Non-Negative Matrix Factorization

Foundations of Data Science Report(E 0229)

Course Project

A Ranga Shaarad (Roll No.: 16056)
Rohit Kumar (Roll No.: 16100)

Under the kind guidance of
Siddharth Barman
Assistant Professor, Department of Computer Science and Automation

Department of Computer Science and Automation (CSA)
Indian Institute of Science(IISc)
December, 2019

Abstract

Linear dimensionality reduction techniques such as principal component analysis and singular value decomposition are powerful tools for dealing with high dimensional data. In this report, we will explore a linear dimensionality reduction technique namely Non negative matrix factorization, a low rank approximation problem which is quite useful while dealing with data in which all entries are non negative, for eg., spectrogram matrix entries or pixels in an image. More precisely, we seek to approximate a given non negative matrix as a product of two low-rank non negative matrices. In this report we will embark on the journey to explore the theoretical complexity associated with this problem, then how to find the non negative factors of our main protagonist and what all applications are there we can use this non negative matrix factorisation.

Linear Dimensionality Reduction, Principal Component Analysis, Singular Value Decomposition, Non-Negative matrix Factorisation, Applications, Algorithms

1 Introduction

Non Negative Matrix Factorization(NMF) is a useful **Linear Dimensionality Reduction Technique**(LDR) for non-negative data, and is widely used tool for compression, visualisation, feature selection and noise filtering. So the idea of NMF is to decompose a given non-negative matrix **X** into factors **W** and **H** which are elementwise nonnegative, i.e

$$X \approx WH \tag{1}$$

The problem can be interpreted as follows.

- Each column of the matrix $X \in \mathbb{R}^{p \times n}$ is a data point, that is $X(:,j) = x_j$ for $1 \leq j \leq n$.
- Each column of the matrix W $\in \mathbb{R}^{p \times r}$ is a basis element, that is $W(:,k) = w_k$ for $1 \leq k \leq r$,and
- Each column of the matrix $H \in \mathbb{R}^{r \times n}$ gives the coordinates of data point $X(:,j)$ in the basis W, that is, $H(:,j) = h_j$ and $x_j = Wh_j$ for $1 \leq j \leq n$.

NMF was introduced in 1994 by Paatero and Tapper [7] and started to be extensively studied after the publication of an article by Lee and Seung [6] in 1999. Following that, it has been used extensively in various machine learning applications like music analysis, graph clustering, food quality and safety analysis.

An important feature of NMF is that its nonnegativity constraints typically induce *sparse vectors*. More formally, the reason for this behavior is that stationary point (U, V) of NMF will typically be located at the boundary of the feasible domain $\mathbb{R}^{p \times r} \times \mathbb{R}^{r \times n}$, and hence will feature zero components. Sparsity of the factors is an important consideration in practice: in addition to **reducing memory requirements** to store the basis elements and their weights, sparsity improves interpretation of the factors, especially when dealing with classification/clustering problems e.g., in text mining and computational biology. By contrast, unconstrained low rank approximations such as PCA do not naturally generate sparse factors.

But these advantages of NMF over PCA come at a certain price. First, because of the additional non-negativity constraints, the approximation error of the input data for a given factorization rank r will always be higher for NMF than in unconstrained case. Second, the optimization problem for NMF is more difficult to solve than its unconstrained counterpart: while PCA problems can be solved in polynomial time, NMF problems belong to the class of NP-hard problems.

The rest of the report is organized as follows. Sec.2 will mainly discuss posing the NMF as an optimization problem, the corresponding cost function involved and the problem in solving that function. Than in Sec.3, we will discuss algorithms to compute the desired factors. We mainly discuss two widely used algorithm for the calculation of the factors. In Sec.4, we will discuss the two major applications of NMF, one from speech domain and one from image processing domain.

2 Problem Statement

2.1 Cost Function

The key aspect of any LDR technique is the choice of the measure to assess the quality of the approximation and it should be chosen depending on the *noise model*. In this report we mainly talk about the following optimization problem:

$$\min_{W \in \mathbb{R}^{p\times r}, H \in \mathbb{R}^{r\times n}} \|\mathbf{X} - \mathbf{WH}\|_F^2 \ni \mathbf{W} \geq 0, \mathbf{H} \geq 0 \tag{2}$$

assuming that the noise present in the data is a Gaussian noise. As mentioned, unlike other LDR techniques, NMF induces sparse vectors, but because of that, there are many issues associated with the NMF. Here, we will mainly talk about 3 major issues.

- **NMF is NP-hard** Vavasis in [8], explores the computational complexity of the NMF optimization problem, specifically stating that exact NMF is NP-hard. In order to prove that, Vavasis introduces the following problem, called the Intermediate Simplex (IS) problem.
  - Given a bounded polyhedron

    $$P = \{x \in \mathbb{R}^{r-1} \mid 0 \leq f(x) = Cx + d\} \tag{3}$$

    where $C \in \mathbb{R}^{n\times(r-1)}$, $d \in \mathbb{R}^n$, $[C,\ d] \in \mathbb{R}^{n\times r}$ has rank r, and given a set $S \subset P$ of p points in P not contained in any hyperplane, determine r points in P whose convex hull $\mathcal{T}$ contains S, i.e a polytope $\mathcal{T}$ with r vertices such that $S \subseteq T \subseteq P$, or determine that such a solution does not exist, if that is the case.

  P is refered to as the outer simplex, $conv(S)$ as the inner simplex, and $\mathcal{T}$ as the intermediate simplex.
  It has been proved that *there exists a polynomial time reduction from exact NMF to IS and vice versa*, and therefore IS is NP-hard.

- **NMF is ill-posed** NMF factorisation does not produce a unique solution, i.e, given an NMF (W, H) of X, there usually exist equivalent NMF's (W', H') with $W'H'^{=WH}$. This can easily be seen to occur by using any monomial matrix Q and letting $W' = WQ$ and $H' = Q^{-1}H$. However, the problem occurs if this happens for a non monomial matrix Q, since that changes the problem specific interpretation of W as a basis. For example[4], when NMF being applied in text mining, this would lead to different topics and classifications [3].

- **Choice of r** The choice of factorization rank r is the most important and tricky in NMF. Some approaches are trial and error, estimation using the SVD and the use of expert insights.

3 Algorithms

3.1 Alternating Nonnegative Least Square (ANLS)

Although NMF is a non convex and difficult problem, it is convex separately in each of the two factors W and H. More precisely, this convex problem corresponds to a non negative least square(NNLS) problem, i.e, a least squares problem with nonnegativity constraints. This problem can be decomposed into p independent NNLS in r variables

$$||X - WH||_F^2 = \sum_{i=1}^{p} ||X_{i:} - W_{i:}H||^2 \tag{4}$$

which make this problem easier to solve. The so-called alternating least square(ANLS) algorithm for NMF minimizes the cost function alternatively over factors W and H so that a stationary point of NMF is obtained in the limit.

Algorithm 1 Alternating Least Square

```
Input: Data matrix X $\in \mathbb{R}^{p\times n}$
Initialization: Generate some initial matrices $W^{(0)} \geq 0$ and $H^{(0)} \geq 0$
while Stopping Criteria not satisfied do
    $W \longleftarrow (\text{argmin}_{W\geq 0}(||X - WH||_F)$        (5)
    $H \longleftarrow (\text{argmin}_{H\geq 0}(||X - WH||_F)$        (6)
end
```

3.1.1 Convergence

ANLS is guaranteed to converge to a stationary point. Since each iteration of ANLS computes an optimal solution of NNLS sub problem, each iteration of ANLS decreases the error the most among NMF algorithms. However, each iteration is computationally expensive and difficult to implement, since the problem solved in each iteration is a constrained problem.

3.2 Multiplicative updates

In the previous algorithm, each step consisted of solving a constrained minimization problem. For this reason, each step in ANLS is computationally expensive. Also ALS is not guaranteed to converge. To overcome these disadvantages, [6] proposed iterative algorithms involving multiplicative updates at each step. Each step is simple and is guaranteed to reduce the objective function. Different multiplicative updates have been proposed for different objective functions.

For the problem of minimizing $||X - WH||_F^2$, the squared Frobenius norm, the following multiplicative updates can be used.

$$H_{\alpha\mu} \leftarrow H_{\alpha\mu} \frac{(W^T X)_{\alpha\mu}}{(W^T W H)_{\alpha\mu}},$$

$$W_{ia} \leftarrow W_{ia} \frac{(X H^T)_{ia}}{(W H H^T)_{ia}}.$$

For the problem of minimizing $D(X||WH)$, where

$$D(A||B) = \sum_{ij} \left(A_{ij} \log \frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij} \right),$$

the multiplicative updates considered are

$$H_{\alpha\mu} \leftarrow H_{\alpha\mu} \frac{\sum_i W_{ia} X_{i\mu}/(WH)_{i\mu}}{\sum_k W_{ka}}$$

$$W_{ia} \leftarrow W_{ia} \frac{\sum_\mu H_{a\mu} X_{i\mu}/(WH)_{i\mu}}{\sum_\nu H_{a\nu}}$$

The measure $D(A||B)$ is analogous to the Kullback-Leibler divergence between probability distributions and reduces to it when A and B considered are probability distributions, ie., when $\sum_{ij} A_{ij} = \sum_{ij} B_{ij} = 1$. In further discussions of convergence etc., we consider the Frobenius norm as the objective function.

It is clear from the update equations that at the optimum, if $X = WH$, then the updates don't change H or W. So, the optimal solution is a fixed point of the updates.

These updates can be shown to reduce the objective function at every step. Since the objective is bounded below, the algorithm is guaranteed to converge.

3.2.1 Intuition

Consider the problem of minimizing $\frac{1}{2}||X - WH||_F^2$ w.r.t H iteratively using gradient descent updates. The gradient of the objective w.r.t H is given by $W^T W H - W^T X$. This can be seen by expanding the objective function as the sum $\frac{1}{2}\sum_j ||X_j - WH_j||_2^2$, where X_j and H_j are the j'th columns of X and H respectively. So the update becomes

$$H_{\alpha\mu} \leftarrow H_{\alpha\mu} + \eta_{\alpha\mu}\left[(W^TX)_{\alpha\mu} - (W^TWH)_{\alpha\mu}\right].$$

If all $\eta_{\alpha\mu}$ are chosen to be equal, then this becomes the standard gradient descent update. Furthermore, if the chosen value is sufficiently small at each step, then the objective function will decrease. However, the optimal choice of $\eta_{\alpha\mu}$ is not clear. It can be seen that choosing

$$\eta_{\alpha\mu} = \frac{H_{\alpha\mu}}{(W^TWH)_{\alpha\mu}}$$

gives the multiplicative updates proposed for minimizing $||X - WH||_F^2$. However, the $\eta_{\alpha\mu}$ are not chosen equal here, but diagonally rescaled and are also not guaranteed to be small. Hence, it is not directly clear if this choice will result in a decrease in the objective function.

3.2.2 Convergence

The convergence analysis of this algorithm actually gives the reason why the updates were chosen in this particular way.

The basic idea is to construct an 'auxiliary' function, which upper bounds the objective function, and to minimize the auxiliary function. The auxiliary function is chosen in such a way that this step is simple. This will result in a decrease in the objective function as well by a simple update which does not require minimizing the objective directly. The idea is formalised as follows.

Definition 3.1. *$G(h, h')$ is an auxiliary function for $F(h)$ if $\forall h, h'$, $G(h, h') \geq F(h)$ and $G(h, h) = F(h)$*

Lemma 3.0.1. *If G is an auxiliary function for F, and h is updated as*

$$h_{t+1} = \arg\min_h G(h, h_t),$$

then $F(h_{t+1}) \leq F(h_t)$.

Proof.

$$\begin{aligned} F(h_{t+1}) &\leq G(h_{t+1}, h_t) && \text{(by the definition of } G\text{)} \\ &\leq G(h_t, h_t) && \text{(by the definition of } h_{t+1}\text{)} \\ &= F(h_t) && \text{(by the definition of } G\text{)} \end{aligned}$$

□

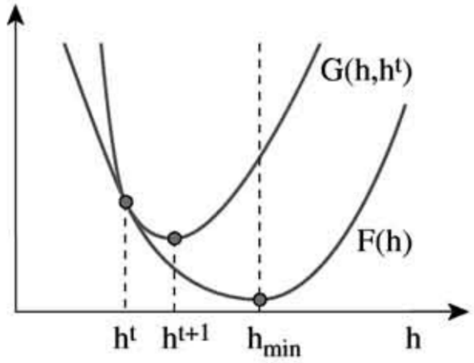


Figure 1: Auxiliary function [6]

Now, we need to define an appropriate auxiliary function for the current objective function. It can be shown [6] that for the function $F(h) = \frac{1}{2}||x - Wh||_2^2$, the following function is an auxiliary function.

$$G(h, h') = F(h, h') + (h - h')^T \nabla F(h') + \frac{1}{2}(h - h')^T K(h')(h - h')$$

where $K(h')$ is the diagonal matrix given by $K_{ab}(h') = \delta_{ab}(W^T W h')_a / h'_a$.

This function is a convex quadratic function in h and hence can be minimized directly as $h - h' = -K(h')\nabla F(h')$. Substituting $h' = h_t$ and $h = h_{t+1}$ and the expressions for K and the gradient gives the multiplicative updates expression.

3.3 Hierarchical alternating Least Squares

All the previous algorithms have split the variables into two blocks corresponding to W and H and update each block separately at each step. The blocks can further be divided and the updates could be done column wise [1]. Considering the problem only in terms of W (the update of H is analogous), we can rewrite the objective function as

$$\begin{aligned} ||X - WH||_F^2 &= ||X - \sum_k W(:,k)H(k,:)||_F^2 \\ &= ||X - \sum_{k \neq l} W(:,k)H(k,:) - W(:,l)H(l,:)|| \end{aligned}$$

This problem can be solved exactly in terms of $W(:,l)H(l,:)$ and the update becomes

$$\begin{aligned} W(:,l) \leftarrow & \operatorname*{arg\,min}_{W(:,l) \geq 0} ||X - \sum_{k \neq l} W(:,k)H(k,:) - W(:,l)H(l,:)||_F \\ &= \max\left(0, \frac{XH(l,:)^T - \sum_{k \neq l} W(:,k)H(k,:)H(l,:)^T}{||H(l,:)||_2^2}\right) \end{aligned}$$

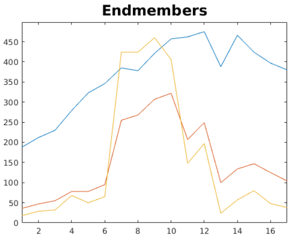
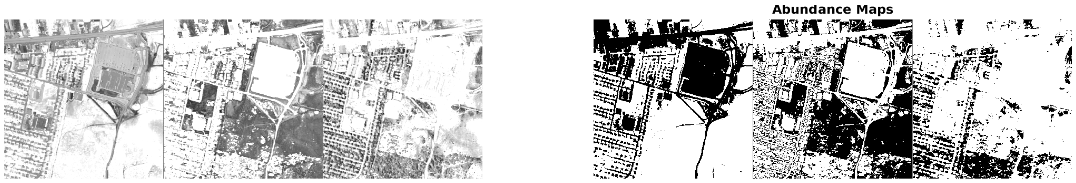


Figure 2: Plot showing variation of endmembers v/s wavelength,where 3 endmembers are taken for analysis.



(a) Clusters or endmembers(3 endmembers)

(b) Abundance Maps(for 3 endmembers)

Figure 3: Hyperspectral image output

Clearly, these updates are computationally relatively inexpensive, just like multiplicative updates. Furthermore, it is seen to converge better in practice.

4 Application

4.1 Hyperspectral Data Analysis

A hyperspectral image is a set of images of the same object or scene taken at different wavelengths. Each image is acquired by measuring the reflectance of each individual pixel at a given wavelength. The main aspect of hyperspectral image analysis is the identification of materials present in the scene being imaged. The model that we will assume is a *linear mixing model* i.e *the spectral signature of each pixel is nothing but the linear combination of the spectral signature of its constituent elements (endmembers).*

The matrix X is constructed as follows: each 2D image corresponding to a wavelength is vectorized and is a row $X_{i:}$,while each column $X_{:j}$ corresponds to the spectral signature of the corresponding pixel. So the problem can simply be stated as:

$$X_{:j} \approx WH_{:j} \quad \forall j, \tag{7}$$

ie, each pixel is to be obtained as a linear combination of columns of W, each of which correspond to an endmember.

Fig. 2 shows the results from using NMF to cluster[5] pixels into three regions corresponding to three different endmembers. The code used to create this was obtained from *https://sites.google.com/site/nicolasgillis/code.*

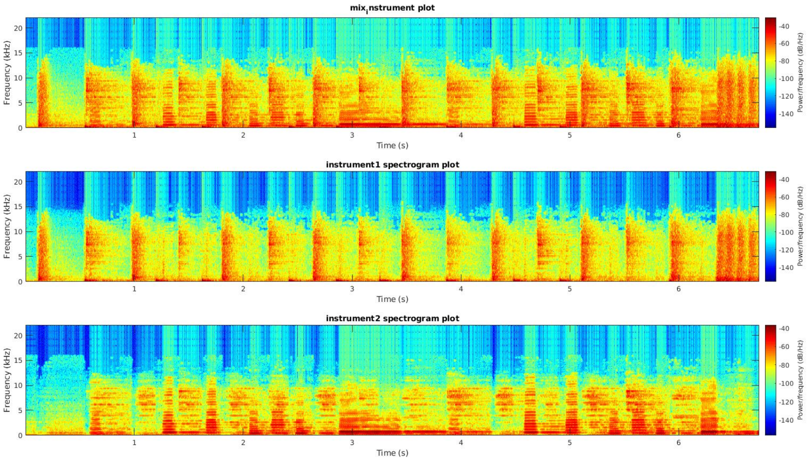


Figure 4: (a) Spectrogram representing an excerpt of the 1964 recording of 'I Got You' by James Brown and The Famous Flames, which is a mixture of a percussive and a melodic instrument. We can see the high concentration of energy in some frames and also a constant energy distributed throughout. (b) Spectrogram of the percussive instrument (drums). (c) Spectrogram of the melodic instrument (Anglo saxophone).

4.1.1 High dimensional images

Using these standard methods directly on large data such as high dimensional images of the Earth's surface (such as in *https://aviris.jpl.nasa.gov/data/get_aviris_data.html*) may pose problems with regards to memory space requirements etc. One heuristic for working past this is to divide the matrix into sufficiently small blocks and iteratively solve the problem on each of the blocks, as in *https://github.com/Gururajk/Block_NMF*.

4.2 Harmonic Percussive Source Separation

The general goal of music source separation is to decompose a recording into its constituent signal components, for example a percussive component and a melodic component. [2] uses a two-stage approach, unifying local and global methods. In the first stage, Kernel Additive Methods(KAM) were used to find initial separation and estimates of the percussive and harmonic parts. In the second stage, these parts are combined and further refined using NMF.

4.2.1 Implementation Details

We take the single channel audio which is the mixture of percussive and melodic instruments, and then apply Short Time Fourier Transform(STFT) with block size of 2048 samples and a hop size of 512 samples(75% overlap). Further computations are applied on the **absolute** magnitude of the STFT output. Then, KAM filtering is applied, which is a local estimate, in the sense that, based on each time-frequency bin, this algorithm will classify whether that bin belongs to the percussive or the melodic portion. These estimates are used as an input to an NMF algorithm, which further refines them to obtain a better separation between the two components.

5 Bibliography

References

[1] Andrzej CICHOCKI and Anh-Huy PHAN. "Fast Local Algorithms for Large Scale Nonnegative Matrix and Tensor Factorizations". In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E92.A.3 (2009), pp. 708–721. DOI: `10.1587/transfun.E92.A.708`.

[2] Christian Dittmar, Patricio López-Serrano, and Meinard Müller. "Unifying local and global methods for harmonic-percussive source separation". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pp. 176–180.

[3] Nicolas Gillis. "Sparse and unique nonnegative matrix factorization through data preprocessing". In: *Journal of Machine Learning Research* 13.Nov (2012), pp. 3349–3386.

[4] Nicolas Gillis. "The why and how of nonnegative matrix factorization". In: *Regularization, Optimization, Kernels, and Support Vector Machines* 12.257 (2014), pp. 257–291.

[5] Nicolas Gillis, Da Kuang, and Haesun Park. "Hierarchical clustering of hyperspectral images using rank-two nonnegative matrix factorization". In: *IEEE Transactions on Geoscience and Remote Sensing* 53.4 (2014), pp. 2066–2078.

[6] Daniel D. Lee and H. Sebastian Seung. "Algorithms for Non-negative Matrix Factorization". In: *Proceedings of the 13th International Conference on Neural Information Processing Systems*. NIPS'00. Denver, CO: MIT Press, 2000, pp. 535–541. URL: `http://dl.acm.org/citation.cfm?id=3008751.3008829`.

[7] Pentti Paatero and Unto Tapper. "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values". In: *Environmetrics* 5.2 (1994), pp. 111–126. DOI: `10.1002/env.3170050203`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/env.3170050203`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1002/env.3170050203`.

[8] Stephen A Vavasis. "On the complexity of nonnegative matrix factorization". In: *SIAM Journal on Optimization* 20.3 (2009), pp. 1364–1377.