A Novel and Efficient Compressed Algorithm for Non-Negative Matrix Factorization (NMF)

Gabriele Torre, Michael Graber & Martin Melchior
University of Applied Sciences and Arts Northwestern Switzerland
gabriele.torre@fhnw.ch michael.graber@fhnw.ch martin.melchior@fhnw.ch

Abstract

Non-negative matrix factorization (NMF) is one of the most popular decomposition techniques for multivariate data. NMF is a core method for many machine-learning related computational problems, such as data compression, feature extraction, word embedding, recommender systems etc. In practice, however, its application is facing challenges in recent years. These challenges stem from the fact that the datasets available are ever-growing in size. For large datasets, NMF algorithm efficiency poses demands on data loading and distribution into and within the available memory and on communication between computational nodes.
Here we present a novel and efficient compressed NMF algorithm. Our algorithm applies a random compression scheme to drastically reduce the dimensionality of the problem while preserving most of the action of the data matrix and inherently limiting memory needs and communication load. As a consequence our algorithm supersedes existing methods in speed but it matches the best non-compressed algorithms in reconstruction precision.

Introduction

A common task in *Machine Learning* applications is to decompose a high dimensional dataset into lower dimensional feature vectors. A standard method for this task is Principal Component Analysis (PCA) *(Jolliffe, 2002)*. PCA allows to find basis vectors along whose directions a given dataset shows biggest variance and thus to capture as much variance of the data with as few components as possible. Independent Component Analysis (ICA) *(Bell and Seijnowski, 1997)* on the other hand finds basis vectors that are statistically independent.

Non-negativity is a natural property of any count-based measurement; e.g. photon-counts in the case of images in astronomy or medical imaging and word-counts in text analysis. The non-negativity of the data naturally proposes a decomposition into non-negative components. PCA and ICA methods do not retain this property in the feature vectors.

The popularity of the Non-negative Matrix Factorization (NMF) approach stems essentially from three properties that distinguish it from standard decomposition techniques. Firstly, the matrix factors are by definition non-negative, which allows their intuitive interpretation as real underlying components within the context defined by the original data. Secondly, NMF implementations can be easily tweaked to produce sparse results, which provide a more compact and local representation, emphasizing even more the features based decomposition of the data. Finally, unlike other decomposition methods such as PCA or ICA, NMF does restrict components to be orthogonal or independent, which is often desirable.

For those reasons the range of applications of NMF spans over many different fields such as:

- **Astronomy and Gravitation**: Blind-Deconvolution and Blind-Source-Separation.
- **Neuroscience**: Simultaneous segmentation and extraction of neural activity.
- **Recommender Systems**: *rating* or *preference* prediction in business applications.
- **Topic Classification of Text Documents**.

Model and Problem Statements

For a given non-negative matrix $X \in \mathbb{R}^{d\times n}$ (d being the number of dimensions, n the number of datapoints) and a desired number of components $k << min(d,n)$, NMF searches for the non-negative factors $A \in \mathbb{R}^{d\times k}$ and $B \in \mathbb{R}^{n\times k}$ that approximate X as:

$$X \sim AB^T \,. \tag{1}$$

Choosing the *Frobenius* norm $\|\cdot\|_F$ of the residual matrix as our cost function, we can formulate the global optimization problem to be solved as follows:

$$\underset{A,A\geq 0;\ B,B\geq 0}{\text{minimize}} \qquad J(A,B) = \|X - AB^T\|_F^2$$

The non-convexity of this *global* optimization problem in A and B compromises the *uniqueness* of the solution. The values for A and B corresponding to a *local* minimum of $J(A,B)$ can be computed by means of algorithms that exploit a *block-coordinate descent* approach (*Lee and Seung, 1999*).

Optimization Methods

Block-coordinate descent algorithms update cyclically blocks of variables only, while keeping the remaining variables fixed. Given that the global constraints are the cartesian product of convex sets on each block of variables, the resulting sequence is guaranteed to converge to a stationary point *(Bertsekas, 1995)*.

Since in NMF the overall non-negativity constraints are in fact the cartesian product of the non-negativity constraints on the individual variables, the NMF problem can be tackled in a block-coordinate descent approach. The resulting basic structure for NMF algorithms can be formulated as follows:

loop (i)

$$\underset{A,A\geq 0}{\text{minimize}} \quad J(A) = \|X - AB^T\|_F^2 \ \text{ with } B \text{ fixed}$$
$$\underset{B,B\geq 0}{\text{minimize}} \quad J(B) = \|X - AB^T\|_F^2 \ \text{ with } A \text{ fixed}$$

Algorithms employing this approach are e.g. the *Multiplicative Updates rule* (MU) by Lee and Seung, the *Active-Set-Like* method by Kim and Park or *Projected Gradient Descent* NMF by Lin.
However, these methods are *computationally expensive* and associated to a *slow convergence rate (Kim et al., 2014)*. In fact, memory needs scale with $\mathcal{O}(dn + dk + nk)$ and number of computations per update with $\mathcal{O}(8dnk)$.

Hierarchical Optimization Methods

Cichocki and collegues *(Cichocki et al., 2007 & 2009)* proposed two hierarchical alternating least squares (HALS) algorithms that reduce the block size in the block-coordinate descent NMF approach to an individual column a_j of A and b_j of B, leading to the following overall optimization algorithm:

loop (i & j)

$$\underset{a_j,a_j\geq 0}{\text{minimize}}\ J_A^{(j)}(a_j) = \frac{1}{2}\|X^{(j)} - a_j b_j^T\|_F^2 \ \text{ for fixed } b_j\,,$$
$$\underset{b_j,b_j\geq 0}{\text{minimize}}\ J_B^{(j)}(b_j) = \frac{1}{2}\|X^{(j)} - a_j b_j^T\|_F^2 \ \text{ for fixed } a_j,$$

where $X^{(j)} = \sum_{i\neq j} X - a_i b_i^T$.

The two algorithm versions, HALS and FastHALS, provide more accurate reconstructions with a higher convergence rate. Their computation time now scales respectively with $\mathcal{O}(8dnk)$ and $\mathcal{O}(4dnk)$. However, a critical issue that remains for large datasets is that they require the entire data matrix X to be held in memory. Hence the memory needs still scale with $\mathcal{O}(dn + dk + nk)$.

Randomly Compressed Hierarchical Alternating Least Squares

To address the large memory requirements of the optimization approaches described above, random compression steps can be introduced to reduce the dimensionality of the optimization problems. Tepper and Sapiro (*2015*) apply *structured random compression* by Halko et al. *(2011)* to the two-block-coordinate descent approach (MU-RP). However, the reconstruction precision of the algorithm suggested is not comparable to the uncompressed ones in our experience.

We therefore propose here to apply the structured random compression method to the HALS and FastHALS algorithms. The method consists of the following steps:

Structured Random Compression Precompute two data projection matrices L and R:

1. *Weighted Random Projection:* $P(X) = \left(XX^T\right)^w X\Omega$, with $\Omega \in \mathcal{N}(0,1)^{(r+r_{ov})\times d/n}$ and $w \in \mathbb{N}_+$
2. *Compute Orthonormal Basis:* Find orthonormal basis $L \in \mathbb{R}^{(r+r_{ov})\times m}$ of $P(X)$ and $R \in \mathbb{R}^{(r+r_{ov})\times n}$ of $P(X^T)$ via QR-decomposition where $r_{ov} \in \mathbb{N}^+$ is an oversampling parameter.

Compressed HALS and fast HALS To reduce the dimensionality of the problem, the optimization steps of HALS and fast HALS have to be modified:

loop (i & j)

$$\underset{a_j,a_j\geq 0}{\text{minimize}}\ J_A^{(j)}(a_j) = \frac{1}{2}\|XR^T - a_j B^T R^T\|_F^2 = \frac{1}{2}\|\hat{X} - a_j \hat{B}^T\|_F^2 \ \text{ for fixed } B$$
$$\underset{b_j,b_j\geq 0}{\text{minimize}}\ J_B^{(j)}(b_j) = \frac{1}{2}\|LX - LAB_j^T\|_F^2 \quad = \frac{1}{2}\|\hat{X} - \hat{A}b_j^T\|_F^2 \ \text{ for fixed } A$$

The computational complexity of the single iteration is hereby reduced to $\mathcal{O}(2dk(r+r_{ov})) << \mathcal{O}(4dnk)$ if compared to the FastHALS algorithm. In addition to that, the memory needs for the variables required inside the optimization loop scale with $\mathcal{O}((2(r+r_{ov})+k)(d+n)) << \mathcal{O}(dn+dk+nk)$ if compared to the standard NMF methods.

Experiments

We tested our new algorithm on the Olivetti Faces dataset and the 20 Newsgroup dataset. The first one is composed of faces of 40 distinct subjects with 10 images each. The images measure 64×64 pixels and are quantized to 256 grey levels. The second dataset is composed by 20000 newsgroup documents partitioned almost evenly across 20 newsgroup classes. In the example shown here we extracted respectively $k = 20$ components for the Olivetti dataset and $k = 60$ for the 20 Newsgroup dataset providing a performance comparison of our algorithm with other state-of-the art NMF algorithms.

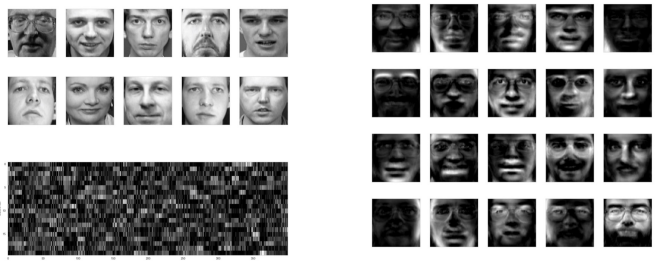
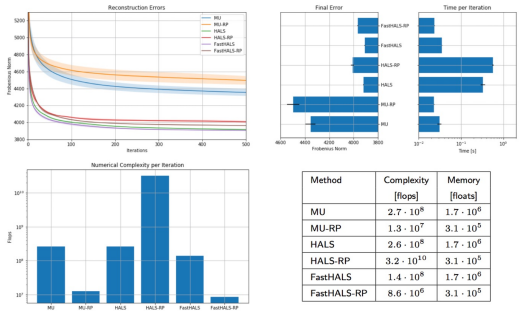
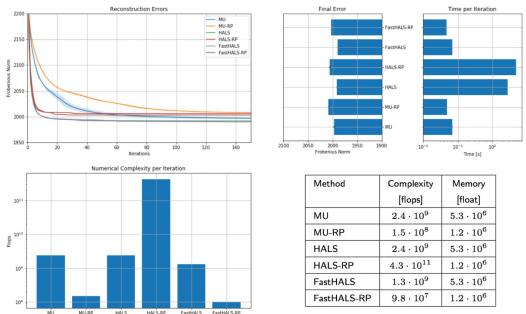


Figure 1: *Top Right:* A sample set of images from the Olivetti Faces dataset. *Bottom Left:* The extracted features, i.e. matrix B. *Right:* The extracted image components, i.e. columns of matrix A, reshaped into images of 64×64 pixels.



Method	Complexity [flops]	Memory [floats]
MU	$2.7 \cdot 10^8$	$1.7 \cdot 10^6$
MU-RP	$1.3 \cdot 10^7$	$3.1 \cdot 10^5$
HALS	$2.6 \cdot 10^8$	$1.7 \cdot 10^6$
HALS-RP	$3.2 \cdot 10^{10}$	$3.1 \cdot 10^5$
FastHALS	$1.4 \cdot 10^8$	$1.7 \cdot 10^6$
FastHALS-RP	$8.6 \cdot 10^6$	$3.1 \cdot 10^5$

Figure 2: Performances comparison of the NMF methods applied on the *Olivetti faces dataset*. *Top Left:* iterative evolution of the reconstruction errors. *Top Right:* Final reconstruction error (left) and computational time for single iteration. *Bottom:* Numerical complexity and memory consumption of the NMF methods.



Method	Complexity [flops]	Memory [float]
MU	$2.4 \cdot 10^9$	$5.3 \cdot 10^6$
MU-RP	$1.5 \cdot 10^8$	$1.2 \cdot 10^6$
HALS	$2.4 \cdot 10^9$	$5.3 \cdot 10^6$
HALS-RP	$4.3 \cdot 10^{11}$	$1.2 \cdot 10^6$
FastHALS	$1.3 \cdot 10^9$	$5.3 \cdot 10^6$
FastHALS-RP	$9.8 \cdot 10^7$	$1.2 \cdot 10^6$

Figure 3: NMF methods performances applied on the *20 Newsgroup dataset*. *Top Left:* iterative evolution of the reconstruction errors. *Top Right:* Final reconstruction error (left) and computational time for single iteration. *Bottom:* Numerical complexity and memory consumption of the NMF methods.

Conclusions

We introduced a novel algorithm for Non-negative Matrix Factorization combining the fastest existing algorithm with a random compression step. The NMF-methods-comparison presented in this poster demonstrates some appealing properties of our algorithm: a **reduced memory consumption** and **numerical complexity**, which are directly related to the compression factor $(r + r_{ov})$. Moreover, the data compression step does not affect the accuracy of the data approximation **matching the reconstruction precision of the best NMF algorithms**. Finally, even including the calculation of the projection matrices L and R, the FastHALS-RP algorithm finds a solution of the NMF problem ~ 2 times faster than the fastest currently existing algorithm.

Acknowledgements

Our work was supported by a SNF Synergia Grant (*EUCLID : high-precision cosmology in the dark sector*).