A Projective Approach to Nonnegative Matrix Factorization

Patrick Groetzner∗
April 23, 2019

Abstract

Nonnegative matrix factorization as a tool in data science to analyse the structure of the underlying dataset appears in various applications and enjoys great popularity. Consider a given square matrix $A$. The symmetric nonnegative matrix factorization aims for a nonnegative low-rank approximation $A \approx XX^T$ to $A$, where $X$ is entrywise nonnegative and of given order. This setting can be seen as demanding a so-called completely positive approximation of $A$. In this paper we introduce an alternating projection type approach to this setting in order to obtain symmetric nonnegative matrix factorizations. Moreover, considering a general rectangular input matrix $A$, the general nonnegative matrix factorization again aims for a nonnegative low-rank approximation to $A$ which is now of the type $A \approx XY$ for entrywise nonnegative matrices $X,Y$ of given order. Here we introduce a new perspective motivated by our results in the symmetric case in order to derive nonnegative matrix factorizations even in this general setting.

Keywords: nonnegative matrix factorization, symmetric nonnegative matrix factorization, low-rank approximation, completely positive matrices

Mathematical Subject Classification 2010: 15B10, 15A23, 15B48

1 Introduction

Throughout the article, let $\mathbb{R}_+^n$, resp. $\mathbb{R}_+^{n \times m}$ denote the sets of entrywise nonnegative vectors and matrices, respectively. We write $S_n$ for the set of symmetric matrices of order $n$ and $S_n^+$ for the set of symmetric positive semidefinite matrices. Moreover, let $\| \cdot \|_F$ denote the Frobenius norm.

Nonnegative matrix factorization (NNMF) aims for a nonnegative approximation to a given matrix which is at the same time a low-rank-approximation of given rank. To be more precise, the NNMF considers the following setting, cf. [17] Equation 3 or [32] Equation 2.

Definition 1.1. Let $A \in \mathbb{R}_+^{n \times m}$, $k \ll \min \{n, m\}$ and assume $k \leq \text{rank}(A)$, then the solution matrices $X \in \mathbb{R}_+^{n \times k}$ and $Y \in \mathbb{R}_+^{k \times m}$ of the problem

$$
\min_{X \in \mathbb{R}_+^{n \times k}, Y \in \mathbb{R}_+^{k \times m}} \|A - XY\|^2_F
$$

yield a nonnegative matrix factorization $XY$ of $A$.

∗University of Augsburg, 86135 Augsburg, Germany. Email: patrick.groetzner@math.uni-augsburg.de
Note that a NNMF is not unique in general, since for any NNMF $XY$ of $A$ and for any nonnegative matrix $Z$ with $Z^{-1}$ being entrywise nonnegative as well, $\tilde{X} := XZ$ and $\tilde{Y} := Z^{-1}Y$ also provide a NNMF of $A$. Adding further constraints to the problem in Definition 1.1 or slight changes in the objective function lead to various specially structured NNMF problems. If we allow $A$ and one of the matrices $X,Y$ to have negative entries, this defines the Semi NNMF, cf. [32, Section 2.2]. This problem is motivated by data clustering. For the Sparse NNMF, cf. [32, Sections 2.7-2.9], we add the (possibly weighted) penalty term $\sum_{i,j} Y_{ij}$ to the objective function to ensure sparsity of the matrix $Y$. Also the following symmetric case can be seen as a special case of the problem in Definition 1.1.

Let $A \in \mathbb{R}_{+}^{n \times n}$ be symmetric. To introduce the symmetric nonnegative matrix factorization (SymNNMF) of $A$, consider the following setting, cf. [5], [20].

**Definition 1.2.** Given $A \in \mathcal{S}_n$, let $k \ll n$ and assume $k \leq \text{rank}(A)$, the solution matrix $X \in \mathbb{R}_{+}^{n \times k}$ of

$$\min_{X \in \mathbb{R}_{+}^{n \times k}} \|A - XX^T\|_F^2$$

yields a symmetric nonnegative matrix factorization $XX^T$ of $A$.

The SymNNMF is related to data clustering, particularly Kernel K-means clustering and Laplacian-based spectral clustering, as discussed in [11]. As a concrete example it can be used to analyze the structure of a given dataset, like facial poses as shown in [19] or heterogeneous microbiome data, as introduced in [26].

There exist several approaches to compute a SymNNMF of a given matrix $A$ and given order $k$. For an algorithm based on certain update rules, the reader is referred to the approach in [11]. Newton-like methods for SymNNMF can be found in [22, Section 3]. Borhani et al. [5] introduce an accelerated proximal gradient method and an alternating direction approach. All of these approaches are heuristic in the sense that convergence is not guaranteed.

Coming back to the more general framework of NNMF in Definition 1.1 a broad field of applications is connected to this problem. For instance some aspects in environmental [29] and meteorological sciences [31] demand a NNMF. Moreover, the NNMF appears in the analysis of financial data [12] and biomedical applications [17], where it helps to classify cancer cells.

NNMF itself can be seen as a special subclass of so called constrained low-rank matrix approximation problems as introduced in [17]. Therefore, various applications of the NNMF approach are related to this topic. One very illustrative application is hyperspectral imaging, cf. [17], where every pixel of a hyperspectral image is represented via more than 100 channels which correspond to deeper information of several wavelengths of the image, some of them invisible to the human eye. This approach boils down to the NNMF framework. See also [16].

Moreover, in the context of data science, NNMF can be used for so called intelligent data analysis, as shown in [9]. Especially when the quantities are known to be nonnegative, for example due to physical laws, NNMF can be used to determine part-based representations of given data. Here a concrete example, again given in [9], is educational data mining. For a survey on this topic, the reader is referred to [27]. Here the goal is to collect, store and analyse data obtained from learning and evaluation processes of students.
Other possible applications are multi-document summarization, cf. [8] and analysis of magnetic resonance spectroscopy data, cf. [23]. As already mentioned for the symmetric case, NNMF is closely related to data clustering, cf. [24].

The most popular approach in the context of nonnegative matrix factorization is to use certain multiplicative update rules as first introduced in [25], see also [11]. Here we should note that we can not modify zero entries and the method is not guaranteed to converge to a stationary point. The authors in [4] provide a summary of the common NNMF methods, like gradient descent methods, where only little can be said about the convergence of these methods, see also [20, Section 3.3]. Here the idea is to rewrite the NNMF problem as a convex problem over a nonconvex set. Further, adding the nonnegativity projection makes the analysis even more difficult. Alternating least squares methods, as in [29], are dealing with the decomposition matrices \( X \) and \( Y \) in an alternating manner. Here no general convergence result is known.

In contrast to the existing literature, this paper introduces a new view on the NNMF setting in a different alternating manner and, for the symmetric case, based on so-called completely positive matrices whose connection to the SymNNMF setting is shown in Section 2. In Section 3, we will formulate the NNMF and the SymNNMF problem as feasibility problems which pave the way to new approaches working with so called alternating projection frameworks. Moreover, in the symmetric case, we provide a local convergence result for this new method and furthermore provide a heuristic extension which avoids numerically expensive computations and, as numerical evidence substantiates, works well for most test instances. For the general rectangular case in Section 3.2 we extend the results of the symmetric case and again introduce a new method to obtain a NNMF. In addition, numerical experiments will be shown in Section 5 to illustrate the performance of the methods in concrete settings.

2 Completely Positive Matrices and their Relation to Symmetric Nonnegative Matrix Factorization

A symmetric matrix \( A \in \mathbb{R}^{n \times n} \) is called completely positive if there exists an entrywise nonnegative matrix \( X \in \mathbb{R}^{n \times r} \) such that \( A = XX^T \). We call such a factorization a cp-factorization of \( A \). The set of all completely positive matrices,

\[
CP_n := \{ A \in \mathbb{R}^{n \times n} \mid A = XX^T \text{ where } X \in \mathbb{R}^{n \times r}, X \geq 0 \} = \text{conv}\{xx^T \mid x \in \mathbb{R}^n_+ \},
\]

is a proper cone whose extreme rays are the rank-one matrices \( xx^T \) with \( x \in \mathbb{R}^n_+ \), cf. [1]. The minimal possible number of columns \( r \) in the factorization matrix \( X \) is called the cp-rank \( \text{cpr}(A) \) and in general we have \( \text{cpr}(A) \gg \text{rank}(A) \). Note that it may well happen (and often does) that \( \text{cpr}(A) > n \). Thus, completely positive matrix factorization is a special case of the SymNNMF, albeit without the low rank constraint. In a cp-factorization, we have \( k \geq \text{cpr}(A) \) and for the SymNNMF, \( k \leq \text{rank}(A) \) is required.

Nevertheless, the problem in Definition 1.2 can be rewritten as

\[
\min_{B \in CP_n, \ cpr(B)=k} \| A - B \|_F^2.
\]

3
Thus, the SymNNMF seeks for the best completely positive approximation of cp-rank $k$ to $A$. Since it is not possible to compute the cp-rank of a given completely positive matrix in general, cf. [2], we will use the rank of the matrix as a lower bound for the cp-rank. Hence, we will try to factorize a rank-$k$ approximation of $A$ instead of $A$ itself.

3 Nonnegative Matrix Factorization as a Feasibility Problem

To introduce our method to determine a NNMF or a SymNNMF of a given matrix $A$, a first step is to compute the best rank-$k$-approximation of $A$. Here we use the well known theorem by Eckart and Young in [15] which was proven to hold for any unitarily invariant norm by Mirsky in [28]. Let $O_k$ denote the set of orthogonal matrices of order $k$ and for a given matrix $A$, let $A_{i\ast}$ resp. $A_{\ast j}$ denote the $i$-th row resp. the $j$-th column of $A$.

**Theorem 3.1.** Let $A \in \mathbb{R}^{n \times m}$ with $\text{rank}(A) = l$ and consider its singular value decomposition $A = U \Sigma V^T$, where $U \in O_n$, $V \in O_m$.

$$
\Sigma = \begin{pmatrix}
\sigma_1 & \cdots & \cdots \\
\vdots & \ddots & \vdots \\
\cdots & \cdots & \sigma_l
\end{pmatrix} \in \mathbb{R}^{n \times m},
$$

$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_l > 0$ are the singular values of $A$. So $A$ can be written as

$$A = \sum_{j=1}^{l} \sigma_j U_{\ast j} V_{\ast j}^T.$$

Let $k \leq \text{rank}(A)$, then

$$A_k := \sum_{j=1}^{k} \sigma_j U_{\ast j} V_{\ast j}^T$$

is the best rank-$k$ approximation (in the Frobenius norm) of $A$, i.e.

$$A_k \in \text{Argmin} \left\{ \| A - X \|_F^2 \mid X \in \mathbb{R}^{n \times m} \text{ with } \text{rank}(X) \leq k \right\}$$

with corresponding minimal value

$$\| A - A_k \|_F^2 = \sum_{j=k+1}^{m} \sigma_j^2.$$

Moreover, if $\sigma_k > \sigma_{k+1}$, then $A_k$ is the unique global minimizer.

In the following $A_k$ will denote the best rank-$k$-approximation to $A$. The subsequent lemmas
now indicate that it is sufficient to determine a nonnegative factorization of $A_k$ to obtain a NNMF or a SymNNMF in the symmetric case.

**Lemma 3.2.** Let $A \in \mathbb{R}_+^{n \times m}$ and $k \leq \text{rank}(A)$. Further let $A_k$ as in Theorem 3.1. Then any factorization $A_k = XY$ with $X \in \mathbb{R}_+^{n \times k}$ and $Y \in \mathbb{R}_+^{k \times m}$ gives a NNMF of $A$.

**Proof.** Let $A_k = XY$ be a nonnegative factorization of the matrix $A_k$. Then

$$A_k \in \text{Argmin} \left\{ \|A - X\|_F^2 \mid X \in \mathbb{R}^{n \times m} \text{ with } \text{rank}(X) \leq k \right\}$$

due to Theorem 3.1. Thus, $XY \in \text{Argmin} \left\{ \|A - X\|_F^2 \mid X \in \mathbb{R}^{n \times m} \text{ with } \text{rank}(X) \leq k \right\}$ and since $X \in \mathbb{R}_+^{n \times k}$ and $Y \in \mathbb{R}_+^{k \times m}$, we get $XY$ as a nonnegative factorization of $A$. $\square$

Thus, to obtain a NNMF of $A$, it is sufficient to factorize $A_k = XY$, where $X \in \mathbb{R}_+^{n \times k}$ and $Y \in \mathbb{R}_+^{k \times m}$.

**Lemma 3.3.** Let $A \in S_n$, $k \leq \text{rank}(A)$ and let $A_k$ as in Theorem 3.1. Further assume that $A_k \in \mathbb{C}P_n$. Then any cp-factorization $A_k = X X^T$ with $X \in \mathbb{R}^{n \times k}$ of $A_k$ is a solution to the problem in Definition 1.2 and hence a SymNNMF of $A$ of rank $k$.

**Proof.** Let $A_k = X X^T$ be a cp-factorization of $A_k$ with $X \in \mathbb{R}^{n \times k}$. Then

$$A_k \in \text{Argmin} \left\{ \|A - X\|_F^2 \mid X \in \mathbb{R}^{n \times m} \text{ with } \text{rank}(X) \leq k \right\}$$

due to Theorem 3.1. Thus, $X X^T \in \text{Argmin} \left\{ \|A - X\|_F^2 \mid X \in \mathbb{R}^{n \times m} \text{ with } \text{rank}(X) \leq k \right\}$ and since $X \in \mathbb{R}_+^{n \times k}$, we get that $X X^T$ is a SymNNMF of $A$. $\square$

To compute a SymNNMF, it is therefore sufficient to find a cp-factorization of $A_k$ of order $n \times k$.

To compute a nonnegative factorization of $A_k$ as in Lemmas 3.2 or 3.3, we will start with an arbitrary initial factorization of $A_k$. Let $A = U \Sigma V^T$ be the SVD of $A$. In the rectangular case, we use the SVD $A_k = U_k \Sigma_k V_k^T$, where the matrices $U_k$, $\Sigma_k$, $V_k$ are the truncated versions of $U$, $\Sigma$ and $V$.

$$X_k := U_k \sqrt{\Sigma_k} \in \mathbb{R}_+^{n \times k} \text{ and } Y_k := \sqrt{\Sigma_k} V_k^T \in \mathbb{R}_+^{k \times m}. \quad (1)$$

now defines an initial factorization $A = X_k Y_k$ which is not necessarily entrywise nonnegative.

For the symmetric case, one can for instance use the Cholesky decomposition $A_k = X_k^2$, where $X_k$ is a lower triangular matrix, or the spectral decomposition $A_k = V \Sigma V^T$ by setting

$$X_k := V \Sigma^\frac{1}{2} \quad (2)$$

to obtain an initial factorization.

In both settings, we assume the the matrices $X_k$ and $Y_k$ to have negative entries. If this is not the case, we already generated a NNMF or a SymNNMF, respectively. Our goal now is to transform the initial factorizations into nonnegative ones.

First, we focus on the symmetric case, where we use the following tool to compute a completely positive factorization of $A_k$, cf. [18, Lemma 2.5].
Lemma 3.4. Let $X_k, Z_k \in \mathbb{R}^{n \times k}$. Then $X_k X_k^T = Z_k Z_k^T$ if and only if there exists $Q \in \mathcal{O}_k$ with $X_k Q = Z_k$.

So to transform the factorization $A_k = X_k X_k^T$ into a nonnegative factorization $A_k = Z_k Z_k^T$, we have to solve the following feasibility problem:

$$\begin{align*}
\text{find } & Q \\
\text{s.t. } & X_k Q \geq 0 \\
& Q \in \mathcal{O}_k.
\end{align*}$$

This problem is feasible if and only if $A_k \in \mathcal{CP}_n$ with $\mathrm{cpr}(A_k) \leq k$. Introducing the polyhedral cone $\mathcal{P} := \{Q \in \mathbb{R}^{k \times k} | X_k Q \geq 0\}$, we rewrite (3) as

$$\text{find } Q \in \mathcal{O}_k \cap \mathcal{P}.$$

Hence, we reduced the task to compute a SymNNMF to solving Problem (4).

In the general nonsymmetric case, Lemma 3.4 does not apply, as the following example substantiates. As it turns out, we need to replace orthogonal matrices by a more general tool.

Example 3.5. Let

$$X = \begin{pmatrix} 6 & 0 \\ 0 & 6 \end{pmatrix}, \quad Y = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}, \quad G = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix} \quad \text{and} \quad H = \begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}.$$

Then we have $XY = GH$. Moreover, observe that the equations $XQ = G$ resp. $\hat{Q}Y = H$ have the unique solutions

$$Q = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \quad \text{resp.} \quad \hat{Q} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} = Q^{-1}.$$

This is the unique $Q$ for which $XQ = G$ and $Q^{-1}Y = H$. This $Q$ is nonsingular, but not in $\mathcal{O}_2$.

This now motivates the following result. In the following, for a given matrix $A$, let $R(A)$ resp. $S(A)$ denote the subspaces spanned by the rows resp. columns of $A$ and let $A^+$ denote the Moore-Penrose-inverse of $A$.

Lemma 3.6. Let $X, G \in \mathbb{R}^{n \times k}$ and $Y, H \in \mathbb{R}^{k \times m}$ all be of rank $k$. Then we have $XY = GH$ if and only if there exists a nonsingular matrix $Q \in \mathbb{R}^{k \times k}$ such that $XQ = G$ and $Q^{-1}Y = H$.

Proof. The if part is obvious. For the reverse part, observe that $X$ and $G$ are of the same rank. It is easy to see that there exists a linear map $f : R(X) \to R(G)$, $x^T \mapsto x^T A_f$ such that $f(X_i \ast) = X_i \ast A_f = G_i \ast$ for all $i \in \{1, \ldots, n\}$. Similarly, there exists a linear map $g : S(Y) \to S(H)$, $y \mapsto A_g y$ such that $g(Y_{\ast j}) = A_g Y_{\ast j} = H_{\ast j}$ for all $j \in \{1, \ldots, m\}$. Due to the equality $XY = GH$, we have

$$X_i \ast Y_{\ast j} = G_i \ast H_{\ast j} \quad \text{for every } i \in \{1, \ldots, n\} \text{ and } j \in \{1, \ldots, m\}.$$
Furthermore, since \( \text{rank}(X) = \text{rank}(G) = k \), the matrix \( A_f \) is nonsingular and \( f \) is bijective, so we have that for every \( i, j \)

\[
X_{is}Y_{sj} = f^{-1}(G_{is})Y_{sj} = G_{is}A^{-1}_f Y_{sj} \quad \text{and} \quad G_{is}H_{sj} = G_{is}A_g Y_{sj}.
\]

With the help of equation (5), we therefore get

\[
G_{is}A^{-1}_f Y_{sj} = G_{is}A_g Y_{sj},
\]

for every \( i = 1, \ldots, n \) and \( j = 1, \ldots, m \). Moreover, \( Y \) is of full row-rank such that \( YY^+ = I_{m \times m} \) and \( G \) is of full column-rank such that \( G^+G = I_{n \times n} \). Thus, equation (6) can be rewritten as

\[
GA^{-1}_f Y = GA_g Y \iff G^+GA^{-1}_f YY^+ = G^+GA_g YY^+ \iff A^{-1}_f = A_g,
\]

which completes the proof. \( \square \)

This lemma therefore shows that instead of using orthogonal matrices to transform one factorization of type \( A = XY \) into another one, we will apply the generalized result of Lemma 3.6 and we will therefore work with nonsingular matrices. For possible regularizations, we use the following approach, for which the proof follows by construction.

**Lemma 3.7.** Consider \( A \in \mathbb{R}^{n \times n} \) with \( \text{rank}(A) = k < n \) and its singular value decomposition \( A = U \Sigma V^T \). Further define a diagonal matrix \( \tilde{\Sigma} \in \mathbb{R}^{n \times n} \) via

\[
\tilde{\Sigma}_{ii} := \begin{cases} 
\Sigma_{ii}, & \text{if } i < k \\
\Sigma_{kk}, & \text{if } i \geq k
\end{cases}
\]

and therewith \( \tilde{A} := U \tilde{\Sigma} V^T \). Then \( \tilde{A} \) is nonsingular and \( A \) is a best rank-k-approximation of \( \tilde{A} \).

To again obtain a feasibility problem, we introduce the sets

\[
\mathcal{P}_{X_k} := \{ Q \in \mathbb{R}^{k \times k} \mid X_k Q \geq 0 \} \quad \text{and} \quad \mathcal{P}_{Y_k} := \{ Q \in \mathbb{R}^{k \times k} \mid Q \text{ is nonsingular and } Q^{-1} Y_k \geq 0 \}
\]

and note the following result.

**Theorem 3.8.** Let \( A \in \mathbb{R}^{n \times m} \) and \( k \leq \text{rank}(A) \). Consider \( A_k \) as in Theorem 3.1 and its initial factorization \( A_k = X_k Y_k \) as in (1). Then, to obtain a NNMF of \( A \) of rank \( k \), it is sufficient to find a matrix \( Q \in \mathcal{P}_{X_k} \cap \mathcal{P}_{Y_k} \).

**Proof.** If \( Q \in \mathcal{P}_{X_k} \cap \mathcal{P}_{Y_k} \), then \( Q \) is nonsingular by definition of \( \mathcal{P}_{Y_k} \). Moreover, we have

\[
A_k = X_k Y_k = \left( X_k Q \right) \left( Q^{-1} Y_k \right)
\]

as a NNMF of \( A \) due to Lemma 3.2, completing the proof. \( \square \)
We can therefore reduce the problem of finding a NNMF to solving the problem

\[ \text{find } Q \in \mathcal{P}_{X_k} \cap \mathcal{P}_{Y_k}. \]  

(8)

In summary, to compute a SymNNMF resp. a NNMF, it is sufficient to solve Problem (4) resp. (8).

3.1 Solving the Feasibility Problem in the Symmetric Case

To find a matrix in the intersection \( \mathcal{O}_k \cap \mathcal{P} \), the method of alternating projections can be used, cf. [10]. Here the idea is to project a starting point orthogonally onto the first set and the resulting element onto the second set and to repeat this procedure in order to obtain an intersection point in the limit. To apply this method, we need to be able to project onto the sets \( \mathcal{P} \) and \( \mathcal{O}_k \). For the latter, we will use the so-called polar decomposition \( M = SQ \) for a given matrix \( M \), where \( S \in \mathbb{S}^n_+ \) and \( Q \in \mathcal{O}_k \) is of minimal distance to \( M \) among all matrices in \( \mathcal{O}_k \). The projection of a matrix \( M \) onto \( \mathcal{P} = \{ Q \in \mathbb{R}^{k \times k} \mid X_k Q \geq 0 \} \) can be written as a so-called second order cone problem (SOCP), cf. [18]:

\[
\begin{align*}
\min & \quad \| Y - M \| \\
\text{s.t.} & \quad X_k Y \geq 0
\end{align*}
\]

\[
\iff
\begin{align*}
\min & \quad t \\
\text{s.t.} & \quad X_k Z \geq -X_k M \\
\quad & \quad (t, \text{vec}(Z)) \in \text{SOC},
\end{align*}
\]

where \( \text{vec}(Z) \in \mathbb{R}^{k^2} \) is the vector that results from stacking all columns \( Z \in \mathbb{R}^{k \times k} \) on top of each other, and \( \text{SOC} := \{ (t, x) \in \mathbb{R} \times \mathbb{R}^{k^2} \mid \| x \|_2 \leq t \} \) is the second order cone. Such second order cone problems are a well-studied class of optimization problems. They can be solved in polynomial time by using interior point methods, see e.g. [6].

The alternating projection approach now motivates Algorithm 1. Here \( \Pi \) represents the orthogonal projection operator.

**Algorithm 1 Symmetric Nonnegative Matrix Factorization based on CP-Factorizations**

**Input:** \( A \in \mathbb{R}^{n \times n} \) symmetric with its singular value decomposition \( A = UV^T; k \leq \text{rank}(A) \); initial matrix \( Q_0 \in \mathcal{O}_k \)

1. \( A_k \leftarrow \sum_{j=1}^{k} \sigma_j U_{*j} V_{*j}^T \)
2. Compute \( X_k \) as in (2)
3. \( Q \leftarrow Q_0 \)
4. while \( X_k Q \not\succcurlyeq 0 \) do
5. \( P \leftarrow \Pi_{\mathcal{P}}(Q) \)
6. \( Q \leftarrow \Pi_{\mathcal{O}_k}(P) \)
7. end while

**Output:** \( Q \in \mathcal{O}_k \) and a symmetric nonnegative matrix factorization \( A_k = (X_k Q)(X_k Q)^T \) of \( A \).

In step 1 of Algorithm 1, we compute the best rank-\( k \) approximation \( A_k \) to \( A \), followed by the computation of \( X_k \) as in (2). In steps 3 to 6, the algorithm then uses the alternating projection approach to obtain a cp-factorization of \( A_k \). With the help of Lemma 3.3, we thus obtained a
We can give the following convergence result for this algorithm, based on the convergence result for the alternating projection approach among semialgebraic sets, cf. [13].

**Theorem 3.9.** Let \( A \in \mathbb{S}_n \) be such that its best rank-\( k \) approximation \( A_k \in \mathcal{CP}_n \) and assume \( \text{cpr}(A_k) \leq k \leq \text{rank}(A) \). Let \( A_k = X_kX_k^T \) be any initial factorization with \( X_k \in \mathbb{R}^{n \times k} \). Define \( \mathcal{P} := \{ Q \in \mathbb{R}^{k \times k} \mid X_kQ \geq 0 \} \). Then we have:

(a) \( \mathcal{P} \cap \mathcal{O}_k \neq \emptyset \).

(b) If started at a point \( Q_0 \) close to \( \mathcal{P} \cap \mathcal{O}_k \), then Algorithm [2] converges to a point \( Q^* \in \mathcal{P} \cap \mathcal{O}_k \).

In this case, \( A_k = (X_kQ^*)(X_kQ^*)^T \) is a completely positive factorization of \( A_k \), which is at the same time a SymNNMF of \( A \).

**Proof.** (a): It follows from \( A_k \in \mathcal{CP}_n \) and \( k \geq \text{cpr}(A) \) that there exists \( Y \in \mathbb{R}^{n \times k} \) such that \( Y \geq 0 \) and \( A_k = YY^T \). Since \( A_k = X_kX_k^T = YY^T \) and the matrices \( X_k, Y \) are of the same order, Lemma [3.4] implies that there exists \( Q \in \mathcal{O}_k \) such that \( X_kQ = Y \geq 0 \), i.e., \( Q \in \mathcal{P} \cap \mathcal{O}_k \).

(b): It is easy to see that both \( \mathcal{P} \) and \( \mathcal{O}_k \) are closed semialgebraic sets. Moreover, \( \mathcal{O}_k \) is bounded. The convergence result now follows by applying [13, Theorem 7.3].

The projection onto \( \mathcal{P} \) requires solving a SOCP, which is very costly. We therefore introduce an alternative approach which avoids solving an SOCP by computing approximations to the projection. Motivated by [18, Section 5], we project \( X_kQ \) onto the nonnegative orthant \( \mathbb{R}^{n \times k}_+ \)

\[ G_{ij} := \max \{ (X_kQ)_{ij}, 0 \} \quad \text{for all} \quad i = 1, \ldots, n \quad \text{and} \quad j = 1, \ldots, k. \]

If \( G = X_kQ \), we have \( Q \in \mathcal{P} \) and we are done. Otherwise, in order to get an approximation of \( \Pi_{\mathcal{P}}(Q) \in \mathbb{R}^{k \times k} \), we need to modify \( G \in \mathbb{R}^{n \times k} \). We define

\[ \hat{P} := X_k^+G + (I - X_k^+X_k)Q \in \mathbb{R}^{k \times k} \]

and take \( \hat{P} \) to approximate \( \Pi_{\mathcal{P}}(Q) \).

The reason behind this is that \( \hat{P} \in \text{Argmin} \{ \|X_kZ - G\|_F \mid Z \in \mathbb{R}^{n \times k} \} \) and among all minimizers \( \hat{P} \) is the one closest to \( Q \). Moreover, we get from the properties of \( X_k^+ \) that

\[ X_k\hat{P} = X_kX_k^+G + (X_k - X_kX_k^+X_k)Q = X_kX_k^+G. \quad (9) \]

If the equation \( X_kZ = G \) is solvable for \( Z \), then the matrix \( \hat{P} \) can be interpreted as the projection of \( Q \) onto the set \( \mathcal{L} := \{ Z \in \mathbb{R}^{k \times k} \mid X_kZ = G \} \subseteq \mathcal{P} \). Clearly, we then have \( \hat{P} \in \mathcal{P} \). This reasoning gives rise to Algorithm [2].

Clearly, if the algorithm terminates successfully, it returns a cp-factorization of \( A_k \) and hence a SymNNMF of \( A \). Moreover, since we do not have to solve SOCPs for this approach, the algorithm is numerically much cheaper. Nevertheless, we loose the local convergence result, but as numerical experiments will substantiate, the approach works well in most test instances.
As mentioned in Section 3.1, the projection onto $P$ initial matrix $Q$.

Due to the lack of symmetry in this setting, we also project approach as in the symmetric case. Here the question is how to project onto the sets $P$. Coming back to the Problem (8), a first idea would be to use a similar alternating projection tool.

Let $X_k$ and $Y_k$ be as defined in (1). As before, we start with some initial nonsingular matrix $Q_0$ and project $X_kQ_0$ onto the nonnegative orthant $\mathbb{R}^n_+$. We obtain

$$G_{ij} := \max \{(X_kQ_0)_{ij}, 0\} \quad \text{for all } i = 1, \ldots, n \text{ and } j = 1, \ldots, k. \quad (10)$$

Due to the lack of symmetry in this setting, we also project $Q_0Y_k$ onto the nonnegative orthant $\mathbb{R}_+^{k \times m}$ and obtain

$$H_{ij} := \max \{(Q_0Y_k)_{ij}, 0\} \quad \text{for all } i = 1, \ldots, k \text{ and } j = 1, \ldots, m. \quad (11)$$

As in Section 3.1 and in order to approximate $\Pi_{P_{X_k}}(Q) \in \mathbb{R}^{k \times k}$ and to obtain an element in $P_{Y_k} \subseteq \mathbb{R}^{k \times k}$, respectively, we now modify $G \in \mathbb{R}^{n \times k}$ respectively $H \in \mathbb{R}^{k \times m}$. This now leads to the following tool.

**Lemma 3.10.** Let $X_k$ and $Y_k$ be as in (1). Further let $G, H$ be as defined in (10) and (11). Then $X_k^+G = \text{Argmin}_{Z \in \mathbb{R}^{k \times k}} \{\|X_kZ - G\|_F\}$ and $HY_k^+ = \text{Argmin}_{Z \in \mathbb{R}^{k \times k}} \{\|ZY_k - H\|_F\}$.

**Proof.** First, we focus on the equation $X_kZ = G$ and assume that there exists a solution $Z$. It is well known, cf. [21, Theorem 2] and [30], that the complete set of solutions of this equation is given as

$$\left\{ Z = X_k^+G + (I - X_k^+X_k)T \ \bigg| T \in \mathbb{R}^{k \times k} \right\} = \{ X_k^+G \}, \quad (12)$$

where the equation holds since $X_k$ is of full column rank such that $X_k^+X_k = I$. Thus, $\|X_kZ - G\|_F = 0$ if and only if $Z = X_k^+G$. On the other hand, for the case where there does not exist a solution $Z$ to $X_kZ = G$, again the residual $\|X_kZ - G\|_F$ is minimal if and only if $Z = X_k^+G$. An analogous argument proves the result for the equation $ZY_k = H$. □
With this we can now give an approximation to \( Q \) in \( \mathcal{P}_{X_k} \) respectively \( \mathcal{P}_{Y_k} \), which can be easily computed.

**Lemma 3.11.**

(a) Let \( G \) be the projection of \( X_kQ \) onto \( \mathbb{R}^k \). If \( G = X_kQ \), then \( Q \in \mathcal{P}_{X_k} \). If \( G \neq X_kQ \) and the equation \( X_kZ = G \) is solvable for \( Z \), then \( X_k^+G \in \mathcal{P}_{X_k} \).

(b) Let \( H \) be the projection of \( QY_k \) onto \( \mathbb{R}^k \). If \( QY_k = H \) and \( Q \) is nonsingular, then \( Q^{-1} \in \mathcal{P}_{Y_k} \). On the other hand, let \( H \neq QY_k \) and assume that the matrix \( HY_k^+ \) is nonsingular. Further assume that the equation \( ZY_k = H \) is solvable for \( Z \). Then we have \( (HY_k^+)^{-1} \in \mathcal{P}_{Y_k} \).

**Proof.** (a) If \( G = X_kQ \), then \( X_kQ \geq 0 \), i.e., \( Q \in \mathcal{P}_{X_k} \). Otherwise, let \( G \neq X_kQ \) and assume that the equation \( X_kZ = G \) is solvable for \( Z \). Then Lemma 3.10 yields \( Z = X_k^+G \). Therefore, \( X_k^+G \) is the projection of \( Q \) onto the set \( \{ Z \in \mathbb{R}^k \mid X_kZ = G \} \). Since \( G \geq 0 \), we get \( X_k^+G \in \mathcal{P}_{X_k} \).

(b) If \( QY_k = H \) and \( Q \) is nonsingular, then \( QY_k \geq 0 \) and \( Q^{-1} \in \mathcal{P}_{Y_k} \) by definition. Otherwise, let \( H \neq QY_k \) and assume that \( ZY_k = H \) has a solution \( Z \). Hence, \( Z = HY_k^+ \) due to Lemma 3.10. Thus, \( HY_k^+ \) is the projection of \( Q \) onto the set \( \{ Z \in \mathbb{R}^k \mid ZY_k = H \} \). Since \( HY_k^+ \) is nonsingular by assumption and \( H \geq 0 \) by definition, we get \( (HY_k^+)^{-1} \in \mathcal{P}_{Y_k} \).

In addition, if the equation \( X_kZ = G \) does not have a solution, then \( Z := X_k^+G \) minimizes the residual \( \|X_kZ - G\|_F \). In this case, we get \( X_kZ = X_kX_k^+G \). Here \( X_kX_k^+ \neq I \) in general since \( X_k \) is not of full row-rank. Thus, it may happen that \( X_k^+G \notin \mathcal{P}_{X_k} \), however this does not seem to impair the good numerical performance.

If on the other hand the equation \( ZY_k = H \) does not have a solution, we get with Lemma 3.10 that \( Z := HY_k^+ \) minimizes the residual \( \|ZY_k - H\|_F \). Thus, \( ZY_k = HY_k^+Y_k \). In this equation we have \( Y_kY_k^+ \neq I \) in general since \( Y_k \) is not of full column rank. Hence, even if \( HY_k^+ \) is nonsingular, it may happen that \( (HY_k^+)^{-1} \notin \mathcal{P}_{Y_k} \). However, this does not seem to impair the good numerical performance either.

If we combine the results in Lemmas 3.10 and 3.11 we derive matrices in \( \mathcal{P}_{X_k} \) or \( \mathcal{P}_{Y_k} \) without solving an SOCP. Moreover, in Lemma 3.11 we assumed that \( HY_k^+ \) is nonsingular. This is equivalent to a certain rank assumption for \( H \).

**Lemma 3.12.**

(a) In our setting, the matrix \( HY_k^+ \in \mathbb{R}^{k \times k} \) is nonsingular if and only if \( H \in \mathbb{R}^{k \times m} \) is of rank \( k \).

(b) In addition, the matrix \( X_k^+G \in \mathbb{R}^{k \times k} \) is nonsingular if and only if \( G \in \mathbb{R}^{n \times k} \) is of rank \( k \).

**Proof.** We only prove (a), part (b) follows analogously: First observe that by Sylvester’s inequality, cf. [3] Corollary 2.5.10, we have

\[
\text{rank}(H) + \text{rank}(Y_k^+) - k \leq \text{rank}(HY_k^+) \leq \min\{\text{rank}(H), \text{rank}(Y_k^+)\}.
\]
Since $\text{rank}(Y_k^+)=\text{rank}(Y_k)=k$, this yields

$$\text{rank}(H) \leq \text{rank}(HY_k^+) \leq \min\{\text{rank}(H), k\}. \hspace{1cm} (13)$$

Now observe that $HY_k^+$ is nonsingular if and only if $\text{rank}(HY_k^+)=k$. Due to (13), this is true if and only if $\text{rank}(H)=k$.

From now on, we will take $X_k^+G$ respectively $(HY_k^+)^{-1}$ as approximations of $P_{P_{X_k}}(Q)$, respectively $P_{P_{Y_k}}(Q)$. This reasoning leads to Algorithm 3.

**Algorithm 3 Nonnegative Matrix Factorization**

**Input:** $A \in \mathbb{R}^{n \times m}$ with its singular value decomposition $A = U \Sigma V^T$; $k \leq \text{rank}(A)$; initial nonsingular matrix $Q \in \mathbb{R}^{k \times k}$

1. $A_k \leftarrow \sum_{j=1}^{k} \sigma_j U s_j V_{kj}^T$
2. Compute $X_k$ and $Y_k$ as in (1)
3. while $X_kQ \not\geq 0$ or $Q^{-1}Y_k \not\geq 0$ do
4. $G \leftarrow \max\{X_kQ, 0\}$ entrywise
5. $Q_G \leftarrow X_k^+G$
6. if $\text{rank}(G) < k$ then
7. $Q_G \leftarrow \tilde{Q}_G$ according to Lemma 3.7
8. end if
9. $H \leftarrow \max\{Q_G^{-1}Y_k, 0\}$ entrywise
10. $Q_H \leftarrow HY_k^+$
11. if $\text{rank}(H) < k$ then
12. $Q_H \leftarrow \tilde{Q}_H$ according to Lemma 3.7
13. end if
14. $Q \leftarrow Q_H^{-1}$
15. end while

**Output:** Nonsingular matrix $Q$ and a nonnegative matrix factorization $(X_kQ)(Q^{-1}Y_k)$ of $A$.

In step 3, the main while loop of the algorithm starts. As long as the matrices $X_kQ$ and $Q^{-1}Y_k$ have some negative entries, we at first define $G$ as the entrywise maximum introduced in (10). In step 5, we then look for a solution $Q_G$ of $\min\|X_kQ_G - G\|_F$ with the help of Lemma 3.11. Moreover, if the equation $X_kQ = G$ is solvable for $Q$, we get that $X_k^+G \in P_{X_k}$. To obtain a matrix in $P_{X_k} \cap P_{Y_k}$, and since the set $P_{Y_k}$ is based on the inverse of the considered matrices, we check in step 6 whether $X_k^+G$ is nonsingular, based on the result in Lemma 3.12 and regularize $X_k^+G$ if necessary. Using its inverse matrix as the matrix $Q_0$ in (11), we obtain the matrix $H$ in step 9 as the introduced entrywise maximum. In step 10, we then look for a solution $Q_H$ of the problem $\min\|Q_HY_k - H\|_F$ again with the help of Lemma 3.11 and check whether $Q_H$ is nonsingular using Lemma 3.12. According to Lemma 3.11, we know that if $Q_H$ is nonsingular and the equation $Q_HY_k = H$ is solvable, we have $Q_H^{-1} \in P_{Y_k}$. If necessary, we regularize the matrix $Q_H$ and take ITS inverse as our next iterate in step 14.

**Remark 3.13.** Consider $A_k$ as in Theorem 3.7. A necessary assumption for Algorithm 3 to terminate successfully is that there exists an exact factorization $A_k = X_kY_k$ with $X_k \in \mathbb{R}_+^{n \times k}$ and $Y_k \in \mathbb{R}_+^{k \times m}$. Clearly, this can only be true if $A_k \in \mathbb{R}_+^{n \times m}$.
4 Numerical Results for Symmetric Nonnegative Matrix Factorization

The following numerical results were carried out on a computer with 88 Intel Xenon ES-2699 cores (2.2 Ghz each) and a total of 0.792 TB Ram. The algorithms were implemented in MatlabR2017a, the SOCPs in Algorithm 1 were solved using Yalmip R20170626 and SDPT3 4.0. Whenever we update $Q$ in either of the algorithms, we increase the iteration counter. The algorithms terminate successfully at iteration $i$ if $X_kQ_i \geq -10^{-15}$, it terminates unsuccessfully if a maximum number of iterations (usually 5000) is reached.

As a first example, we consider the following matrix, cf. [14]

$A_{DS} = \begin{pmatrix} 8 & 5 & 1 & 1 & 5 \\ 5 & 8 & 5 & 1 & 1 \\ 1 & 5 & 8 & 5 & 1 \\ 1 & 1 & 5 & 8 & 5 \\ 5 & 1 & 1 & 5 & 8 \end{pmatrix} \in \mathbb{CP}_5.$

In the following we will show that Algorithms 1 and 2 return a SymNNMF for $k = 2$ of this matrix. First consider the best rank-2 approximation $A_2$ to $A_{DS}$ due to Theorem 3.1:

$A_2 = \begin{pmatrix} 7.7889 & 5.1708 & 0.9348 & 0.9348 & 5.1708 \\ 5.1708 & 4.3618 & 3.0528 & 3.0528 & 4.3618 \\ 0.9348 & 3.0528 & 6.4798 & 6.4798 & 3.0528 \\ 0.9348 & 3.0528 & 6.4798 & 6.4798 & 3.0528 \\ 5.1708 & 4.3618 & 3.0528 & 3.0528 & 4.3618 \end{pmatrix} = \sum_{j=1}^{2} \sigma_j U_{x_j} V_{x_j}^T,$

The initial factorization $A_2 = X_2 X_2^T$ is then given via the spectral decomposition of $A_2$ with

$X_2 = \begin{pmatrix} -2.0000 & 1.9465 \\ -2.0000 & 0.6015 \\ -2.0000 & -1.5747 \\ -2.0000 & -1.5747 \\ -2.0000 & 0.6015 \end{pmatrix}.$

Using an orthogonal matrix, we can easily transform the nonpositive columns into nonnegative columns. Thus, we use

$A_2 = \begin{pmatrix} 2.0000 & 1.9465 \\ 2.0000 & 0.6015 \\ 2.0000 & -1.5747 \\ 2.0000 & -1.5747 \\ 2.0000 & 0.6015 \end{pmatrix}$

as an initial factorization. Starting Algorithm 1 with this initial factorization, the very first starting
point returns the following decomposition in 2 seconds.

$$A_2 = \begin{pmatrix}
2.7897 & 0.0817 \\
1.8238 & 1.0177 \\
0.2609 & 2.5321 \\
0.2609 & 2.5321 \\
1.8238 & 1.0177
\end{pmatrix}^T$$

Here only one iteration was necessary. This decomposition is an exact cp-factorization to $A_2$ and therefore a SymNNMF of $A$ due to Lemma 3.3.

Figure 1: Success rate of Algorithms 1 and 2 for the matrix $A_{DS}$ with $k = 2$ for the same starting points.

A comparison of the performance of Algorithms 1 and 2 for the same 1000 starting points, again for the matrix $A_{DS}$ and $k = 2$, can be found in Figure 1.

To analyse the graphs in Figure 1 in more detail, we first focus on the blue dashed graph, representing the performance of Algorithm 1. Observe that any of the 1000 starting points returns a SymNNMF in less than 5 iterations. Considering the red graph, representing the performance of Algorithm 2, note that to exceed a success rate of 50%, it becomes necessary to allow more than 50 iterations. But after around 80 iterations every starting point returns a SymNNMF of $A_{DS}$. For the 1000 starting points, Algorithm 1 takes around 350 seconds in total, whereas Algorithm 2 takes only 1.3 seconds. This is due to the fact that it is not necessary to solve an expensive SOCP in every iteration step in Algorithm 2.

Increasing values of $k$ lead to a failure in both approaches. This is due to the fact that the approaches start struggling to show $A_k$ is completely positive for $k > 2$, cf. [18, Section 7.6].
Other test instances show a similar picture concerning the success rates of Algorithms 1 and 2 in comparison.

To show the influence of the parameter \( k \), we test both approaches on randomly generated matrices. To be more precise, we use the Matlab command \texttt{randn} to generate a random matrix \( D \in \mathbb{R}^{n \times n} \) for a given scalar value \( n \). Next, we compute \( \hat{D} \) by setting \( \hat{D}_{ij} := |D_{ij}| \) for all \( i, j \), and finally take \( A = \hat{D}\hat{D}^T \) as the matrix for which we want to find a SymNNMF. Then by construction \( A \in \mathcal{CP}_n \). We use the square root of \( A_k \) as an initial factorization for both Algorithms 1 and 2.

The performance of Algorithm 2 for \( n = 10 \) and several values of \( k \leq n \) can be found in Figure 2. Here we test the same randomly generated matrix \( A \in \mathbb{R}^{10 \times 10} \) and its best rank-\( k \) approximation \( A_k \) for every \( k \). More precisely, for every \( k \), we use 100 randomly chosen initial orthogonal matrices and test if the algorithm terminates successfully in less than 5000 iterations. As it turns out, it is possible to determine a SymNNMF of \( A \) for every value of \( k \). The closer \( k \) gets to the order \( n \) of \( A \), the lower the success rate.

Figure 2: Success rate of Algorithm 2 for random matrices \( A \) of order \( 10 \times 10 \) for several values of \( k \) and 100 randomly chosen starting points for each \( k \).

In the following experiment, we analyse the influence of the order \( n \) of the matrix \( A \) on the performance of Algorithm 2. We test the performance of the algorithm for randomly generated matrices \( A \in \mathbb{R}^{n \times n} \) for several values of \( n \). For every \( n \), we fix \( k = \lfloor 0.7n \rfloor \), where \( \lfloor \cdot \rfloor \) denotes the floor function. We initialize the algorithm with 100 randomly chosen initial starting points \( Q_0 \) and a maximum of 5000 iterations per starting point. The success rate of Algorithm 2 in this setting is illustrated in Figure 3. Here we see that for every \( n \), the algorithm terminates successfully for some starting points. For every \( n \leq 7 \), the algorithm returns a SymNNMF for any considered starting point.
Figure 3: Success rate of Algorithm 2 for random matrices of order $n \times n$ with $k = \lceil 0.7n \rceil$.

5 Numerical Results for Nonnegative Matrix Factorization

We now test Algorithm 3 under the same technical conditions as in Section 4. Numerically, the algorithm terminates successfully if it returns matrices $(X_kQ)$ and $(Q^{-1}Y_k)$ which are entrywise greater than or equal to $-10^{-12}$ and it terminates without success if these conditions are not fulfilled and the maximum number of iterations is reached.

As a first example, consider the following randomly generated matrix

$$A = \begin{pmatrix} 7 & 3 & 5 & 4 & 13 & 1 & 1 & 4 \\ 6 & 2 & 2 & 4 & 10 & 16 & 8 & 2 \\ 9 & 11 & 1 & 9 & 9 & 5 & 3 & 14 \\ 3 & 10 & 1 & 13 & 19 & 7 & 0 & 13 \\ 21 & 4 & 2 & 2 & 20 & 6 & 4 & 3 \end{pmatrix} \in \mathbb{R}^{5 \times 8} \quad (14)$$

of rank 5 and let $k = 3$. Then we compute the best rank-3 approximation $A_3$ to $A$:

Based on the technique introduced in \cite{1}, we get the initial factorization $A_3 = X_3Y_3$ with

$$A_3 = \begin{pmatrix}
-2.2164 & 0.3624 & -1.0752 \\
-2.4818 & 1.1425 & -3.2988 \\
-3.1257 & -1.7305 & 0.3226 \\
-3.7549 & -2.4724 & -0.8640 \\
-3.8469 & 2.8992 & 1.3067
\end{pmatrix}_{X_3 \in \mathbb{R}^{5 \times 3}},
$$

$$Y_3 = \begin{pmatrix}
-3.0116 & -1.9850 & -0.6127 & -2.0773 & -4.6122 & -2.1422 & -0.9359 & -2.3673 \\
2.4952 & -1.5193 & 0.3004 & -1.8944 & 0.6106 & 0.5318 & 0.8395 & -2.3186 \\
1.2916 & 0.3463 & 0.1195 & -0.3008 & 0.6402 & -3.0811 & -1.3905 & 0.3861
\end{pmatrix}_{Y_3 \in \mathbb{R}^{3 \times 8}},$$

where $\text{rank}(X_3) = \text{rank}(Y_3) = 3$ by definition. As an initial nonsingular matrix, we take

$$Q_0 = \begin{pmatrix} 0.4397 & 0.0464 & 0.2059 \\
0.3518 & 0.8796 & 0.0828 \\
0.2594 & 0.3400 & 0.4412 \end{pmatrix}.$$

Algorithm\cite{3} then takes 153 iterations and 0.0144 seconds to return the factorization

$$A_3 = \begin{pmatrix}
0.7170 & 0.0226 & 0.0047 \\
0.2196 & 0.3784 & 0.0020 \\
0.4051 & 0.1073 & 0.0128 \\
0.3566 & 0.2145 & 0.0165 \\
1.5761 & 0.4596 & 0.0014
\end{pmatrix}_{X_3 \in \mathbb{R}^{5 \times 3}},
$$

$$Y_3 = \begin{pmatrix}
12.2800 & 1.6756 & 1.9019 & 0.1499 & 10.9775 & 0.1221 & 1.0755 & 1.0565 \\
24.1982 & 0.2940 & 0.6981 & 1.8080 & 4.5600 & 11.2559 & 5.5133 & 0.0000 \\
21.5553 & 64.4149 & 46.0188 & 736.7893 & 675.0740 & 275.6546 & 0.0000 & 868.9005
\end{pmatrix}_{Y_3 \in \mathbb{R}^{3 \times 8}}.$$

Due to Theorem\cite{3.1} we have

$$\|A - A_3\|_F = \|A - \tilde{X}_3\tilde{Y}_3\|_F \leq \|A - B\|_F,$$

for every $B \in \mathbb{R}^{n \times m}$ with $\text{rank}(B) \leq 3$. Combined with the fact that $\tilde{X}_3$ and $\tilde{Y}_3$ are entrywise nonnegative, we get that $\tilde{X}_3\tilde{Y}_3$ is the desired NNMF of $A$ of rank 3.

Moreover, in the following experiment, we will have a closer look at the influence of the parameter $k$ for a given matrix $A \in \mathbb{R}^{m \times n}$.

To this end, we consider the randomly generated matrix

$$A = \begin{pmatrix}
16 & 40 & 29 & 9 & 42 & 36 & 24 & 26 \\
19 & 41 & 30 & 11 & 26 & 31 & 22 & 30 \\
24 & 34 & 50 & 36 & 25 & 42 & 41 & 48 \\
13 & 24 & 26 & 25 & 16 & 34 & 28 & 35 \\
9 & 39 & 29 & 18 & 19 & 39 & 19 & 38
\end{pmatrix}_{\in \mathbb{R}^{5 \times 8}}.$$

For the experiment, we test the performance of 100 starting points $Q_0$ with a maximum of 3000 iterations per starting point in Algorithm\cite{3} for different values of $k$. The results are collected in Figure\cite{4}.

Figure\cite{4} indicates that the performance of Algorithm\cite{3} depends on the choice of $k$. Whereas for $k = 2$ nearly every starting point returns a NNMF, the success rate decreases for higher values of $k$. Taking $k = 4$ still returns a success rate of more than 90%. For $k = 5 = \text{rank}(A)$, it is still possible to derive a NNMF of $A$. This therefore shows that it is not necessary to add a low-rank constraint to obtain a NNMF. Figure\cite{4} therefore substantiates the good performance of Algorithm\cite{3} for different choices of $k$ for the concrete test instance.
Figure 4: Success rate of Algorithm 3 for a given matrix of order $5 \times 8$ and different values of $k$.

Figure 5: Success rate of Algorithm 3 for matrices of order $n \times m$ and $k = \lfloor 0.7 \cdot \min\{n, m\} \rfloor$.

In the next experiment will analyse the influence of the order $n \times m$ of the given matrix $A$ on the performance of Algorithm 3. To this end, we consider randomly generated matrices $A \in \mathbb{R}^{n \times m}$ for different values of $n$ and $m$. The instances are generated as follows: Given the values $n, m,$
we define \( l := \min\{n, m\} \) and with this, we construct matrices \( X \in \mathbb{R}^{n \times l} \) and \( Y \in \mathbb{R}^{l \times m} \) using the Matlab command \texttt{randn}. Then we construct \( |X| \) and \( |Y| \) by taking the absolute values of the entries. Finally, we define \( A := |X| \cdot |Y| \).

For each such generated matrix, we set \( k = \lfloor 0.7l \rfloor \) and analyse the success rate of 100 randomly generated initial nonsingular matrices \( Q_0 \). For every \( Q_0 \), we allow at most 3000 iterations. The performance of Algorithm 3 in this setting is illustrated in Figure 5.

Note that Algorithm 3 terminates successfully in every case, such that the success in total is independent of \( n \) and \( m \) and it does not make a difference whether \( m > n \) or \( m < n \). Especially for small dimensions like \( n = 4 \), \( m = 5 \) and \( n = 5 \), \( m = 4 \), plotted as the solid blue and the dashed red line, it turns out that the algorithm terminates successfully for around 90% of the initial nonsingular matrices. Furthermore, if the algorithm terminates successfully for one of the initial nonsingular matrices, a NNMF is provided in less than 100 iterations. Increasing values of \( m \) and \( n \) do not seem to influence the success rate in less than 3000 iterations, but it takes more iterations on average to return a NNMF. Here especially the green line, representing \( n = 8 \), \( m = 10 \), illustrates this behaviour. Altogether, Figure 5 shows that the introduced method to derive NNMF works well for matrices \( A \), which are randomly generated.

6 Conclusion

In this paper, we introduced a new method to derive symmetric nonnegative matrix factorizations which is based on computing completely positive factorizations. Here especially the method of alternating projections was the key tool and a local convergence result for this first approach was given. Nevertheless, since computing the projection onto one of the sets amounts solving an SOCP, we introduced an alternative approach where we loose the local convergence theory but the approach avoids this computational effort and numerical experiments indicate that it still works well. Even for general rectangular input matrices, we used the alternating projection method and extended the approach to this more general framework. As it turned out, it was necessary to replace orthogonal matrices by a more general matrix class to generate a new fundamental lemma in this context. Based on this lemma, we developed a method to obtain general nonnegative matrix factorizations. Unfortunately, since one of the subsets is not closed, we could not apply straightforward alternating projections. Instead, we generalised the alternative approach to avoid SOCP in the symmetric case to the general rectangular case which yielded an applicable heursitic approach for nonnegative matrix factorization. As numerical experiments substantiate, this algorithm still works well and hence gives a new approach to derive nonnegative matrix factorizations.

Acknowledgements

I wish to thank Mirjam Dür for her highly useful comments and recommendations.
References


