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

spars2017.lx.it.pt
Random Moments for Compressive Learning

Rémi Gribonval
Inria Rennes - Bretagne Atlantique

remi.gribonval@inria.fr
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
From Compressive Sensing to Compressive Learning

R. Gribonval
London Workshop on Sparse Signal Processing, September 2016
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
Challenging dimensions

\[ \text{Point cloud} = \text{large matrix of feature vectors} \]
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

- High feature dimension $n$
- Large collection size $N$
Challenging dimensions

- Point cloud = large matrix of feature vectors

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

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

Point cloud = large matrix of feature vectors

\[ X \]

\[ Y = M X \]
Compressive Machine Learning?

- Point cloud = large matrix of feature vectors

\[
Y = MX
\]
Compressive Machine Learning?

- **Point cloud = large matrix of feature vectors**

- **Reduce feature dimension**
  - [Calderbank & al 2009, Reboredo & al 2013]
  - (Random) feature projection
  - Exploits / needs low-dimensional **feature model**

\[
X = \begin{bmatrix}
    x_1 \\
    x_2 \\
    \vdots \\
    x_N
\end{bmatrix}
\]

\[
Y = MX
\]

\[
Y = \begin{bmatrix}
    y_1 \\
    y_2 \\
    \vdots \\
    y_N
\end{bmatrix}
\]
Challenges of large collections

Feature projection: limited impact

\[ Y = MX \]
Challenges of large collections

Feature projection: limited impact

\[ Y = MX \]

“Big Data” Challenge: compress collection size
Compressive Machine Learning?

- Point cloud

\[ x \]
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

\[ \chi \xrightarrow{\mathcal{M}} z \in \mathbb{R}^m \]

Sketching operator
- nonlinear in the feature vectors

- 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**

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

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

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

Sketching operator

- **nonlinear in the feature vectors**
- **linear in their probability distribution**
Example: Compressive K-means

\[ \chi \xrightarrow{\mathcal{M}} z \in \mathbb{R}^m \]

\[ N = 1000; n = 2 \]

\[ m = 60 \]

Recovery algorithm

- estimated centroids
- ground truth
Computational impact of sketching

Computation time

Memory

Collection size N

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
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
- method of moments

Signal Processing
- inverse problems

Signal space

Sketch space

Observation space

Signal Processing

inverse problems

Machine Learning
method of moments

Linear "projection"

finite-dimensional Mean Map Embedding, cf Smola & al 2007, Sriperumbudur & al 2010
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?

- **Machine Learning**
  - method of moments
  - compressive learning

- **Signal Processing**
  - inverse problems
  - compressive sensing
Compressive Learning (Heuristic) Examples

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

- Point cloud = empirical probability distribution

\[ x \xrightarrow{\mathcal{M}} z \in \mathbb{R}^m \]

Sketching operator

- Reduce collection dimension ~ sketching

\[ z_\ell = \frac{1}{N} \sum_{i=1}^{N} h_\ell(x_i) \quad 1 \leq \ell \leq m \]

Choosing information preserving sketch?
Example: Compressive K-means

Goal: find $k$ centroids

Standard approach = K-means
Goal: find $k$ centroids

Sketching approach

- $p(x)$ is spatially localized
  - need “incoherent” sampling
  - choose Fourier sampling

Standard approach = K-means
Example: Compressive K-means

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

- **Sketching approach**
  - $p(x)$ is **spatially localized**
  - Need “incoherent” sampling
  - Choose **Fourier** sampling
  - Sample characteristic function
    \[
    z_{\ell} = \frac{1}{N} \sum_{i=1}^{N} e^{j \omega_{\ell}^\top x_i}
    \]
  - 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

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

$N = 1000; n = 2$

$m = 60$
Example: Compressive K-means

Goal: find k centroids

\[ \mathbf{X} \xrightarrow{\mathcal{M}} \mathbf{z} \in \mathbb{R}^m \]
\[ N = 1000; n = 2 \]
\[ m = 60 \]

Density model = mixture of K Diracs
\[ p \approx \sum_{k=1}^{K} \alpha_k \delta_{\theta_k} \]
ground truth

\[ \mathcal{M} \]
Sampled Characteristic Function

\[ N = 1000; n = 2 \]
\[ m = 60 \]
Example: Compressive K-means

Goal: find $k$ centroids

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

$N = 1000; n = 2$

$\mathcal{M}$

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

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

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

CLOMP = Compressive Learning OMP

similar to: OMP with Replacement, Subspace Pursuit & CoSaMP

Density model = mixture of $K$ Diracs

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

ground truth

$\alpha_k, \theta_k$

estimated centroids

$\Delta$

Recovery algorithm = “decoder”
Compressive K-Means: Empirical Results

Training set

\[ x_i \in \mathbb{R}^n \]

N training samples
K clusters

K-means objective

\[ \text{SSE}(\mathcal{X}, C) = \sum_{i=1}^{N} \min_k \| x_i - c_k \|^2 \]
Compressive K-Means: Empirical Results

**Training set**

\[ x_i \in \mathbb{R}^n \]

N training samples
K clusters

**K-means objective**

\[
\text{SSE}(\mathcal{X}, C) = \sum_{i=1}^{N} \min_k \| x_i - c_k \|^2.
\]

**Sketch vector**

\[ z \in \mathbb{R}^m \]

**Matrix of centroids**

\[ C = \text{CLOMP}(z) \in \mathbb{R}^{n \times K} \]

**Relative memory**

**Relative time of estimation**

**Relative SSE**

\[ \text{N=10}^4, \text{ N=10}^5, \text{ N=10}^6, \text{ N=10}^7 \]
Compressive K-Means: Empirical Results

Training set

\[ x_i \in \mathbb{R}^n \]

N training samples
K clusters

K-means objective

\[
\text{SSE}(\mathcal{X}, C) = \sum_{i=1}^{N} \min_{k} \|x_i - c_k\|^2.
\]

Sketch vector

\[ z \in \mathbb{R}^m \]

Matrix of centroids

\[ C = \text{CLOMP}(z) \in \mathbb{R}^{n \times K} \]
Compressive K-Means: Empirical Results

Training set \( x_i \in \mathbb{R}^n \)

Spectral features

Sketch vector \( z \in \mathbb{R}^m \)

Matrix of centroids \( C = \text{CLOMP}(z) \in \mathbb{R}^{n \times K} \)

N training samples

K=10 clusters

K-means objective

\[
\text{SSE}(\mathcal{X}, C) = \sum_{i=1}^{N} \min_k \| x_i - c_k \|^2.
\]

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 $(\Theta, \alpha)$

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

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

$N = 60,000,000$; $n = 12$

$M = 5,000$
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
  
  \[ N = 60\,000\,000 \]
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**

\[ N = 60 \, 000 \, 000 \]

**Maximum size manageable by EM**

\[ N = 300 \, 000 \]
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

\[ N = 60\,000\,000 \]

Maximum size manageable by EM

\[ N = 300\,000 \]

\( K=64, N_{CHS} = N_{EM} = 3 \times 10^5 \)
Proof of Concept: Speaker Verification Results (DET-curves)

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

  \[ N = 300\,000\,000 \]

- **After silence detection**

  \[ N = 60\,000\,000 \]
  for CHS

- **Maximum size manageable by EM**

  \[ N = 300\,000 \]
  for EM

\(~50\) Gbytes
~ 1000 hours of speech

\[ K=64, N_{CHS} = 200*N_{EM} = 6.10^7 \]
Proof of Concept: Speaker Verification Results (DET-curves)

~ 50 Gbytes
~ 1000 hours of speech

$K=64, N_{\text{CHS}} = 200*N_{\text{EM}} = 6.10^7$

$\begin{align*}
m= 500 & \quad 7 \text{ 200 000-fold compression} \\
m= 1000 & \quad 3 \text{ 600 000-fold compression} \\
m= 5000 & \quad 720 \text{ 000-fold compression} & \text{exploit whole collection} & \text{improved performance}
\end{align*}$
Computational Efficiency
Computational Aspects

**Sketching**
- empirical characteristic function

\[ z_\ell = \frac{1}{N} \sum_{i=1}^{N} e^{jw^\top \ell x_i} \]
Computational Aspects

**Sketching**

- empirical characteristic function

\[ z_\ell = \frac{1}{N} \sum_{i=1}^{N} e^{jw_\ell^T x_i} \]
Computational Aspects

**Sketching**

- empirical characteristic function

\[ z_\ell = \frac{1}{N} \sum_{i=1}^{N} e^{jw_\ell^T x_i} \]
Computational Aspects

**Sketching**

- empirical characteristic function

\[ z_\ell = \frac{1}{N} \sum_{i=1}^{N} e^{j w_\ell^T x_i} \]

\[ h(\cdot) = e^{j(\cdot)} \]

\[ h(WX) \]

\[ WX \]

\[ X \]
Computational Aspects

**Sketching**
- empirical characteristic function

\[ z_\ell = \frac{1}{N} \sum_{i=1}^{N} e^{jw_ell^T x_i} \]

\[ h(\cdot) = e^{j(\cdot)} \]

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

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

**Sketching**
- empirical characteristic function

\[ z_\ell = \frac{1}{N} \sum_{i=1}^{N} e^{j w_\ell^T x_i} \]

\[ h(\cdot) = e^{j(\cdot)} \]

\[ h(WX) \]

\[ WX \]

\[ X \]

**Privacy-preserving**
- sketch and forget

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

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

**Sketching**
- empirical characteristic function

\[
z_\ell = \frac{1}{N} \sum_{i=1}^{N} e^{j w_\ell^T x_i}
\]

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

\[h(\cdot) = e^{j(\cdot)}\]

\[h(WX)\]

\[WX\]

\[X\]
Computational Aspects

**Sketching**
- empirical characteristic function

\[ z_\ell = \frac{1}{N} \sum_{i=1}^{N} e^{jw_\ell^T x_i} \]

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

\[ h(\cdot) = e^{j(\cdot)} \]

\[ h(WX) \]

\[ WX \]

\[ X \]
Computational Aspects

**Sketching**
- empirical characteristic function

\[ z_\ell = \frac{1}{N} \sum_{i=1}^{N} e^{j w_\ell^T x_i} \]

\[ h(\cdot) = e^{j(\cdot)} \]

\[ h(WX) \]

**Distributed computing**
- Decentralized (HADOOP) / parallel (GPU)
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
Projections & Learning

- Signal Processing
  - compressive sensing

- Machine Learning
  - compressive learning

- Signal space
  - $x$
  - $M$
  - Linear “projection”

- Observation space
  - $y$

- Probability space
  - $p$
  - $M$

- Sketch space
  - $z$

- 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

- **Compressive k-means**
  - Keriven & al, *Compressive K-Means* submitted to ICASSP 2017

- **Compressive spectral clustering** (with graph signal processing)

- Ex: with Amazon graph ($10^6$ edges), 5 times speedup (3 hours instead of 15 hours for $k=500$ classes)

$$O(k^2 N) \longrightarrow O(k^2 \log^2 k + N(\log N + k))$$
Recent / ongoing work / challenges

Guarantees?

- When is information preserved with sketches / projections ?
    - 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

- 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
please
projection, learning and sparsity for efficient data processing

PANAMA
Parsimony and New Algorithms for Audio and Signal Modeling

Papers
2016

Journal articles
Flexible Multi-layer Sparse Approximations of Matrices and Applications
Luc Le Magaorou, 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 Perraudeau, Vassilis Kalofolias, Gilles Puy, Pierre Vandergheynst
IEEE Journal of Selected Topics in Signal Processing, IEEE, 2016, 10 (8), pp 740 – 756, <10.1109/JSTSP.2016.2560290>