Nonnegative matrix factorization with linear programming

Fadhel Ayed
fadhel.ayed@gmail.com

Xiayang Zhou
xiayang.zhou@polytechnique.edu

Abstract

In this paper we present the main results and algorithms described in [1] to provably compute a nonnegative matrix factorization (NMF) under weak assumptions using linear programming.
Keywords : Nonnegative Matrix Factorization, linear algebra, machine learning,...

1 Introduction

Nonnegative matrix factorization (NMF) is a common method to represent a data matrix while making explicit its latent structures. More precisely, given a data matrix of non negative entries, we aim to find F and W nonnegative as well, such that $X \approx FW$. The intuition is to find r rows (defining W) which convex cone contains the rows of X. We want r to be as small as possible. This method is therefore particularly used in machine learning for feature selection for instance.

However, in the general case, finding such an optimal factorization is a NP problem. Therefore, in order to have a tractable problem, one needs to make some further assumptions. In this paper we will present the work of ***Recht et al*** from [1]. We will therefore make an assumption of separability and give its geometrical interpretation. Under that assumption we will show how we can derive a linear program to solve the problem. Finally in the last section we will show how we can implement an algorithm to find a solution efficiently.

2 Separable Nonnegative Matrix Factorization

2.1 Nonnegative Matrix Factorization and geometric interpretation

Let $X \in \mathcal{M}_{f,n}(\mathbb{R})$ matrix to factorize, we assume that its rows are normalized (they sum to 1). The rows index the features and the columns the examples. We note $X_{i,.}$ the i^{th} row of X (also noted X_i when there are no ambiguities) and $X_{.,j}$ its j^{th} column.

Definition 2.1 *(rank-r NMF factorization) $X = FW$ is called a rank-r nonnegative factorization (NMF) of X if $F \in \mathcal{M}_{f,r}(\mathbb{R}_+)$ and $W \in \mathcal{M}_{r,n}(\mathbb{R}_+)$. F is then called feature matrix and W weight matrix.*

However, finding such an r turns out to be a NP problem. Therefore, from now on we suppose that we know r. Since we assumed that the rows of X are normalized, we have the following Lemma 2.1 that will allow us to give a geometrical interpretation of the NMF factorization

Lemma 2.1 *If X has normalized rows and has a rank-r nonnegative factorization, then we can find F and W, feature and weight matrices, which rows are also normalized.*

Proof The proof is quite straight forward. Let $X = FW$ a rank-r factorization. Let $D = diag(d_1, ..., d_r)$ where $d_i = \frac{1}{\sum_j W_{i,j}}$. Then $W' = DW$ has normalized rows. Furthermore, if

we call $F' = FD^{-1}$, $X = F'W'$ is a rank-r nonnegative factorization of X. Besides, since X is normalized, we have:

$$\forall i \in \{1, .., f\},\ 1 = \sum_j X_{i,j} = \sum_j \sum_k F'_{i,k} W'_{k,j} = \sum_k F'_{i,k}\ \ (since\ W'\ is\ normalized)\ \blacksquare$$

Geometric interpretation We deduce from Lemma 2.1 that we can, without loss of generality suppose that F and W are normalized. Therefore, the problem turns out to finding r elements (the rows of W) of $\mathbb{R}^n_+$, in the $l_1 - unit\ sphere$, called $hott$, which convex hull contains all the rows of X. More precisely, for each row index i, X_i is the convex combination of the rows of W defined by the coefficients $F_{i,.}$

In the following, we will always assume that F and W are normalized. Since the objective is to find a factorization with small $rank - r$, we need to make sure that none of the rows of W is redundant (already in the convex hull of the other rows).

Definition 2.2 *(simplicial set and matrix) A set of vectors $\{x_1, ..., x_r\}$ is simplicial if no vector x_i is in the convex hull of the others. We say that a matrix is simplicial if its rows are simplicial.*

Therefore, we are interested in a factorization where W is $simplicial$. Like said previously, finding W in the general case cannot be done efficiently, therefore we need further assumptions. A common assumption is to suppose that we can find a $seperable\ NMF$ of X (see below)

Definition 2.3 *(separable NMF) A NMF $X = FW$ is separable if W is simplicial and*

$$\forall i \in \{1, ...r\}, \exists i' \in \{1, ...f\}\ such\ that\ W_{i,.} = X_{i',.}$$

Or, equivalently, since W is separable, if there exists a permutation matrix Π such that

$$\Pi F = \left[\frac{I_r}{M}\right]$$

2.2 Separable and near separable matrix factorization

We will present here the results described in [2]: under separable or near separable factorization assumption, we will show how we can find F and W in a polynomial time .
Let us first discuss the case where X admits a separable matrix factorization.

Definition 2.4 *(Loner row) A row of a matrix is called a $loner$ if it doesn't lie in the convex hull of the remaining rows (ignoring copies)*

Theorem 2.1 *If X admits a separable factorization $X = FW$, then the rows of W are exactly the ones of X that are loners*

Proof Suppose that $X_{i,.}$ is a loner for some i and $X_{i,.}$ is not a row of W. Since the NMF is separable, the rows of W are among the remaining rows of X. Besides, by definition of a loner, $X_{i,.}$ is not in the convex hull of the remaining rows of X, and therefore is not in the convex hull of the rows of W, which is absurd.
Suppose now that $X_{i,.} = W_{i',.}$ for some i, i' and $X_{i,.}$ is not a loner. Therefore $X_{i,.}$ can be written as a convex combination of the remaining different rows of X, we note J the set of corresponding indexes. Hence we have

$$W_{i',.} = \sum_{j \in J} a_j X_{j,.} = \sum_{j \in J} \sum_{k=1}^{r} a_j F_{j,k} W_{k,.} = \sum_{k=1}^{r} \alpha_k W_{k,.}$$

Where $\alpha_k = \sum_{j \in J} a_j F_{j,k} \geq 0$ and since F is normalized, $\sum \alpha_k = 1$. Besides, since for all $j \in J$, $X_{j,.} \neq W_{i',.}$, we have $F_{j,i'} < 1$. So $\alpha_{i'} < \sum_{j \in J} a_j = 1$. And finally, we find that

$$W_{i',.} = \sum_{k \neq i'} \frac{\alpha_k}{1 - \alpha_{i'}} W_{k,.}$$

Which is absurd since W is simplicial $\blacksquare$

Remarks:

- From Theorem 2.1, we understand that in the separable case, solving the NMF factorization problem turns out to finding the r rows of X which don't lie in the convex hull of the remaining rows. We therefore only need to compute the distance between a row and the convex hull of the others, which can be expressed as a linear program.
- However, considering only matrices admitting a separable NMF is too restrictive in practice. Therefore, under stronger assumptions, AGKM have developed an algorithm that provably solves the problem in the near separable case, that is when we add some noise.

We introduce the $(\infty, 1) - norm$ to control the amplitude of the noise

Definition 2.5 *($(\infty, 1) - norm$) Let $\Delta \in \mathcal{M}_{f,n}(\mathbb{R})$, we note*

$$||\Delta||_{\infty,1} := \max_{i \leq f} \sum_{j=1}^{n} |\Delta_{i,j}|$$

The algorithm (Algorithm 1), presented in [2], allows us to find in a polynomial time a rank-r NMF of a matrix well approximated by a separable NMF (if the $(\infty, 1) - norm$ of the noise is small), as stated in the Theorem 2.2.

Algorithm 1 AGKM: Approximably Separable nonnegative Matrix Factorization

1. **Input:** X, ϵ, α (see Theorem 2.2
2. Initialize $R = \emptyset$
3. Compute $D \in \mathcal{M}_{f,f}(\mathbb{R}_+)$ such that $D_{i,j} = ||X_{i,.} - X_{j,.}||_1$
4. **for** k=1..f **do**
5. Find the set $\mathcal{N}_k$ of rows at least $5\epsilon/\alpha + 2\epsilon$ away from $X_{k,.}$
6. Compute the distance δ_k of $X_{k,.}$ from $conv(\{X_{j,.},\ j \in \mathcal{N}_k\})$
7. **If** $\delta_k > 2\epsilon$, add k to R
8. **endfor**
9. Cluster the rows of R such that i and j are in the same cluster iff $D_{i,j} \leq 10\epsilon/\alpha + 6\epsilon$
10. Choose an element from each cluster to yield W
11. $F = \underset{Z \in \mathcal{M}_{f,r}(\mathbb{R})}{\arg\min} ||X - ZW||_{\infty,1}$

Definition 2.6 *(α-robust simplicial) We call W α-robust simplicial if no row in W has 1_1 distance smaller than α to the convex hull of the remaining rows in W. (Here all rows have unit l_1-norm.)*

Theorem 2.2 *Suppose ϵ and α nonnegative constants such that $\epsilon \leq \frac{\alpha^2}{20+13\alpha}$, and $\hat{X} = X + \Delta$, where X admits a separable factorization FW, where W is α-robust simplicial and $||\Delta||_{\infty,1} \leq \epsilon$. Then Algorithm 1 finds a rank-r factorization $\tilde{F}\tilde{W}$ such that $||\hat{X} - \tilde{F}\tilde{W}||_{\infty,1} \leq 10\epsilon/\alpha + 7\epsilon$*

Remarks:

1. The proof of the Theorem 2.2 is close to the one of Theorem 2.1, we won't give the details here (it can be found in [2]). We can however notice that the separability condition isn't sufficient in this case. Indeed, we need that each row of W is far enough from the convex hull of the others (α-robust simplicial condition).
2. The distance δ_k in step 6 can be done by solving a linear program
3. At first sight, the clusters in step 9 seems to be ill-defined in the general case. However, it is proved in [2] that under the assumptions of Theorem 2.2, the rows of R are gathered within small l_1-balls that don't intersect. Therefore step 9 can be done in $O(rf)$
4. Step 9 consists in computing the projection of each row of X on the convex cone defined by the hott features (rows of W). Hence it can be done by solving a linear program.

5. However, this algorithm has numerous undesirable features. It needs the knowledge of both α and ϵ. It has a computational complexity of $O(nf^2)$ (step 3 for instance). Finally, the algorithm is not as robust to noise an one could need.

3 NMF by Linear Programming

In order to overcome the issues described in the last section, the authors of [1] thought of using Linear programming to solve the problem. The idea came from the following observation: in the separable case, we have, using the previous notations that

$$X = \Pi^T \begin{bmatrix} I_r \\ M \end{bmatrix} W = \Pi^T \begin{bmatrix} I_r & 0 \\ M & 0 \end{bmatrix} \Pi X$$

Let $C = \Pi^T \begin{bmatrix} I_r & 0 \\ M & 0 \end{bmatrix} \Pi$, called $factorization\ localization$. It is straightforward to notice that if we know C, finding the simplicial rows of X forming W comes down to finding the indexes i such that $C_{i,i} = 1$. The idea of this section is to try to find a characterization of C using Linear programming. We notice first that C is a element of the following set:

$$\Phi(X) = \{C \geq 0 : CX = X,\ Tr(C) = r,\ diag(C) \leq 1, \forall i, j\ C_{i,j} \leq C_{j,j}\}$$

In order to take into consideration the noise, we relax the condition $CX = X$, and consider the following set instead:

$$\forall \tau,\ \Phi_\tau(X) = \{C \geq 0 : ||CX - X||_{\infty,1},\ Tr(C) = r,\ diag(C) \leq 1, \forall i, j\ C_{i,j} \leq C_{j,j}\}$$

3.1 Primal and dual problems

Let $t \in \mathbb{R}^f$, we will be using the following linear program in order to prove the results.

$$\begin{aligned} \underset{C}{\text{minimize}} \quad & t^T diag(C) \\ \text{subject to} \quad & C \in \Phi_\tau(X) \end{aligned} \tag{1}$$

After forming the Lagrangian (we skip the computational details here), we find that the dual problem of (1) is given by

$$\begin{aligned} \underset{R,M,\gamma,\eta}{\text{maximize}} \quad & \text{Tr}(RX^T) - \tau||R||^*_{\infty,1} - \sum_{i=1}^{f} \gamma_i + r\eta \\ \text{subject to} \quad & RX^T - diag(\gamma) + diag(1^T M) - M + \eta I \leq diag(t) \\ & \gamma \geq 0,\ M \geq 0 \end{aligned} \tag{2}$$

3.2 Main Theoretical results

We now have all the ingredients to describe the approach to compute the NMF using linear programming. Let us first introduce the Algorithm 2

Algorithm 2 Approximably Separable Nonnegative Matrix Factorization by Linear Programming

Input: An $f \times n$ nonnegative matrix X, ϵ.

1. Choose any vector p with distinct entries
2. Compute C solution of (1) with $t = p$ and $\tau = 2\epsilon$
3. Let I the set of indexes of the r largest entries of $diag(C)$ and set $W = X_{I,.}$
4. $F = \underset{Z \in \mathcal{M}_{f,r}(\mathbb{R})}{\arg\min}\ ||X - ZW||_{\infty,1}$

Proposition 3.1 allows us to give theoretical guaranties about the quality of the approximation.

Theorem 3.1 *Suppose $X = FW$ a rank-r NMF with W α-robust simplicial and $F = \begin{bmatrix} I_r \\ M \end{bmatrix}$ such that $||M||_{\infty,1} \leq \beta$. Suppose $\epsilon < \frac{1}{8}\alpha(1-\beta)$. Suppose $\hat{X} = X + \Delta$ with $||\Delta||_{\infty,1} \leq \epsilon$. Then Algorithm 2 constructs a NMF satisfying $||\hat{X} - \hat{F}\hat{W}||_{\infty,1} \leq 2\epsilon$*

Remarks: Before proving the proposition, let us first make a few remarks

1. The assumption $||M||_{\infty,1} \leq \beta$ is always true with $\beta = 1$. In that case, $\epsilon = 0$. Therefore we are in the case where X is separable. We see that in that case, the previous theorem assures us that the algorithm will find an exact NMF
2. The interesting situation is therefore when $\beta < 1$. The condition on M is then just another way of saying that there are no copies of the hott topics (a row of W cannot appear twice in X). We can however skip that assumptions using Theorem 3.2. Its proof is similar to the one of Theorem 3.1, with technicalities due to copies. Here, we won't do the proof of Theorem 3.2
3. We notice that Algorithm 2 is more robust to the noise that Algorithm $AGKM$ (see also Theorem 3.2). We also notice that here, the algorithm only needs ϵ to run, without knowing α
4. Finally, expressing the problem as a Linear Program enables scaling to very large data sets.

To prove theorem 3.1, we will need this technical lemma:

Lemma 3.1 *Let $\{X_1, ..., X_r\}$ simplicial vectors of $\mathbb{R}^r_+$, such that $\forall k, ||X_k||_1 = 1$. Let $i \in \{1, .., r\}$. Then:*

$$\forall t > 0, \exists \rho \in \mathbb{R}^r \text{such that } \rho.X_i = 1 \text{ and } \forall j \neq i,\ \rho.X_j < -t$$

Proof: Let $t > 0$. Let us note $u = \frac{1}{||X_i||_2^2} X_i$ and $\mathcal{H} = \{X_i\}^\perp$. The objective is therefore to find $v \in \mathcal{H}$ such that $\forall j \neq i,\ v.X_j < 0$. Afterwards, taking $\rho = u + \lambda v$ for λ large enough ends the proof. We first notice that $\forall k \in \{1, ..., r\},\ X_k \in X_i + \{1\}^\perp$ (where $1 = (1, ..., 1)$). Let us note $\forall k,\ x_k = X_k - X_i \in \{1\}^\perp$. It is straightforward to notice that

$$X_i \notin \text{Conv}(X_j,\ j \neq i) \iff 0 \notin \text{Conv}(x_j,\ j \neq i) = \mathcal{C}$$

If $\mathcal{H} = \{1\}^\perp$, let w the l_2 projection of 0 on $\mathcal{C}$. Then, since $\mathcal{C}$ is convex and $0 \notin \mathcal{C}$, we know that $\forall y \in \mathcal{C},\ w.y > 0$. We then take $v = -w$.
If $\mathcal{H} \neq \{1\}^\perp$. Let Π the projection from $\{1\}^\perp$ to $\mathcal{H}$. We then know that Π is an isomorphism. Let Π^- its inverse (which is also linear). Suppose that $0 \in \text{Conv}(\Pi(x_j),\ j \neq i) = \mathcal{C}'$. Then we can find $(\lambda_j,\ j \neq i)$ such that $\sum_{j \neq i} \lambda_j \Pi(x_j) = 0$ and $\sum_{j \neq i} \lambda_j = 1$. Therefore, since $\Pi^-(0) = 0$, $\sum_{j \neq i} \lambda_j x_j = 0$. Which is absurd. Therefore $0 \notin \mathcal{C}'$. Let now w be the projection of 0 on $\mathcal{C}'$. Then $\forall y \in \mathcal{C}',\ w.y > 0$. Noting $v = -w$, we have that $\forall j \neq i, v.\Pi(x_j)$. Finally, since $v \in \mathcal{H},\ v.x_j = v.\Pi(x_j)$. We conclude the proof by taking $\rho + \lambda v$ with λ large enough. ■

Proof of Theorem 3.1: Let us first show that the problem 1 with $\tau = 2\epsilon$ is feasible. Since X is separable, we know that we can find $C_0 \in \Phi(X)$ (in particular, we have that $||C_0||_\infty \leq 1$). We then have

$$\begin{aligned} ||C_0\hat{X} - \hat{X}||_{\infty,1} &\leq ||C_0\Delta - \Delta||_{\infty,1} \\ &\leq (||C_0||_\infty + 1)||\Delta||_{\infty,1} \\ &\leq 2\epsilon \end{aligned}$$

Therefore C_0 is a feasible point for the Linear Program 1.
Now let us show that if C is feasible for the linear program then $||CX - X||_{\infty,1} \leq 4\epsilon$. We have

$$\begin{aligned} ||CX - X||_{\infty,1} &\leq ||C(X - \hat{X}) + C\hat{X} - \hat{X} + \hat{X} - X||_{\infty,1} \\ &\leq (||C||_\infty + 1)||\hat{X} - X||_{\infty,1} + ||C\hat{X} - \hat{X}||_{\infty,1} \\ &\leq 4\epsilon \end{aligned}$$

Therefore, if $C \in \Phi_{2\epsilon}(\hat{X})$, then $C \in \Phi_{4\epsilon}(X)$. Now let us show that finding a $C \in \Phi_{4\epsilon}(X)$ is sufficient to identify the hott topics of X. Without loss of generality, to simplify the expression, let

us suppose that the hott rows of X are the r first ones. We will now show that if C is feasible for the linear program of the algorithm, then $\forall i \in \{1, ..., r\},\ C_{i,i} > 1/2$ and $\forall j > r,\ C_{j,j} \leq 1/2$. In order to do so, we will need to show that for all $i \in \{1, ..., r\}$ we can find $\rho \in \mathbb{R}^n$ such that $\rho.X_{i,\cdot} = 1,\ \forall j \neq i,\ \rho.X_{j,\cdot} \leq \frac{-\beta}{1-\beta}$ (here $j \leq r$) and $||\rho||_\infty < \frac{1}{8\epsilon}$.

To do so let $i \leq r$, and let us consider the optimization program on the affine hyperplane $\mathcal{H} = \{\rho : \rho.X_i = 1\}$

$$\text{minimize } ||\rho||_\infty \text{ subject to } (X_j - X_i).\rho \leq \frac{-1}{1-\beta} \forall j \neq i \text{ and } j \leq r \tag{3}$$

Note that since we are on the hyperplane $\mathcal{H}$, the condition is actually equivalent to $\rho.X_{j,\cdot} \leq \frac{-\beta}{1-\beta}$

The dual of Problem 3 is

$$\text{maximize } \frac{1}{1-\beta} \sum_{j \neq i,\ j \leq r} u_j \text{ subject to } || \sum_{j \neq i,\ j \leq r} u_j (X_j - X_i)||_1 \leq 1,\ u \geq 0 \tag{4}$$

Let $u \neq 0$ dual feasible. Then we have

$$\begin{aligned} || \sum_{j \neq i,\ j \leq r} u_j (X_j - X_i)||_1 \leq 1 \quad &\implies \quad \sum_{k \neq i,\ k \leq r} u_k || \sum_{j \neq i,\ j \leq r} \frac{u_j}{\sum_{k \neq i,\ k \leq r} u_k} (X_k - X_i)||_1 \leq 1 \\ &\implies \quad \sum_{k \neq i,\ k \leq r} u_k \alpha \leq 1 \ (\alpha - robust\ separable) \\ &\implies \quad \sum_{k \neq i,\ k \leq r} u_k \leq \frac{1}{\alpha} \end{aligned}$$

Therefore, $\frac{1}{1-\beta} \sum_{j \neq i,\ j \leq r} u_j \leq \frac{1}{\alpha(1-\beta)}$. Hence, $d^* \leq \frac{1}{\alpha(1-\beta)}$. Besides, using Lemma 3.1, we know we can find a vector ρ strictly feasible, therefore, since the problem is convex, Slater's conditions are met. Hence strong duality holds. So $p^* = d^* \leq \frac{1}{\alpha(1-\beta)} < \frac{1}{8\epsilon}$. Hence we can find ρ satisfying the conditions described before. Let us take such ρ_i for the rest of the proof.

Let C a feasible point for the linear program of the algorithm. Now let us show that $C_{i,i} > 1/2$. From what was said, we know that $C \in \Phi_{4\epsilon}(X)$. Therefore we only need to show that the solution of the following problem is stricly larger than 1/2.

$$\begin{aligned} \underset{C}{\text{minimize}} \quad & e_i^T diag(C) = C_{i,i} \\ \text{subject to} \quad & C \in \Phi_{4\epsilon}(X) \end{aligned} \tag{5}$$

Where e_i is the i-th element of the canonical basis. We use the notations of section 3.1 about the dual. We take R such that $R_{i,\cdot} = \rho_i$ and $R_{j,\cdot} = 0$, $M = 0$, $\mu = 0$ and $\eta = 0$. To prove that the assignment is feasible, one only needs to prove that $\forall k \leq f,\ k \neq i,\ X_k.\rho \leq 0$, which is straightforward using the properties of ρ_i and the condition $||M||_{\infty,1} \leq \beta$. After computation, we find that for that dual feasible assignment, we have $d > 1/2$. Since we know that $p^* \geq d$ by weak duality, we have proved the result, that is to say that $C_{i,i} > 1/2$ for all $C \in \Phi_{4\epsilon}(X)$

Now let us show that $\forall j > r,\ C_{j,j} \leq 1/2$. To do so, we only need to show that the solution of the following program is larger than $-1/2$ (since the entries of C are nonnegative)

$$\begin{aligned} \underset{C}{\text{minimize}} \quad & -\sum_{j>r} e_j^T diag(C) = -\sum_{j>r} C_{j,j} \\ \text{subject to} \quad & C \in \Phi_{4\epsilon}(X) \end{aligned} \tag{6}$$

Like previously we show the result by considering the dual feasible assignment such that $\forall i \leq r,\ R_{i,\cdot} = \rho_i, \forall j > r,\ R_{j,\cdot} = 0$, $M = 0$, $\gamma = 0$ and $\eta = -1$. The cost of that assignment being larger than $-1/2$.

Therefore, Algorithm 2, by finding in particular a point $C \in \Phi_{4\epsilon}(X)$ allows us to find the hott features of X. Let us note $X = F_0 W_0$ the NMF of X. So we found $\hat{W}$ such that $||\hat{W} - W_0||_{\infty,1} \leq \epsilon$. Therefore we have that $||F_0\hat{W} - F_0 W_0||_{\infty,1} \leq \epsilon$. Finally, we have when noting $\hat{F}$ the matrix returned by the algorithm

$$\begin{aligned} ||\hat{X} - \hat{F}\hat{W}||_{\infty,1} &\leq ||\hat{X} - F_0\hat{W}||_{\infty,1} \text{ (Since by definition } \hat{F} \text{ is the minimizer of the distance)} \\ &\leq ||\hat{X} - X||_{\infty,1} + ||X - F_0\hat{W}||_{\infty,1} \\ &\leq 2\epsilon \quad \blacksquare \end{aligned}$$

We can notice that in the previous case, where the simplicial rows of W appear only once in X, in the step 2 of Algorithm 2, we actually don't need to find the solution of the optimization program, we only need to find a feasible point. However, in the general case (when we allow copies), solving the program is necessary.

Theorem 3.2 *Suppose that ϵ and α are positive constants such that $\epsilon \leq \frac{\alpha^2}{8+4\alpha}$. Suppose $\hat{X} = X + \Delta$ where X admits an $\alpha-$robust simplicial NM. Then Algorithm 2 constructs a NMF $\hat{F}\hat{W}$ such that $||\hat{X} - \hat{F}\hat{W}||_{\infty,1} \leq 4\epsilon$*

4 Experimentation and Incremental Gradient Algorithms for NMF

We are going to solve the previous optimization question by a fast implementation using two techniques which are dual decomposition and incremental gradient descent. Actually, instead of solving completely the dual problem, we can separate the constraints of Φ_τ into two parts and solve the relaxed optimization with respect to the first set of constraints which are : $||CX - X||_{\infty,1} \leq \tau$, $Tr(C) = r$, then project to the space defined by the constraints left. We can notice that this space is actually

$$\Phi_0(X) = \{C \geq 0 : \text{ diag}(C) \leq 1, \forall i, j \; C_{i,j} \leq C_{j,j}\}$$

Lagrangian over the constraint set Φ_0 can be written as:

$$\mathcal{L}(\mathbf{C}, \beta, \mathbf{w}) = \mathbf{p}^T diag\mathbf{C} + \beta(Tr(\mathbf{C}) - r) + \sum_{i=1}^{f} w_i(||\mathbf{X}_{i.} - [\mathbf{CX}]_{i.}|| - \tau)$$

What we need to do is to alternate between minimizing the Lagrangian over Φ_0 and then taking a subgradient step with respect to the dual variables.

$$w_i \leftarrow w_i + s(||\mathbf{X}_{i.} - [\mathbf{C}^*\mathbf{X}]_{i.}|| - \tau) \qquad \beta \leftarrow \beta + s(Tr(C^*) - r)$$

According to the article, the update of w_i does not make influential difference with respect to the solution quality, so we typically only update β.

While we minimize the Lagrangian, we use the projected incremental gradient descent method. It signifies to minimize $\mathcal{L}$ using a stochastic gradient descent, after that we project the minimizer C of gradient descent onto the allowed set Φ_0.

Notice that the Lagrangian can be rewritten as

$$\mathcal{L}(\mathbf{C}, \beta, \mathbf{w}) = -\tau \mathbf{1}^T \mathbf{w} - \beta r + \sum_{k=1}^{n} \left(\sum_{j=1}^{f} w_i(||\mathbf{X}_{jk} - [\mathbf{CX}]_{jk}||_1 + \frac{1}{n}(p_j + \beta)C_{jj} \right)$$

So the principal part of algorithm 3 is a loop of N iterations, in each iteration, we use a SGD to minimize the Lagrangian $\mathcal{L}$ with respect to C, then project C on Φ_0, after that update the dual variables β. According to previous theoretical studies, W is composed of r rows of X where the indexes of those rows correspond to the r largest diagonal elements' indexes of C. Once we get W, we can solve a simple minimization problem to get F.

The projection on Φ_0 is given by the following algorithm. Actually we proceed the projection by columns of C, since each column is independent.

Algorithm 3 Hottopixx: Approximate Separable NMF by Incremental Gradient Descent

Input: An $f \times n$ nonnegative matrix X. Primal and dual stepsizes s_p and s_d.
Output: An $f \times r$ nonnegative matrix F and $r \times n$ nonnegative matrix W such that $||X - FW||_{\infty,1} \leq 4\varepsilon$.

1. Choose a cost $\mathbf{p}$ with distinct entries
2. Initialize $\mathbf{C} = 0, \beta = 0$
3. for $t = 1, ..., N_{iteration}$ do
4. for $i = 1, ...n$ do
5. Choose k uniformly at random from $[1, .., n]$
6. $\mathbf{C} \leftarrow \mathbf{C} + s_p \cdot sign(\mathbf{X}_{.k} - \mathbf{C}\mathbf{X}_{.k}X_{.k}^T) - s_p \cdot \ textdiag(\mu \cdot (\beta\mathbf{1} - \mathbf{p}))$
7. end for
8. Project C onto Φ_0
9. $\beta \leftarrow \beta + s(Tr(C) - r)$
10. end for
11. Let I be the set of index of r largest elements in $\{C_{ii}\}$
12. $F = \underset{Z \in \mathcal{M}_{f,r}(\mathbb{R})}{\arg\min} \ ||X - ZW||_{\infty,1}$
13. return F,W

For instance we want to project the i-th column $C_{.i}$. Let $\mathbf{z}$ denote this vector. We want then to project z in a set $\{\mathbf{x} \in \mathbb{R}^f : 0 \leq x_j \leq x_i \forall j \neq i, \text{and} x_i \leq 1\}$. Without loosing the generality, we can illustrate an example of projection with $i = 1$, and in addition suppose the other elements of $\mathbf{z}$ is sorted as $z2 \geq z3 \geq ... \geq z_n$. Using the following "Column Squishing" algorithm

$$minimize \quad \frac{1}{2}||z - x||^2 \tag{7}$$

$$\text{subject to} \quad 0 \leq x_j \leq x_1 \forall j, x_i \leq 1 \tag{8}$$

Algorithm 4 Column Squishing

Input: A vector $z \in \mathbb{R}^f$ with $z2 \geq z3 \geq ... \geq z_n$.
Output: The projection of z onto $\{z \in \mathbb{R}^f : 0 \leq x_i \leq x_1 \forall i, \text{and } x_1 \leq 1\}$.

1. $\mu \leftarrow z_1$
2. for $k = 2, .., f$ do
3. if $z_k \leq \Pi_{[0,1]}(\mu)$, Set $k_c = k - 1$ and break
4. else set $\mu = \frac{k-1}{k}\mu + \frac{1}{k}z_k$
5. end for
6. $x_1 \leftarrow \Pi_{[0,1]}(\mu)$
7. for $k = 2, ..., k_c$, set $x_k = \Pi_{[0,1]}(\mu)$
8. for $k = k_c + 1, ..., f$, set $x_k = (z_k)_+$
9. return x

Expirimentation in Python
We used these two algorithms in solving small-scale problems. So the AGKM performs well and even better in solving matrix decomposition with noise in terms of precision. The Matrices F and W to make X in our experimentation are:

$$F = \begin{bmatrix} 0.25 & 0.75 \\ 1 & 0 \\ 0.3 & 0.7 \\ 0.5 & 0.5 \\ 0.9 & 0.1 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad W = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Then we can compute the object X by a simple matrix product of FW with a gaussian noise or not.

References

[1] Recht, B., Re, C., Tropp, J., Bittorf, V. (2012). Factoring nonnegative matrices with linear programs. In Advances in Neural Information Processing Systems (pp. 1214-1222).

[2] Arora, S., Ge, R., Kannan, R., Moitra, A. (2012, May). Computing a nonnegative matrix factorization–provably. In Proceedings of the forty-fourth annual ACM symposium on Theory of computing (pp. 145-162). ACM.