width of **normalized secant set**
- **What is the achievable compression for learning tasks ?**
  - ***Compressive statistical learning,*** work in progress with G. Blanchard, N. Keriven, Y. Traonmilin
    - Number of random moments = "intrinsic dimension" of PCA, k-means, Dictionary Learning …
    - Statistical learning: risk minimization + generalization to future samples with same distribution

Laurent Duval
@laurentduval
Suivre

Have a look at genuine #panamapapers team.inria.fr/panama/publica… @RemiGribonval #FreeAdvertising #CastAndCurious

PLEASE

projection, learning and sparsity for efficient data processing

PANAMA

Parsimony and New Algorithms for Audio and Signal Modeling

Presentation | Team members | Papers | Events | Projects | News | Job offers | Software | Contact | Intranet

English Français

Papers

2016

Journal articles

Flexible Multi-layer Sparse Approximations of Matrices and Applications

Luc Le Magoarou, Rémi Gribonval

IEEE Journal of Selected Topics in Signal Processing, IEEE, 2016, <10.1109/JSTSP.2016.2543461>

Random sampling of bandlimited signals on graphs

Gilles Puy, Nicolas Tremblay, Rémi Gribonval, Pierre Vandergheynst

Applied and Computational Harmonic Analysis, Elsevier, 2016, <10.1016/j.acha.2016.05.005>

Fast Robust PCA on Graphs

Nauman Shahid, Nathanael Perraudin, Vassilis Kalofolias, Gilles Puy, Pierre Vandergheynst

IEEE Journal of Selected Topics in Signal Processing, IEEE, 2016, 10 (4), pp.740 – 756, <10.1109/JSTSP.2016.2555239>

News

(closed) PhD offer – Interactive Navigation for a Video Audio True Experience @ PANAMA, Inria Rennes 2016/07/08

Paper + code on random sampling of bandlimited signals on graphs 2016/06/16

2016 Award for Outstanding Contributions in Neural Systems 2016/05/25

(closed) PhD offer – Estimating the Geometry of Audio Scenes Using Virtually-Supervised Learning @ Inria Rennes 2016/05/24

Paper + code on Compressive Spectral Clustering 2016/05/22

PLEASE

projection, learning and sparsity for efficient data processing

THANKS

erc