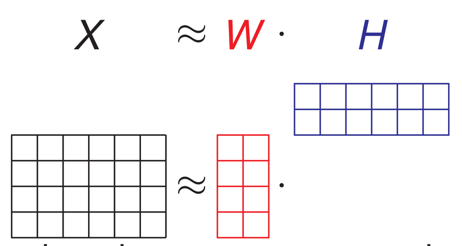
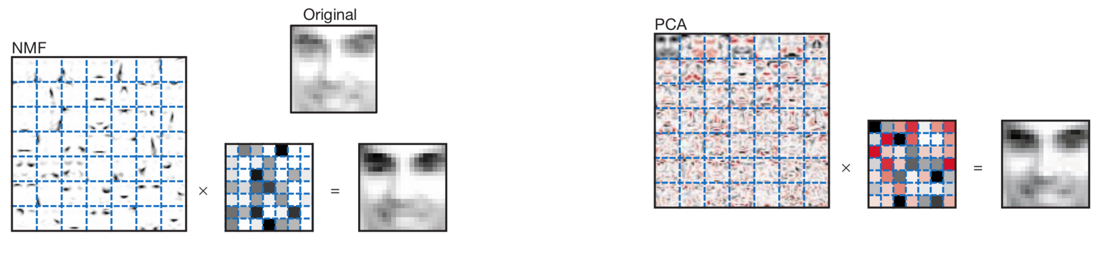
# Random Projection Streams: a New Framework for Compressive Learning

Farouk YAHAYA[1], Matthieu PUIGT, Paul DUCOROY[2], Gilles DELMAIRE, and Gilles ROUSSEL Univ. Littoral Côte d'Opale, LISIC, F-62228 Calais, France [1] Ph.D. student ; [2] student also affiliated with EILCO

## Context




- **Nonnegative Matrix Factorization (NMF)** is a popular tool in Signal/Image Processing and Machine Learning
- Goal: estimate two nonnegative $n \times p$ and $p \times m$ matrices $W$ and $H$ such that an observed low-rank nonnegative $n \times m$ matrix $X$ can be written as $X \approx W \cdot H$
- Some applications:
  - Source separation, dictionary learning, graph analysis, topic modelling, hyperspectral unmixing...
- Why is NMF so popular? Better interpretability than no-sign-constrained approaches




NMF and PCA applied to face dataset (source: Lee and Seung, 1999)

- How is NMF working?
  - Iterative procedure where $W$ and $H$ are alternatingly updated
  - Historical techniques known to be slow (multiplicative updates, projected gradients, nonnegative least squares, etc)
- NMF and Big Data: How to face the data deluge?
  - Distributed computing (e.g., Liu *et al.*, 2010)
  - Online factorization (e.g., Mairal *et al.*, 2010)
  - Fast solver (e.g., Guan et al., 2020)
  - Randomized strategies (e.g., Wang *et al.*, 2010, Tepper and Sapiro, 2016)
    - ➪ **Dimensionality reduction of $X$ by right (resp. left) multiplication with $R$ (resp. $L$)**

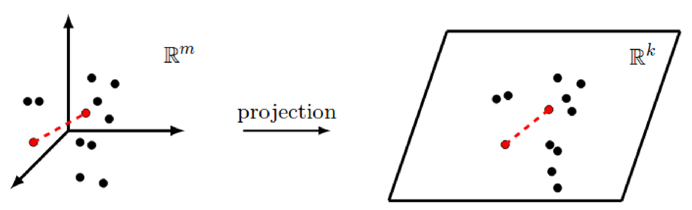
$$\underbrace{X \cdot R}_{X_R} \approx W \cdot \underbrace{H \cdot R}_{H_R} \qquad \underbrace{L \cdot X}_{X_L} \approx \underbrace{L \cdot W}_{W_L} \cdot H$$


## How to design $R$ and $L$

- **Random projections** is a popular tool in machine learning to speed-up computations while preserving pairwise structure
- Mathematical foundations based on the **Johnson-Lindenstrauss Lemma**

  Given $0 < \varepsilon < 1$, a set $X$ of $n$ points in $\mathbb{R}^m$, and a number $k > 8\ln(n)/\varepsilon^2$, there is a linear map $f : \mathbb{R}^m \to \mathbb{R}^k$ such that :

$$\forall u, v \in X, \quad (1-\varepsilon)\|u-v\|^2 \leq \|f(u)-f(v)\|^2 \leq (1+\varepsilon)\|u-v\|^2$$




Principles of the random projections (source: Erichson *et al.*, 2019)

- Existing random projections techniques (possibly) applied to NMF:
  1. Gaussian Compression (GC): Gaussian random matrices as projection matrices (Zhou *et al.*, 2012)
     - Simple but computationally demanding
  2. CountSketch (Clarkson & Woodruff, 2017): $L$ is sparse where each column has only a single non-zero entry chosen in a uniformly random position with value randomly equal to $\pm 1$
     - Faster than GC but requires less compression to provide the same performance
  3. CountGauss (Kapralov *et al.*, 2016): combination of GC and CountSketch
     - Same performance as GC but faster to compute
  4. (Very) Sparse Random Projections ( (V)SRP, Achlioptas, 2001, Li *et al.*, 2006):

$$L_{ij} = \sqrt{s} \cdot \begin{cases} 1 & \text{with prob. } 1/(2s), \\ 0 & \text{with prob. } (s-1)/s, \quad \text{with } s = 3 \text{ (SRP) or } s \gg 3 \text{ (VSRP)} \\ -1 & \text{with prob. } 1/(2s), \end{cases}$$

     - Cheap to compute and assymptocally equivalent to GC
  5. *Structured* random projections aka Randomized Power Iterations (RPIs, Tepper & Sapiro, 2016): $L = \text{QR}((XX^T)^q \cdot X \cdot \Omega_L)^T$
     - Structured random projections (SRP) data-dependent compression
     - Was found to be much more accurate than GC in practice
     - But much more expensive than GC to compute

## Problem Statement

- SotA data-independent compression can be efficiently implemented
- But is less accurate than SRP in practice (Tepper & Sapiro, 2016)
- However SRP is very expensive
- We aim to propose a data-independent alternative to SRP providing the same performance

## Random projection streams

- **Recall the JLL:** there is a linear projection operator which maps the data to $\mathbb{R}^k$ and which preserves the pairwise distances up to a distorsion parameter $\varepsilon$
- **Applied to NMF:** $k \triangleq p + \nu$ where $\nu$ is a user-defined value
  - ↘ $\nu$ ➪ ↗ $\varepsilon$ but ↘ computational cost
  - ↗ $\nu$ ➪ ↘ $\varepsilon$ but ↗ computational cost
- We assume:
  - $L$ and $R$ are drawn according to a SotA data-independent compression and cannot fit in memory.
  - These matrices to be observed in a streaming fashion.
  - At each NMF iteration compression sub-matrices $L^{(i)}$ and $R^{(i)}$ are considered.
  - $W$ and $H$ updated using different compressed matrices $X_R{}^{(i)}$ and $X_L{}^{(i)}$, resp.

**Require:** initial matrices $W$, $H$, $i = 0$
  **repeat**
    Update $i = i + 1$ and get $L^{(i)}$ and $R^{(i)}$
    Define $X_R^{(i)} \triangleq X \cdot R^{(i)}$ and $X_L^{(i)} \triangleq L^{(i)} \cdot X$
    **for** counter $= 1$ **to** $\omega$ **do**
      Define $H_R^{(i)} \triangleq H \cdot R^{(i)}$ and $W_L^{(i)} \triangleq L^{(i)} \cdot W$
      Solve $\min_{W \geq 0} \|X_R^{(i)} - W H_R^{(i)}\|_{\mathcal{F}}$
      Solve $\min_{H \geq 0} \|X_L^{(i)} - W_L^{(i)} H\|_{\mathcal{F}}$
    **end for**
  **until** a stopping criterion

➪ We derive streamed versions of SotA data-independent compression techniques
- GC Streams (GCS), CountSketch Streams (CountSketchS), CountGauss Streams (CountGaussS), (V)SRP Streams ((V)SRPS)...

## Random Projection Streams for Compressive Weighted NMF

- In many problems, observed data matrix $X$ with **missing entries or confidence measures** associated to each entry
  - Some applications: collaborative filtering, source apportionment, low-rank nonnegative matrix completion, mobile sensor calibration

➪ Weighted NMF (WNMF):

$$\min_{W,H \geq 0} \|Q \circ X - Q \circ (W \cdot H)\|_{\mathcal{F}}$$

➪ **Compressive Weighted NMF (Yahaya *et al.*, 2019)**
- EM procedure
- E-step: we complete $X$ from the last estimates of $W$ and $H$

$$X^{\text{comp}} = Q \circ X + (\mathbb{1}_{n,m} - Q) \circ (W \cdot H),$$

  where $\mathbb{1}_{n,m}$ is the $n \times m$ matrix of ones.
- M-step: we update $W$ and $H$ using a compressive framework

!!! Computing $R$ and $L$ done at each E-step
- ➪ Computing RPIs/RSIs is the bottleneck of the framework

➪ Random Projection Streams for WNMF

**Require:** initial matrices $W$ and $H$
  **repeat**
    **for** counter $= 1$ **to** $\omega$ **do**
      Update $i = i + 1$ and get $L^{(i)}$ and $R^{(i)}$
      {**E-step**}
      Compute $X^{\text{comp}}$ as above
      Define $X_R^{(i)} \triangleq X^{\text{comp}} \cdot R^{(i)}$ and $X_L^{(i)} \triangleq L^{(i)} \cdot X^{\text{comp}}$
      {**M-step**}
      **for** compt=1 **to** $\text{Max}_{\text{OutIter}}$ **do**
        Define $H_R^{(i)} \triangleq H \cdot R^{(i)}$ and $W_L^{(i)} \triangleq L^{(i)} \cdot W$
        Solve $\min_{W \geq 0} \|X_R^{(i)} - W H_R^{(i)}\|_{\mathcal{F}}$
        Solve $\min_{H \geq 0} \|X_L^{(i)} - W_L^{(i)} H\|_{\mathcal{F}}$
      **end for**
    **end for**
  **until** a stopping criterion

## Experiments

- We draw random nonnegative matrices $W$ and $H$ with $n = m = 10000$ and $p = 5$.
- Two NMF solvers, *i.e.*, Nesterov gradient (NeNMF – Guan *et al.*, 2012) and Active Set (AS-NMF, Kim and Park, 2008)
- We apply three different compression strategies, *i.e.*, RSIs, GC and RPS (mainly GCS in the poster)
- For NMF:
  - we repeat each test 15 times to get statistics on the performance
  - we investigate the influence of $\nu$ and $\omega$ on the performance
  - The performance criterion used in this paper is a Relative Reconstruction Error (RRE)

$$\text{RRE} \triangleq ||X - W \cdot H||_{\mathcal{F}}^2 \,/\, ||X||_{\mathcal{F}}^2$$

- For WNMF:
  - We randomly remove some data (from 10 to 90%)
  - We apply the proposed techniques during 1 min and we repeat each test 15 times
  - We use the RRE computed over the theoretical full matrix $X^{\text{theo}}$

## Results

- Influence of $\omega$ and $\nu_l$

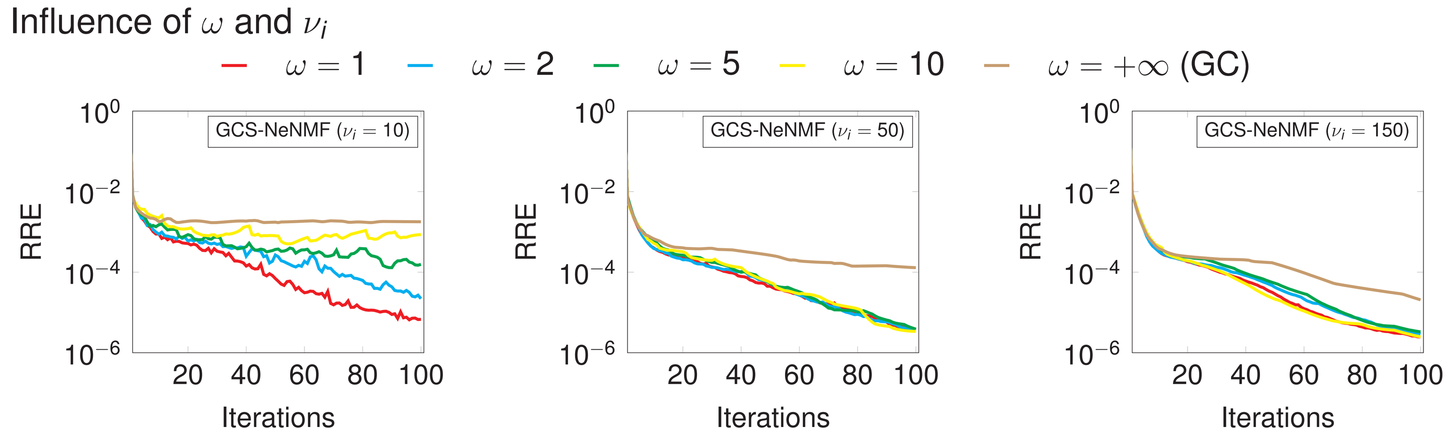



Figure: NMF performance for different parameters of the GCS strategy.

- Influence of the choice of the compression technique

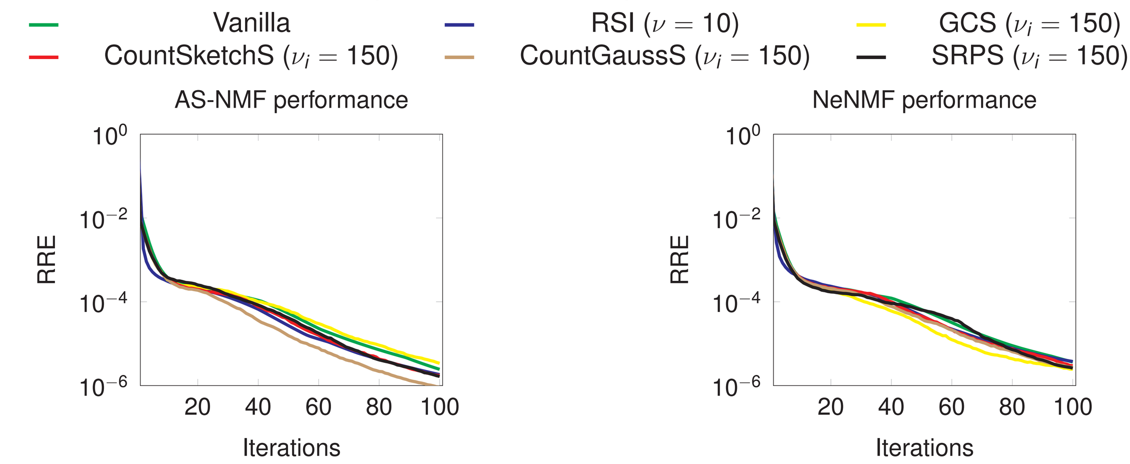



Figure: Influence of compression techniques on NMF performance along iterations.

- Performance in the weighted case

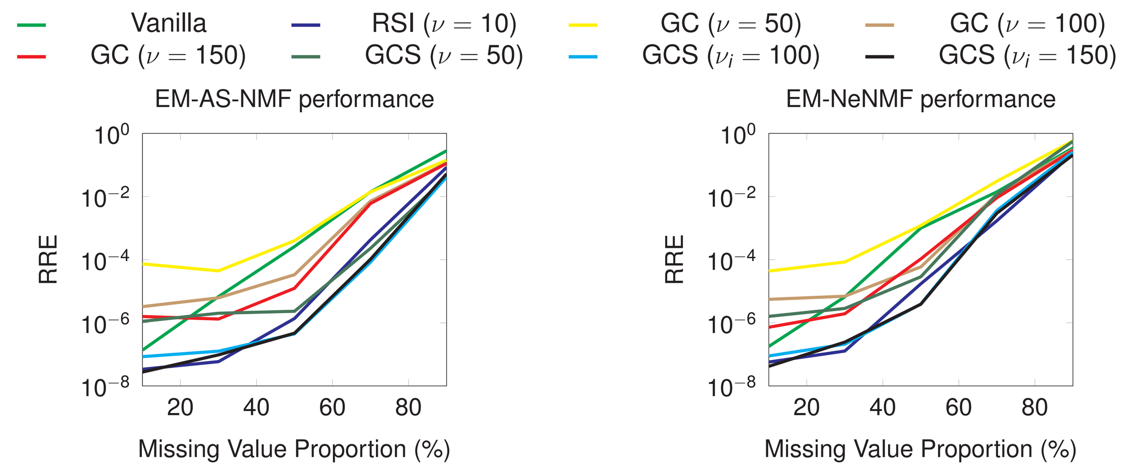



Figure: WNMF performance vs the missing value proportion.

## On-going work

- Computational cost of RPS might be prohibitive with CPU.
  - ➪ It linearly increases with the number of NMF iterations
- On-going work to investigate the use of a **photonics computing processors** to perform RPS
  - ➪ Very fast and low-energy computations of GC.
  - In collaboration with the LightOn company.
- Investigation of the enhancement of RPS for other machine learning approaches.

## Summary

- We proposed random projection streams (RPS), i.e., a data-independent alternative strategy to data-dependent random projections (SRP)
- Our strategy can be applied to Weighted NMF as well
- We aim to investigate GCS to other compressive learning problems
- We aim also to apply the proposed strategy to informed and structured NMF techniques for mobile sensor calibration (**TI Dunkerque**) and hyperspectral imaging (**LISIC Longuenesse activities, SFR Campus de la Mer, CNES Osynico project**)

## Acknowledgements

This work was partially funded by the Région Hauts-de-France. The experiments were performed using the CALCULCO computing platform supported by SCoSI/ULCO. We also acknowledge LightOn to freely provide its cloud computing platform for this work.

ulco Université Littoral Côte d'Opale — Région Hauts-de-France — LightOn We bring Light to AI