Application of NMF to topic modeling and facial feature extraction

Nicolas Gillis

1 Introduction

The goal of this session is to implement a few simple algorithms for NMF, and then use them for two applications, namely topic modeling and facial feature extraction.

We will consider the following NMF optimization model

$$\min_{U \geq 0, V \geq 0} \|M - UV\|_F^2, \tag{1}$$

where $M \in \mathbb{R}^{m \times n}_+$ is the input matrix, $U \in \mathbb{R}^{m \times r}_+$ and $V \in \mathbb{R}^{r \times n}_+$ are the variables with r being the given factorization rank, and $\|.\|_F^2$ is the squared Frobenius norm defined as $\|M - UV\|_F^2 = \sum_{i,j} (M - UV)_{i,j}^2$. It is the maximum likelihood estimator in the presence of i.i.d. Gaussian noise.

2 Algorithm I: Multiplicative Updates

Most NMF algorithm rely on alternating optimization; see Algorithm 1. The reason is that the subproblem in U and V separately are convex nonnegative least squares problems.

Algorithm 1 Alternated framework for NMF

Require: $M \in \mathbb{R}^{m \times n}$, $U^{(0)} \in \mathbb{R}^{m \times r}_+$, $V^{(0)} \in \mathbb{R}^{r \times n}_+$

1: **for** $i = 1, 2, ...$ **do**

2: $\quad U^{(i)} \leftarrow$ optimize U in (1) for $V = V^{(i-1)}$ fixed, that is, solve exactly or approximately

$$\min_{U \geq 0} \|M - UV^{(i-1)}\|_F^2.$$

3: $\quad V^{(i)} \leftarrow$ optimize V in (1) for $U = U^{(i)}$ fixed, that is, solve exactly or approximately

$$\min_{V \geq 0} \|M - U^{(i)}V\|_F^2.$$

4: **end for**

In the original paper introducing NMF (Lee and Seung, "Learning the parts of objects by non-negative matrix factorization", Nature, 1999), the authors propose the multiplicative updates (MU) that are guaranteed to decrease the objective function; see Algorithm 2. The notation $A \circ B$ means the component-wise product between matrices A and B of the same size, and $\frac{[A]}{[B]}$ the component-wise division, and $A^\top$ denotes the transpose of A.

Algorithm 2 Multiplicative Updates (MU)

Require: $M \in \mathbb{R}^{m \times n}$, $U^{(0)} \in \mathbb{R}^{m \times r}$, $V^{(0)} \in \mathbb{R}^{r \times n}$

1: **for** $i = 1, 2, ...$ **do**

2: $\quad U^{(i)} \leftarrow U^{(i-1)} \circ \frac{\left[M {V^{(i-1)}}^\top\right]}{\left[U^{(i-1)} V^{(i-1)} {V^{(i-1)}}^\top\right]}$

3: $\quad V^{(i)} \leftarrow V^{(i-1)} \circ \frac{\left[{U^{(i)}}^\top M\right]}{\left[{U^{(i)}}^\top U^{(i)} V^{(i-1)}\right]}$

4: **end for**

An intuition behind the MU is based on gradient descent, we have

$$\nabla_V := \nabla_V \|M - UV\|_F^2 = 2U^\top (UV - M),$$

while the KKT optimality conditions of (1) are given by

$$V \geq 0, \quad \nabla_V \geq 0 \quad \text{and} \quad V \circ \nabla_V = 0.$$

Hence if $\nabla_V > 0$, that is, $(U^\top UV)_{kj} > (U^\top M)_{kj}$, V_{kj} should be decreased, and vice versa.

Task 1. Implement the MU for NMF. Try it on a simple problem by generating `M = rand(m,r)*rand(r,n)`, and display the evolution of the relative error, $\frac{\|M-UV\|_F}{\|M\|_F}$.

Questions: How do you initialize U and V? What stopping criterion will you use?

3 Applications

The two data sets are available from `https://gitlab.com/ngillis/nmfbook/`, along with some useful functions.

3.1 Facial feature extraction: CBCL

Task 2. Load the CBCL data set, and compute an NMF of rank 49 using the MU. Then display the facial features extracted using the function 'affichage'.

3.2 Topic modeling: TDT2

Task 3. Load the TDT2 data set, and compute an NMF of rank 20 using the MU. Then interpret the result by identifying the 10 most important words in each column of U.

3.3 Sensitivity to initialization

Repeat task 2 and 3 for different initial factors $U^{(0)}$ and $V^{(0)}$. Do you get the same results? What do you observe?

4 Faster algorithm: HALS

An algorithm that typically converges much faster than the MU is the hierarchical alternating least squares (HALS); see Algorithm 3. HALS was introduced by Cichocki, Zdunek and Amari ("Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization", In International Conference on Independent Component Analysis and Signal Separation, pp. 169-176, 2007).

Algorithm 3 Hierarchical Alternating Least Squares (HALS)

Require: $M \in \mathbb{R}^{m \times n}$, $U \in \mathbb{R}^{m \times r}$, $V \in \mathbb{R}^{r \times n}$

for $i = 1, 2, ...$ **do**

  Calcul de $MV^\top$ et $VV^\top$

  for $k = 1 \to r$ **do**

    $U_{:k} \leftarrow \max\left(0, \frac{(MV^\top)_{:k} - \sum_{l=1, l\neq k} U_{:l}(VV^\top)_{lk}}{(VV^\top)_{kk}}\right)$

  end for

  Calcul de $U^\top M$, $U^\top U$

  for $k = 1 \to r$ **do**

    $V_{k:} \leftarrow \max\left(0, \frac{(U^\top M)_{k:} - \sum_{l=1, l\neq k} (U^\top U)_{kl} V_{l:}}{(U^\top U)_{kk}}\right)$

  end for

end for

The idea behind HALS is to update each column of U (resp. row of V) sequentially.

Task 4. Show that the formula used in HALS is the optimal solution of the NMF problem when optimizing only the kth column of U (that is, $U_{:k}$) in NMF (1), that is, show that the update in HALS corresponds to

$$U_{:k} \quad \leftarrow \quad \operatorname{argmin}_{U_{:k} \geq 0} \|M - UV\|_F^2.$$

Task 5. Implement HALS, and compare it with the MU on the two data sets. How much faster is HALS?

References

Cichocki, A., Zdunek, R., Phan, A. H., & Amari, S. I. (2009). Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation. John Wiley & Sons.

Gillis, N. (2020). Nonnegative Matrix Factorization. Society for Industrial and Applied Mathematics.