Convex analysis of Nonnegative Matrix Factorization

Linjian Ma, William Krinsman, Aviral Kumar

Abstract— **This report focuses on the literature review of convex analysis of non-negative matrix factorization (NMF). NMF is an ill-posed problem with non-unique solutions and is a NP-hard problem and is difficult to solve. NMF has wide applicability in the domains of Information Retrieval, Spectral Clustering, Data Compression, and many other domains, thereby making the problem important. The existence of many sub-optimal solutions makes it important to design algorithms/techniques so that good solutions can be found.**

We talk about some variations of NMF, for example, Convex-NMF, Semi-NMF, that are solvable using iterative techniques and have interesting properties, and how they can be applied to some problems with missing data. We also talk about how NMF can be cast as a conic program over the cone of completely-positive(CP) matrices. This relation helps us prove the existence of a non-trivial NMF for every non-negative matrix. We also summarize different ways in which NMF can be relaxed to a convex program. We also briefly summarize how matrix under-approximation constraints can be used to give sparser factorizations than what the general NMF delivers which is practically useful in many cases.

List of papers we summarize in this report:

- *Introduction to nonnegative matrix factorization*, Nicholas Gillis, 2017
- *Convex and semi-nonnegative matrix factorizations*, Ding, Chris HQ and Li, Tao and Jordan, Michael I, 2010
- *Convex nonnegative matrix factorization with missing data*, Hamon, Ronan and Emiya, Valentin and Févotte, Cédric, 2016
- *Convex Algorithms for Nonnegative Matrix Factorization*, Krishnamurthy, V. and d'Aspremont, A., 2012
- *Using underapproximations for sparse nonnegative matrix factorization*, Gillis, Nicolas and Glineur, François, 2010
- *Non-Negative Matrix Factorization, Convexity and Isometry*, Nikolaos Vasiloglou, Alexander G. Gray, David V. Anderson, 2008

Overview

In this literature review we study the problem of Non-negative Matrix Factorization. As mentioned in the abstract, NMF is a widely applicable problem in the domains of Information Retrieval, Generative Modelling, Image recognition, Spectral Clustering, etc.

There are multiple formulations of the NMF problem for a matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ and in the most general form, the problem is cast as follows:

$$\min_{\substack{\mathbf{X} \in \mathbb{R}^{n \times k}, \\ \mathbf{Y} \in \mathbb{R}^{k \times m}}} ||\mathbf{A} - \mathbf{X}\mathbf{Y}^\top||_* \quad \text{s.t.} \quad \mathbf{X} \geq 0, \quad \mathbf{Y} \geq 0$$

where, the minimization problem involves some matrix norm (denoted by $*$), and the $\geq$ constraints indicate element-wise non-negativity. In general, this problem is non-convex, however, as we will discuss, a number of approximations can be made to this optimization problem.

Applications of NMF include the following:

- Graph clustering in general graphs into a fixed number of clusters
- Feature generation for co-occurrence matrix in word embedding
- Semi-supervised or distantly-supervised classification
- Recommendation systems for generating features of users and commodities.

In this report, we study this problem of NMF. In general, NMF has non-unique solutions and the solutions are hard-to-find. The quest is to develop practically applicable algorithms in the gradient descent framework, which can reach local solutions of NMF provably or either in a PAC sense.

As an overview of this report, we first describe the NMF problem in detail and rigorously in Chapter I, where we start off by describing how NMF belongs to the family of constrained-low rank approximations of a matrix and how it is related to a very other well known algorithm of the same class – PCA. This is mainly from [6].

We then go ahead to describe how we can add constraints to the problem and talk about Convex-NMF and Semi-NMF which can be related to K-means clustering. This also gives us a method to derive some practical algorithms on the lines of algorithms used for K-means clustering in the ALS framework. This part is mainly derived from [2].

We also then describe an application of this version of convex NMF to the problem of missing data [8], which was implemented as an optimization problem. In this chapter, we also describe one experiment where we used Kuls image dataset and compared convex-NMF, Semi-NMF to K-means clustering.

In Chapter III, we describe the geometric interpretation of the NMF problem and how this geometric interpretation can be used to show NP-hardness. This is mainly derived from [6].

In Chapter IV, we describe how the NMF problem links with the problem of determining whether a matrix is Completely-Positive(CP) or not. This analogy gives us a solution to the problem that is not practically realizable. Then we discuss several relaxations to this approach which boil down to convex SDPs and discuss the properties of these optimization problems.

We also apply one of these methods to a graph partitioning problem and reproduce some results from the paper. This Chapter is mainly derived from [10] and [11].

In the last chapter, Chapter V, we talk about Non-negative Matrix Underapproximations, where we talk about how adding an underapproximation constraint in the NMF optimization problem can induce sparsity in the factorized solutions and how this can be achieved in practice. This section is mainly derived from [7].

We then conclude by discussing the main avenues for future work including related problems like matrix completion.

CONTENTS

I Introduction to Non-Negative Matrix Factorization 2
- I-A Constrained Low-Rank Matrix Approximations 2
- I-B Specializations of the problem: NMF and other loss functions . . 3

II Convex Nonnegative Matrix Factorizations and its applications 3
- II-A Convex and Semi-Nonnegative Matrix Factorizations 4
- II-B Algorithms for Semi-NMF and Convex-NMF 4
- II-C Example: Formulation of the Convex-NMF with missing data problem 5
- II-D Implementation of CNMF with missing data as an optimization problem 6
- II-E Experimental results 6

III Geometric interpretation of NMF and NP-hardness 7
- III-A NMF is NP-hard 7
- III-B Relationship to Intermediate Simplex Problem 8

IV Convex Relaxations of Nonnegative Matrix Factorization 9
- IV-A Review of some properties 9
- IV-B Existence of NMF factorization . . 10
- IV-C Solving NMF optimization problems 10
  - IV-C.1 NMF as a convex conic program 10
- IV-D Convex relaxations of the NMF problem 10
- IV-E Isometric NMF 12
  - IV-E.1 Convex Maximum Furthest Neighbour Unfolding (MFNU) . . 12
  - IV-E.2 Non-convex MFNU . . 12
  - IV-E.3 Convex IsoNMF 12
  - IV-E.4 Non-Convex IsoNMF . 12
- IV-F Some more (global) convex approximations to NMF 12
- IV-G Global Approximations for NMF with synmmetric data matrices . . 12
  - IV-G.1 Factorizing $\exp_H(\mathbf{X})$ when $\mathbf{X}$ is PSD 13
  - IV-G.2 Sparse Decomposition . 13
  - IV-G.3 Recursive decomposition 14
- IV-H Global approximations to NMF with Non-Symmetric Data Matrices 14
  - IV-H.1 Algorithms 14
- IV-I Experiments 14

V Nonnegative Matrix Underapproximation 15
- V-A Sparsity of NMU 15
- V-B Algorithm for NMU based on Lagrangian relaxation 16
- V-C Experimental results 17

References 17

I. INTRODUCTION TO NON-NEGATIVE MATRIX FACTORIZATION

A. *Constrained Low-Rank Matrix Approximations*

Here we discuss the general framework of constrained low-rank matrix approximations, and explain how they can be viewed properly as a ”data-reduction” problem.

The author in [6] places the problem of non-negative matrix factorization (NMF) into the context of the broader problem of constrained low-rank matrix approximations, with NMF being interpreted as a specific example. Any low-rank matrix factorization can be considered a data reduction problem in the following manner. Assuming that

$$\mathbf{M} = [m_1 \dots m_n] \approx [u_1 \dots u_r]\mathbf{V} = \mathbf{UV}$$

where each of the m_j, $j = 1, \dots, n$ represent one of our n data points, the above matrix factorization gives us that each data point $m_{j'}$ can be written as:

$$m_{j'} = \sum_{j=1}^{n} m_{jj'} e_j \approx \sum_{j=1} \left(\sum_{k=1}^{r} u_{jk} v_{kj'} \right) e_j = \sum_{k=1}^{r} \sum_{j=1}^{n} u_{jk} v_{kj'} e_j = \sum_{k=1}^{r} v_{kj'} \sum_{j=1}^{n} u_{jk} e_j = \sum_{k=1}^{r} v_{kj'} u_k .$$

In other words, the matrix factorization guarantees us that every one of our n data points is approximately equal to the linear combination of at most r "data points", which is useful when $r \ll n$. Nevertheless, this decomposition may not say anything meaningful if either the values of the weights $v_{kj'}$ appearing in the decomposition

$$m_{j'} \approx \sum_{k=1}^{r} v_{kj'} u_k \ ,$$

or the u_k's themselves, prevent the interpretation of the data points being equal to linear combinations of unobserved data, due to either the weights or the u_k's not corresponding to data values which are actually realizable.

B. Specializations of the problem: NMF and other loss functions

Here we use the framework for constrained low-rank matrix approximations established in the previous section to describe the NMF problem as well as its relationship to similar problems.

The establishment of this framework leads the author to conclude that there are two main choices of directions for specializing the general low-rank matrix approximation problem: (1) the choice of the loss criterion for measuring the discrepancy between the data matrix $\mathbf{M}$ and the factorization $\mathbf{UV}$, and (2) the choice of constraints (if any) to place on $\mathbf{U}$ and $\mathbf{V}$. Then non-negative matrix factorization corresponds to (2), where we constrain the entries of both $\mathbf{U}$ and $\mathbf{V}$ to be non-negative. This means that all of the entries of $\mathbf{M}$ must themselves be non-negative, and corresponds to writing each data point, $m_{j'}$, all of whose entries are known to be non-negative, as a conic combination (with the entries of $\mathbf{V}$ being the weights) of vectors which also have all non-negative entries. Gillis does not consider variations of NMF here, although any such variations (such as those considered in [2]) are easy to understand in this framework. The author does, however, mention other constrained low-rank matrix approximation schemes, e.g.

- PCA: Frobenius norm or spectral norm loss criterion (Eckart-Young theorem), no constraints
- "robust PCA": entrywise absolute value loss criterion, no constraints
- missing data: one's favorite choice of loss criterion along with a Hadamard product with a binary weight matrix $\mathbf{W}$ (also called a *masking matrix*), whose entries are 0 for missing entries of $\mathbf{M}$ and 1 otherwise (this is essentially the approach taken in [8]). This is a special case of what [6] calls *weighted low-rank matrix approximation* (WLRA).
- K-means clustering: the columns of $\mathbf{V}$ are restricted to have only one non-zero entry (1) each, making them cluster indicators – the data point m_n is approximated in the factorization by the column of u corresponding to the nonzero entry of the column v_n.

The author writes this framework mathematically in the following way: constrained low-rank matrix approximation is the same as the problem

$$\min_{\substack{\mathbf{U} \in \Omega_U, \\ \mathbf{V} \in \Omega_V}} ||\mathbf{M} - \mathbf{UV}|| \ ,$$

where both (1) the loss criterion $|| \cdot ||$ and (2) the spaces Ω_U and Ω_V need to be chosen. To reiterate, for NMF the choice of $|| \cdot ||$ is (essentially) arbitrary, but both Ω_U and Ω_V are chosen to be spaces of matrices whose entries are all non-negative.

The author considers applications of NMF to hyperspectral imaging, the existence of more efficient algorithms when somewhat restrictive assumptions on $\mathbf{M}$ are made, and results about the non-negative rank of matrices (the smallest r such that an *exact* NMF with $\mathbf{U} \in \mathbb{R}^{n \times r}, \mathbf{V} \in \mathbb{R}^{r \times p}$ exists). A general critique of the most common convexification approaches for these classes of problems is also given, which seems to presage the conclusion later in the paper that "there does not exist, to the best of our knowledge, a successful convexification approach for NMF, as opposed to other low-rank models". Therefore the author focuses on two-block coordinate descent (alternating between fixing $\mathbf{U}$ then minimizing the loss criterion with respect to $\mathbf{V}$, then fixing $\mathbf{V}$ and minimizing the loss criterion with respect to $\mathbf{U}$, and so on) as the most useful approach for find approximate NMF's in practice. This corresponds to the focus on block coordinate descent algorithms found in both [2] and [8], and seemingly overlooks the approaches found in [10].

II. Convex Nonnegative Matrix Factorizations and its applications

In this section we will review some literature that link variants of NMF, which are Convex-NMF and Semi-NMF, to K-means clustering. Methods described in the previous chapter try to solve the non-convex formulations of NMF using gradient descent techniques hoping for the best. But such solutions might often end up at local minima. Instead, we could relax the problem to make it convex or quasi-convex and then try to reach the global optima of the relaxed problem. Usually, the latter has been found to be a better approach and we talk about these methods from now on in this report.

A. Convex and Semi-Nonnegative Matrix Factorizations

Reference [2] shows the link between Convex-NMF, Semi-NMF and statistical data analysis. The authors develop NMF like algorithms that yields nonnegative factor matrices while the input data is not necessary nonnegative. In addition, the authors link these variations to K-means clustering and show that they are the relaxations of K-means. The definition for Semi-NMF and Convex-NMF are as follows:

Definition II.1. *Given* $\mathbf{M} \in \mathbb{R}^{m\times n}$ *and* $1 \le k < \min(m,n)$*, the* ***Semi Nonnegative Matrix Factorization*** *(Semi-NMF) problem is defined as:*

$$\min_{\substack{\mathbf{V}\in\mathbb{R}^{m\times k},\\ \mathbf{W}\in\mathbb{R}^{k\times n}}} ||\mathbf{M}-\mathbf{V}\mathbf{W}||_F \text{ such that } \mathbf{W} \ge 0.$$

Definition II.2. *Given* $\mathbf{M} \in \mathbb{R}^{m\times n}$ *and* $1 \le k < \min(m,n)$*, the* ***Convex Nonnegative Matrix Factorization*** *(Convex-NMF) problem is defined as:*

$$\min_{\substack{\mathbf{V}\in\mathbb{R}^{n\times k},\\ \mathbf{W}\in\mathbb{R}^{k\times n}}} ||\mathbf{M}-\mathbf{M}\mathbf{V}\mathbf{W}||_F \text{ such that } \mathbf{V} \ge 0, \mathbf{W} \ge 0.$$

The motivations of both these NMF variants are from clustering. Suppose we do a K-means clustering on $\mathbf{M}$ and obtain cluster centroids $\mathbf{V} = [v_1, \dots, v_k]$. Let $\mathbf{W}^\top$ denote the cluster indicators, i.e., $w_{ki} = 1$ if w_i belongs to kth cluster, $w_{ki} = 0$ otherwise. We can write the K-means clustering objective function as

$$J_{\text{K-means}} = \|\mathbf{M}-\mathbf{V}\mathbf{W}\|_F, \quad \text{s.t. } w_{ij} \in \{0,1\}.$$

This is a NP-Hard problem and considering the constraints on $\mathbf{W}$. The authors also proved that when $\mathbf{W}$ is orthogonal, $\mathbf{W}\mathbf{W}^\top = \mathbf{I}$, both Semi-NMF and Convex-NMF are the relaxations of K-means. The brief argument is as follow:

The objective of the K-means cluster is

$$J = \|\mathbf{M}-\mathbf{V}\mathbf{W}\|_F = \text{Tr}(\mathbf{M}^\top\mathbf{M} - 2\mathbf{M}^\top\mathbf{V}\mathbf{W} + \mathbf{V}\mathbf{V}^\top)\,.$$

The first order optimality condition implies that $\mathbf{V} = \mathbf{M}\mathbf{W}^\top$. Thus, we obtain

$$J = \text{Tr}(\mathbf{M}^\top\mathbf{M} - \mathbf{W}\mathbf{M}^\top\mathbf{M}\mathbf{W}^\top)\,.$$

Considering that the first term is constant and doesn't affect the results, one can see that the minimization problem is equivalent to

$$\max_{\mathbf{W}\mathbf{W}^\top=\mathbf{I}} \text{Tr}(\mathbf{W}\mathbf{K}\mathbf{W}^\top)\,,$$

where $\mathbf{K}$ is a linear kernel $\mathbf{M}^\top\mathbf{M}$. It is known that this is identical to K-means clustering under the correct assumptions. When $\mathbf{W}$ is not restricted to be orthogonal, these NMF variants are called soft versions of K-means clustering.

B. Algorithms for Semi-NMF and Convex-NMF

The authors compute the Semi-NMF factorization via an iterative updating algorithm that alternatively updates $\mathbf{V}$ and $\mathbf{W}$. $\mathbf{V}$ is updated based on directly solving the lease squares problem:

$$\mathbf{V} = \mathbf{M}\mathbf{W}^\top(\mathbf{W}\mathbf{W}^\top)^{-1}.$$

Note $\mathbf{W}\mathbf{W}^\top$ is a $k \times k$ positive semidefinite matrix. In most cases, $\mathbf{W}\mathbf{W}^\top$ is nonsingular, and pseudoinverse will be taken when it is singular. After $\mathbf{V}$ is updated, $\mathbf{W}$ will be updated using

$$[\mathbf{W}]_{ki} \leftarrow [\mathbf{W}]_{ki}\sqrt{\frac{(\mathbf{M}^\top\mathbf{V})^+_{ik} + [\mathbf{W}^\top(\mathbf{V}^\top\mathbf{V})^-]_{ik}}{(\mathbf{M}^\top\mathbf{V})^-_{ik} + [\mathbf{W}^\top(\mathbf{V}^\top\mathbf{V})^+]_{ik}}}.$$

It is trivial to show that fixing $\mathbf{W}$, the update rule for $\mathbf{V}$ gives the optimal solution to the objective. The authors also show that the limiting solution of the update rule for $\mathbf{W}$ satisfies the KKT condition. Consider the Lagrangian function for Semi-NMF:

$$\mathcal{L}(\mathbf{W},\beta) = \text{Tr}(-2\mathbf{M}^\top\mathbf{V}\mathbf{W} + \mathbf{W}^\top\mathbf{V}^\top\mathbf{V}\mathbf{W} - \beta\mathbf{W})\,.$$

The first KKT optimality condition gives us then that:

$$-2\mathbf{M}^\top\mathbf{V} + 2\mathbf{W}^\top\mathbf{V}^\top\mathbf{V} - \beta = 0\,.$$

From the complementary slackness condition, we obtain that

$$(-2\mathbf{M}^\top\mathbf{V} + 2\mathbf{W}^\top\mathbf{V}^\top\mathbf{V})_{ik}\mathbf{W}_{ki} = \beta_{ik}\mathbf{W}_{ki} = 0.$$

At the convergence point of the iteration,

$$\sqrt{\frac{(\mathbf{M}^\top\mathbf{V})^+_{ik} + [\mathbf{W}^\top(\mathbf{V}^\top\mathbf{V})^-]_{ik}}{(\mathbf{M}^\top\mathbf{V})^-_{ik} + [\mathbf{W}^\top(\mathbf{V}^\top\mathbf{V})^+]_{ik}}} = 1,$$

which is the same as the condition obtained from the complementary slackness condition based on the fact that

$$\mathbf{A} = \mathbf{A}^+ - \mathbf{A}^-.$$

It is also shown in the reference [2] that updating $\mathbf{W}$ will lead to the monotonic decrease of the objective, proving the stability of the algorithm.

The iterative procedure for Convex-NMF is similar to that for Semi-NMF; the update rules are as follows:

$$[\mathbf{W}]_{ki} \leftarrow [\mathbf{W}]_{ki}\sqrt{\frac{[(\mathbf{M}^\top\mathbf{M})^+\mathbf{V}]^+_{ik} + [\mathbf{W}^\top\mathbf{V}^\top(\mathbf{M}^\top\mathbf{M})^-\mathbf{V}]_{ik}}{[(\mathbf{M}^\top\mathbf{M})^+\mathbf{V}]^-_{ik} + [\mathbf{W}^\top\mathbf{V}^\top(\mathbf{M}^\top\mathbf{M})^+\mathbf{V}]_{ik}}},$$

$$\mathbf{V}_{ik} \leftarrow \mathbf{V}_{ik}\sqrt{\frac{[(\mathbf{M}^\top\mathbf{M})^+\mathbf{W}^\top]^+_{ik} + [(\mathbf{M}^\top\mathbf{M})^-\mathbf{V}\mathbf{W}\mathbf{W}^\top]_{ik}}{[(\mathbf{M}^\top\mathbf{M})^+\mathbf{W}^\top]^-_{ik} + [(\mathbf{M}^\top\mathbf{M})^+\mathbf{V}\mathbf{W}\mathbf{W}^\top]_{ik}}}.$$

It can be shown based on the similar analysis that the limiting solutions of the update rule satisfy the KKT condition.

C. Example: Formulation of the Convex-NMF with missing data problem

We now take the case of a specific example where we don't know in advance some part of the data, and we want to factorize the data matrix even in this scenario. The authors in this paper [8] begin by defining non-negative NMF and stating the other relevant definitions. They denote a NMF by

$$\mathbf{V} = \mathbf{W}\mathbf{H}, \quad \mathbf{V} \in \mathbb{R}_{+}^{F\times N}, \mathbf{W} \in \mathbb{R}_{+}^{F\times K}, \mathbf{H} \in \mathbb{R}_{+}^{K\times N},$$

thus, comparing with the notation of [6], one has that $\mathbf{V} = \mathbf{M}$, $\mathbf{W} = \mathbf{U}$, $\mathbf{V} = \mathbf{H}$, $F = p$, $N = n$, and $K = r$, and comparing with the notation of [2], one has that $\mathbf{V} = \mathbf{X}$, $\mathbf{F} = \mathbf{W}$, $\mathbf{G}^{\top} = \mathbf{H}$, $F = p$, $N = n$, and $K = k$. They call the columns of $\mathbf{W}$ *components* or *patterns*, and $\mathbf{V}$ the *data matrix*. Using this definition of NMF, they go on to define convex NMF as the following factorization:

$$\mathbf{V} = \mathbf{S}\mathbf{L}\mathbf{H} \ ,$$
$$\mathbf{S} = [s_1, \dots, s_P] \in \mathbb{R}_{+}^{F\times P}, \quad \mathbf{L} = [l_1, \dots, l_K] \in \mathbb{R}^{P\times K}$$

i.e. $\mathbf{W} = \mathbf{S}\mathbf{L}$, now calling $\mathbf{W}$ the *dictionary matrix*. The columns $s_1, \dots, s_P$ are called *atoms*, and $\mathbf{L}$ is called the *labeling* matrix. The *convex* in convex NMF comes from the following additional assumption: for each $k = 1, \dots, K$, we impose the constraint

$$||l_k|| = \sum_{p=1}^{P} l_{pk} = 1 \ .$$

This means that, since the columns w_k of $\mathbf{W}$ can be written, for each $k = 1, \dots, K$:

$$w_k = \sum_{p=1}^{P} l_{pk} s_f \ ,$$

each of the columns w_k of the dictionary matrix $\mathbf{W}$ is assumed to be a *convex* combination of the atoms s_p, the columns of the matrix S.

It is claimed that usually $P \gg K$. In other words, the possible number of atoms summed together to create each dictionary element is usually much larger than the rank of the corresponding NMF. The authors interpret choosing specific values for the atoms s_p as a form of supervised learning, and state that the most extreme case of this supervision is when the matrix of atoms is assumed to be known to be the data matrix $\mathbf{V}$ itself, which they point out is exactly the setting considered in [2]. Therefore the notion of convex NMF studied in this paper is strictly more general than that considered in [2]. Note that in the case where $\mathbf{V} = \mathbf{S}$, i.e. the data is "auto-encoded", we have $P = N$, and therefore the condition that $P \gg K$ is the same as $N \gg K$, which corresponds to the intuition that the factorization should be "low-rank".

The authors then generalize this definition further to get a notion of convex NMF with missing data. Note that the means they use to do so is a special case of the weighted low-rank approximation mentioned in [6]. Specifically, they define the so-called *masking matrices* $\mathbf{M}_S$ and $\mathbf{M}_V$ as follows:

Definition II.3. *The* ***masking matrix for the atom matrix*** $\mathbf{S}$ *is:*

$$\mathbf{M}_S \in \mathbb{R}^{F\times P}, [\mathbf{M}_S]_{fp} := \begin{cases} 1 & \textit{if } s_{fp} \textit{ is known} \\ 0 & \textit{else} \end{cases} ,$$

while the ***masking matrix for the data matrix*** $\mathbf{V}$ *is:*

$$\mathbf{M}_V \in \mathbb{R}^{F\times N}, [\mathbf{M}_V]_{fn} := \begin{cases} 1 & \textit{if } v_{fn} \textit{ is known} \\ 0 & \textit{otherwise} \end{cases} .$$

By taking the Hadamard (entrywise) product of $\mathbf{S}$ with $\mathbf{M}_S$, we effectively "*mask*" the unknown entries of $\mathbf{S}$, likewise when we take the Hadamard product of $\mathbf{V}$ with $\mathbf{M}_V$.

Definition II.4. *The* ***Hadamard product*** *of two matrices* $\mathbf{A}$*,* $\mathbf{B}$ *is denoted* $\mathbf{A} \circ \mathbf{B}$ *and defined as:*

$$[\mathbf{A} \circ \mathbf{B}]_{ij} = [\mathbf{A}]_{ij}[\mathbf{B}]_{ij}$$

Of course, the two masking matrices $\mathbf{M}_S$ and $\mathbf{M}_V$ coincide in the case where $\mathbf{S} = \mathbf{V}$. Throughout, we assume that $\mathbf{S}^O := \mathbf{M}_S \circ \mathbf{S}$ is specified by the user (O for "observed") and therefore known.

The goal is then, given the partially observed data matrix $\mathbf{M}_V \circ \mathbf{V}$ and $\mathbf{S}^O$, to estimate $\mathbf{L}$, $\mathbf{H}$, and the missing entries of $\mathbf{S}$. (Note that the missingness of values in $\mathbf{S}$ can be independent of the missingness of values in $\mathbf{V}$.) Thus, we can write the problem of convex NMF with missing data as:

Definition II.5. ***Convex NMF with missing data*** *is the problem of minimizing the objective*

$$\mathscr{L}_{\mathbf{V},\mathbf{M}_V}(\mathbf{S}, \mathbf{L}, \mathbf{H})$$

for a given choice of loss function $\mathscr{L}_{\mathbf{V},\mathbf{M}_V}$ *which has the data matrix* $\mathbf{V}$ *and the data masking matrix* $\mathbf{M}_V$ *as two of its parameters, subject to the constraints that:*

- $\mathbf{S} \in \mathbb{R}_{+}^{F\times P}$ *is non-negative,*
- $\mathbf{L} \in \mathbb{R}_{+}^{P\times K}$ *is non-negative,*
- $\mathbf{H} \in \mathbb{R}_{+}^{K\times N}$ *is non-negative,*
- $\mathbf{S}$ *is equal to the known values* $\mathbf{S}^O$ *when "masked", i.e.* $\mathbf{M}_S \circ \mathbf{S} = \mathbf{S}^O$*,*
- *for each* $k = 1, \dots, K$*, the corresponding column of* $\mathbf{L}$ *satisfies*

$$||l_k|| = \sum_{p=1}^{P} l_{pk} = 1 \ .$$

Thus the missingness of values in $\mathbf{S}$ enters into the CNMF with missing data problem through the constraints, whereas the missingness of values in $\mathbf{V}$ enters the CNMF with missing data problem through the choice of objective. Ideally our objective should not depend on any values of $\mathbf{V}$ corresponding to the zero entries of $\mathbf{M}_V$, which can be accomplished using the WLRA formalism previously referred to from [6], the masking matrix $\mathbf{M}_V$ being the weight matrix for our objective.

D. Implementation of CNMF with missing data as an optimization problem

For their choice of loss function, the authors use entrywise β-divergence, or more specifically:

$$D_\beta(\mathbf{M}_V \circ \mathbf{V} | \mathbf{M}_V \circ \mathbf{SLH}) := \sum_{f=1}^{F} \sum_{n=1}^{N} d_\beta([\mathbf{M}_V \circ \mathbf{V}]_{fn} | [\mathbf{M}_V \circ \mathbf{SLH}]_{fn}) \ .$$

The β-divergence has the following definition:

Definition II.6. *Given two positive numbers x and y, the **β-divergence** between them is:*

$$d_\beta(x|y) := \begin{cases} x \log \frac{x}{y} - x + y & \beta = 1 \\ \frac{x}{y} - \log \frac{x}{y} - 1 & \beta = 0 \\ \frac{1}{\beta(\beta-1)}(x^\beta + (\beta-1)y^\beta - \beta x y^{\beta-1}) & \textit{else} \end{cases}$$

The β-divergence can be motivated by observations from information geometry, but no motivation for this choice is provided in [8] itself. The β-divergence is also popular in general for applications of NMF to music signal processing, which corresponds to previous research by the authors of [8].

However, as noted in [4]:

Theorem II.7. *The β-divergence is equal to:*

- *Euclidean distance when $\beta = 2$,*
- *KL divergence when $\beta = 1$,*
- *Itakura-Saito divergence when $\beta = 0$.*

Therefore the β-divergence can be used to interpolate between several different popular kinds of divergence, making results established using a general β-divergence more easily generalizable to a wide range of application areas.

Another reason to consider the β-divergence [4] is that the parameter β itself can be cross-validated, giving us another way (potentially) to improve the performance of our models. Moreover, different values of β correspond to interpolations between "Gaussian additive, Poisson, and multiplicative Gamma observation noise" [4].

A major advantage of the criterion given above is its separability in the entries of the matrix $\mathbf{V}$, since it allows us to decompose the optimization problem into smaller optimization problems for each of the entries of $\mathbf{M}_V \circ \mathbf{V}$. Note also that the above criterion is a generalization of the usual Frobenius norm squared criterion, since that corresponds to the special case where $\beta = 2$. Therefore we are in a strictly more general situation than in other papers where only the squared Frobenius norm is considered.

The authors propose a three-block coordinate descent algorithm for obtaining a local minimum solution (with each block corresponding to one of the three unknown matrices $\mathbf{S}$, $\mathbf{L}$, and $\mathbf{H}$), similar to the approaches used in [2], except that the algorithms in that paper were only two-block coordinate descent. (However, the authors here refer to the approach as majorization-minimization, rather than block coordinate descent.) More specifically, just like in [2], they derive what are called auxiliary functions for each of the updates, and then show correctness and convergence of the algorithm using convex analysis of the corresponding auxiliary functions.

Definition II.8. *An **auxiliary** function $\mathbb{R}_+^{I \times J} \times \mathbb{R}_+^{I \times J} \to \mathbb{R}$ of an (objective) function $\mathbb{R}_+^{I \times J} \to \mathbb{R}$ C is a function G such that*

$$C(\mathbf{A}) = G(\mathbf{A}, \mathbf{A}) \quad C(\mathbf{A}) \le G(\mathbf{A}, \mathbf{A}')$$

for all $\mathbf{A}, \mathbf{A}' \in \mathbb{R}_+^{I \times J}$.

Decomposing the β-divergence into a convex part (which can be lower bounded using Jensen's inequality, and whose minimum can be found from its Hessian), a concave part (which can be bounded using the fact that, for a differentiable concave function C, $C(\tilde{s}) + \langle \nabla C(s), s - \tilde{s} \rangle \le C(s)$ for any $s, \tilde{s}$ in the domain of C), and a constant part (which is irrelevant for the optimization problem), the authors are able to show the correctness of each of the proposed update steps for their alternating minimization algorithm. Unlike the authors of [2], the authors of [8] do not *directly* appeal to KKT conditions to argue for the correctness of their update steps.

E. Experimental results

To show that Convex-NMF can produce similar results as K-means, we performed experiments on the synthetic data, which is shown in Figure 1. We let the input data to be 4 blobs with different distribution and sizes, and test the effect of both Convex-NMF and Semi-NMF. It can be seen from the figure that the clustering centroids predicted through Semi-NMF is pretty far off the true centroids on all these 4 input data, while Convex-NMF performed pretty well on the evenly sized blobs, anisotropicly distributed blobs and the blobs with unequal variance. For the input with unevenly sized blobs, the convex-NMF cannot tract the position of the blob with too less data points.

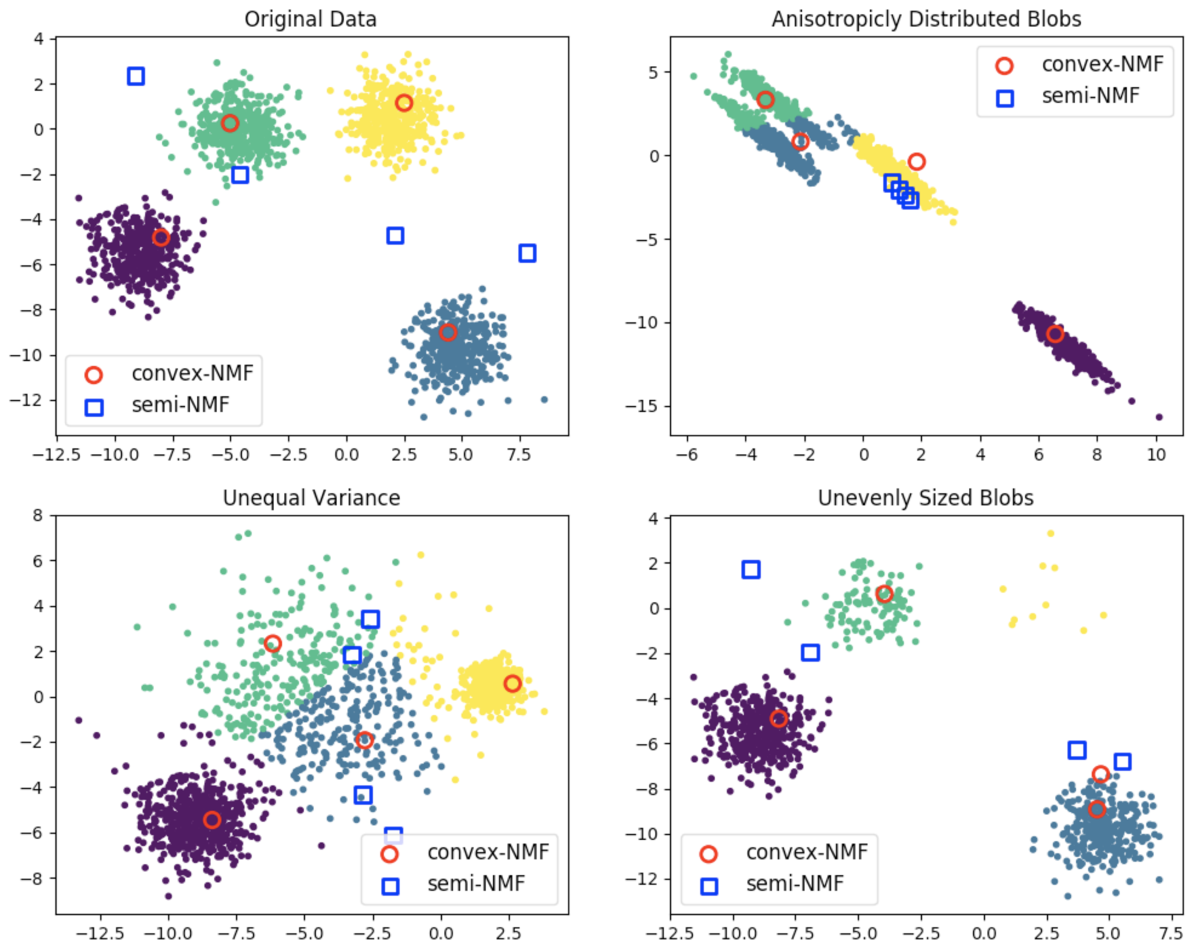

Fig. 1. Comparisons of the Convex-NMF and Semi-NMF on clustering. The decomposition rank is set to 4. For figure on the upper left, the input data is four evenly sized blobs with 1500 data points. For the other three figures, the input data are anistropicly distributed blobs, unequal variance blobs and unevenly sized blobs, respectively. The red dots and blue squares are the clustering centroids from the left decomposed matrix.

We also showed the prediction error for different methods on the data described above in Table I. The error is defined as the Frobenius norm of the difference between the predicted category matrices and the real category matrices. Convex-NMF is always performing better than Semi-NMF, and is performing pretty well on the first three data.

error	kmeans	convex-NMF	semi-NMF
Evenly sized blobs	0.0	2.24	32.11
Anisotropicly distributed data	29.15	28.23	35.94
Different variance data	27.86	29.24	35.27
Unevenly sized blobs	0.0	10.54	24.49

TABLE I
COMPARISON OF PREDICTION ERROR WITH DIFFERENT METHODS ON BLOBS WITH DIFFERENT DISTRIBUTIONS.

III. GEOMETRIC INTERPRETATION OF NMF AND NP-HARDNESS

A. NMF is NP-hard

One interesting aspect of NMF is its analysis in geometric terms. Specifically, the author in [6] shows that NMF is a strict generalization of what is known as the nested polytope problem (or the intermediate simplex problem [12]), using an argument we will detail below. By showing an example of the nested polytope problem where it is visually obvious that solutions are non-unique, the author heuristically motivates the non-uniqueness of solutions to NMF.

Moreover, we have the following:

Theorem III.1. *The nested polytope problem is NP-hard (see [1]).*

Therefore, the fact that NMF reduces to the intermediate simplex problem in special cases, or rather that any solution to NMF implies a solution of the nested polytope problem implies the following:

Theorem III.2. *NMF is NP-hard.*

(For a source which details the NP-hardness of the nested polytope problem and relates it explicitly to NMF and the difficulties involved in solving that problem, see [12].)

Note that this NP-hardness result does *not* contradict the result in [11] that:

Theorem III.3. *NMF can be formulated as a convex problem.*

This is because the convex problem to which NMF can be reduced is copositive programming, i.e. optimization over the convex cone of copositive matrices, and it is known (cf. [3]) that:

Theorem III.4. *Copositive programming is NP-hard.*

The NP-hardness of copositive programming corresponds to the absence of any known efficiently computable barrier functions [11] for applying interior point methods over the copositive cone.

B. Relationship to Intermediate Simplex Problem

The author in [6] shows the relationship between NMF and the nested polytope problem in the case of exact NMF (where $\mathbf{M} = \mathbf{UV}$, not just $\mathbf{M} \approx \mathbf{UV}$).

Definition III.5. *The **nested polytope problem** is the problem of finding a polytope U, given polytopes P_1 and P_2, such that*

$$P_1 \subseteq U \subseteq P_2 \,,$$

where P_1, P_2, and U are usually assumed to all have the same definition.

One can remove any columns which are all zero without loss of generality. Therefore no problems arise when we attempt to normalize all of the columns to have length 1 in the ℓ^1 norm. It turns out that this normalization can also be done without loss of generality, since

$$\mathbf{MD}_M^{-1} = \mathbf{UD}_U^{-1}\mathbf{D}_U\mathbf{VD}_M^{-1} \ ,$$

with $\mathbf{D}_M$ a diagonal matrix such that

$$\mathbf{D}_M(j,j) = ||m_j||_1 = ||\mathbf{M}(:,j)||_1 \ ,$$

and $\mathbf{D}_U$ entirely analogously defined to be a diagonal matrix such that

$$\mathbf{D}_U(j,j) = ||u_j||_1 = ||\mathbf{U}(:,j)||_1 \ .$$

The claim is that this gives an NMF

$$\tilde{\mathbf{M}} = \tilde{\mathbf{U}}\tilde{\mathbf{V}} \ ,$$

such that the columns of all matrices are normalized by 1, by taking

$$\tilde{\mathbf{M}} := \mathbf{MD}_M^{-1} \ , \quad \tilde{\mathbf{U}} := \mathbf{UD}_U^{-1} \ , \quad \tilde{\mathbf{V}} := \mathbf{D}_U\mathbf{VD}_M^{-1} \ .$$

$\tilde{\mathbf{M}}$ and $\tilde{\mathbf{U}}$ by construction clearly have the desired properties, so in order for the claim to hold, all that remains to show is that for every $j = 1, \dots, n$:

$$||\tilde{\mathbf{V}}(:,j)||_1 = \sum_{k=1}^{r} \tilde{v}_{kj} = 1 \ .$$

This is true since for all $j = 1, \dots, n$:

$$1 = ||\tilde{\mathbf{M}}(:,j)||_1 = ||\tilde{\mathbf{V}}(:,j)||_1 \ ,$$

since

$$||\tilde{\mathbf{M}}(:,j)||_1 = \sum_{l=1}^{p} m_{lj} = \sum_{l=1}^{p} (\tilde{\mathbf{U}}\tilde{\mathbf{V}}(:,j))_l =$$
$$\sum_{l=1}^{p}\sum_{k=1}^{r} \tilde{u}_{lk}\tilde{v}_{kj} = \sum_{k=1}^{r}\sum_{l=1}^{p} \tilde{v}_{kj}\tilde{u}_{lk} = \sum_{k=1}^{r} \left(\tilde{v}_{kj} \sum_{l=1}^{p} (\tilde{u}_{lk}) \right) =$$
$$\sum_{k=1}^{r} \tilde{v}_{kj}||\mathbf{U}(:,k)||_1 = \sum_{k=1}^{r} \tilde{v}_{kj}(1) = \sum_{k=1}^{r} \tilde{v}_k j = ||\tilde{\mathbf{V}}(:,j)||_1 \ ,$$

due to the columns of $\tilde{\mathbf{U}}$ being normalized. Since the columns of $\tilde{\mathbf{V}}$ sum to 1, and because we have that (due to the data reduction of matrix factorization given above) for all $j = 1, \dots, n$:

$$\tilde{\mathbf{M}}(:,j) = \sum_{k=1}^{r} \tilde{v}_{kj}\tilde{\mathbf{U}}(:,k)$$

it follows that the columns of $\tilde{\mathbf{M}}$ are all convex combinations of the columns of $\tilde{\mathbf{U}}$, thus:

$$\tilde{\mathbf{M}}(:,j) \in \operatorname{conv}(\tilde{\mathbf{U}}(:,1), \dots, \tilde{\mathbf{U}}(:,r)) =: \operatorname{conv}(\tilde{\mathbf{U}}) \ \forall j = 1, \dots, n \ .$$

Since convex sets like $\operatorname{conv}(\tilde{\mathbf{U}})$ are closed under convex combinations, it follows that the convex hull of all of the $\tilde{\mathbf{M}}(:,j)$, which we will denote by $\operatorname{conv}(\tilde{\mathbf{M}})$, must be a subset of $\operatorname{conv}(\tilde{\mathbf{U}})$, in other words we have shown

$$\operatorname{conv}(\tilde{\mathbf{M}}) \subseteq \operatorname{conv}(\tilde{\mathbf{U}}) \ .$$

On the other hand, since every one of the $\tilde{\mathbf{U}}(:,1), \dots, \tilde{\mathbf{U}}(:,r)$ is a vector in $\mathbb{R}^p$ with all non-negative entries which sum to 1, we have that all of these vectors are in Δ_p, the p-simplex. And since the p-simplex is convex and thus closed under convex combinations, we have further that:

$$\operatorname{conv}(\tilde{\mathbf{M}}) \subseteq \operatorname{conv}(\tilde{\mathbf{U}}) \subseteq \Delta_p \ .$$

Therefore finding an NMF $\tilde{\mathbf{U}}\tilde{\mathbf{V}}$ for $\tilde{\mathbf{M}}$ implies finding a polytope, $\operatorname{conv}(\tilde{\mathbf{U}})$, which is ”nested” in between the two polytopes $\operatorname{conv}(\tilde{\mathbf{M}})$ and Δ_p. And once we have found an NMF for $\tilde{\mathbf{M}}$, we can easily use that to recover an NMF for $\mathbf{M}$ by scaling the factors appropriately. The dimension of $\mathrm{conv}(\tilde{\mathbf{M}})$ is known in advance and equal to $\mathrm{rank}(\tilde{\mathbf{M}}) - 1 = \mathrm{rank}(\mathbf{M}) - 1$. Similarly, the dimension of Δ_p is also known in advance to be $p-1$. But the dimension of $\mathrm{conv}(\tilde{\mathbf{U}})$ is not known in advance. In the special case where $\mathrm{rank}(\mathbf{M}) = p$, and thus the dimensions of all three polytopes must coincide, this is the nested polytope problem.

IV. Convex Relaxations of Nonnegative Matrix Factorization

In this chapter, we talk about a convex NMF formulation which takes into account how NMF can be formulated as a convex cone program [11]. Although it was widely believed that NMF is a non-convex problem and only local minima can be found, the authors [11] showed, that this relation of NMF to CP matrices can be used to prove that a non-trivial NMF solution exists for every non-positive matrix. However, this approach doesn't give a practically realizable solution/algorithm and hence we shall discuss approximations derived out of this scheme. Despite the existence of the convex formulation, they also show that a formulation of the problem as a generalized geometric program, which is non-convex, could give a better approach for finding the global optimum. After that, we shall also discuss some content from [10], where the completely-positive NMF formulation is used to derive practically useful projected gradient descent algorithms for NMF under sparse matrix settings.

A. Review of some properties

We first review some basic definitions about Completely Positive Matrices and CP-Rank and then go ahead to discuss how these are relevant in the context of NMF.

Definition IV.1. *A matrix $\mathbf{X} \in \mathbb{S}^n$ is called completely positive if all entries of $\mathbf{X}_{ij} \geq 0$ and it can be factorized as*

$$\mathbf{X} = \mathbf{B}\mathbf{B}^\top$$

for some $\mathbf{B} \in \mathbb{R}_+^{n \times k}$.

Definition IV.2. *The cp-rank of a completely positive matrix $\mathbf{X}$ is given by the smallest k, such that we can express*

$$\mathbf{X} = \sum_{i=1}^{k} u_i u_i^\top$$

with $u_i \geq 0$.

Theorem IV.3. *Let $\mathcal{K} \subset \mathbb{R}^{m \times m}$ be the cone of completely positive matrices. Then the dual cone is given by*

$$\mathcal{K}^* = \{\mathbf{W} \in \mathbb{R}^{m \times m} | y^\top \mathbf{W} y \geq 0, \quad \forall y \geq 0\}$$

– which is the set of co-positive matrices. Checking for co-positivity of a matrix is NP-hard.

Theorem IV.4. *The cp-rank of a matrix $\mathbf{X} \in \mathbb{R}_+^{n \times n}$ satisfies the following property:*

$$\textit{rank}(\mathbf{X}) \leq \textit{cp-rank}(\mathbf{X}) \leq \frac{n(n+1)}{2}.$$

Proof: (Sketch) For the first part of the inequality, cp-rank$(\mathbf{X}) \geq$ rank$(\mathbf{X})$, because SVD of a matrix is the most compact decomposition of a matrix into rank$(\mathbf{X})$ many components. Now, a completely positive decomposition just imposes additional constraints on the factorized vectors, thereby making the number of components larger, hence cp-rank$(\mathbf{X}) \geq$ rank$(\mathbf{X})$. For the second part, we can show that a completely positive decomposition of size $\frac{n(n+1)}{2}$ of a CP-matrix exists – which is designed trivially by selecting each vector in the factorization to model exactly one element of the matrix $\mathbf{X}$, as and the matrix $\mathbf{X}$ is symmetric, we have at most $\frac{n(n+1)}{2}$ distinct elements, hence the bound.

Theorem IV.5. *If a matrix $\mathbf{X} \in \mathbb{S}_+^n$ is such that $[\mathbf{X}]_{ij} \geq 0 \ \forall \ i,j \in [1,\cdots,n],[1\cdots n]$ and rank($\mathbf{X}$) ≤ 2, then cp-rank($\mathbf{X}$) = rank($\mathbf{X}$). Proof: (Sketch) We know by using the previous theorem, the cp-rank of matrix is lower bounded by its rank. Now, consider the SVD for $\mathbf{X} = us_1u^\top + vs_2v^\top$, where u and v are the two singular vectors (as rank ≤ 2). Now, as $s_1, s_2 > 0$, we can flip signs of negative entries in vectors u and v such that all entries in u' and $v' \geq 0$ and subsume s_1, s_2 in u' and v'. The sum $u'u'^\top + v'v'^\top$ is still equal to $\mathbf{X}$, and u', v' are CP. Hence, cp-rank$(\mathbf{X}) \leq$ rank$(\mathbf{X})$, which implies that cp-rank($\mathbf{X}$) = rank($\mathbf{X}$) in this special case.*

Theorem IV.6. *If a matrix $\mathbf{X} \in \mathbb{S}^n \cap \mathbb{R}_+^{n \times n}$ is diagonally dominant, then it is CP. Proof: (Sketch) We know that every diagonally dominant symmetric matrix with non-negative diagonal entries is positive semi definite. Hence, a CP matrix which is diagonally dominant is PSD. So, we can write the SVD/ eigenvalue decomposition for this matrix $\mathbf{X}$ as $\mathbf{X} = \sum_{i=1}^n u_i \sigma_i u_i^\top$ and then flip signs of entries of u_i which are negative. $\sigma_i \geq 0$, which means that we can multiply u_i by $\sqrt{\sigma_i}$.*

Theorem IV.7. *If $\mathbf{X} \in \mathbb{S}_+^n$ is a positive semi-definite matrix, then $\exp_H(\mathbf{X})$ which is the Hadamard (or component-wise) exponential of $\mathbf{X}$ is completely positive. Proof: We talk about this in the Section F, where we explicitly show why this holds and how we can decompose/factorize this Hadamard exponential.*

Now, we are ready to show that every non-negative matrix has a non-trivial, NMF of the form $\mathbf{V} = \mathbf{W}\mathbf{H}$.

B. Existence of NMF factorization

Theorem IV.8. *Every Non-negative matrix $\mathbf{V} \in \mathbb{R}^{m\times n}$ has a non-trivial, non-negative matrix factorization of the form $\mathbf{V} = \mathbf{WH}$.*

Proof: Consider the following matrix:

$$\mathbf{Z} = \begin{bmatrix} \mathbf{D} & \mathbf{V} \\ \mathbf{V}^\top & \mathbf{E} \end{bmatrix} .$$

Now, the theorem translates to showing that one could always find a $\mathbf{B} \in \mathbb{R}_+^{n\times k}$ such that $\mathbf{Z} = \mathbf{BB}^\top$. If this is true, then $\mathbf{B}$ can take the form of

$$\mathbf{B} = \begin{bmatrix} \mathbf{W} \\ \mathbf{H}^\top \end{bmatrix} .$$

If $\mathbf{D}$ and $\mathbf{E}$ are arbitrary diagonally dominant completely positive matrices, then $\mathbf{B}$ always exists – this is because $\mathbf{W}$ and $\mathbf{H}$ are independently factorizations of diagonally dominant matrices and all diagonally dominant matrices are CP and a NMF for a CP-matrix always exists. (Follows from Theorem IV.6, IV.2) Now, let's choose $\mathbf{D}$ and $\mathbf{E}$ in such a way that they are diagonal matrices and their diagonal entries are more than the absolute sum of rows and columns of $\mathbf{V}$, thus making $\mathbf{Z}$ diagonally dominant. As $\mathbf{Z}$ is diagonally dominant, it is CP (Theorem IV.6), and hence $\mathbf{B}$ exists. Taking a closer look, the factorization of $\mathbf{V}$ then becomes $\mathbf{V} = \mathbf{WH}$, which also exists.

C. Solving NMF optimization problems

We first describe how NMF can be formulated as a rank minimization problem on the cone of the completely positive matrices and then go ahead to describe the convex approximate formulations of NMF:

1) NMF as a convex conic program:

Theorem IV.9. *The set of Completely Positive Matrices $\mathcal{K}^{CP}$ is a convex cone. Proof: Done in Class (EE 227B), as a pre-midterm quiz question.*

The problem of finding the minimum rank NMF factorization (as it is desirable to do so for the sake of compactness) can be cast as the following optimization problem inspired from the last section:

$$\min_{\mathbf{D},\mathbf{E}} \text{ rank} \begin{bmatrix} \mathbf{D} & \mathbf{V} \\ \mathbf{V}^\top & \mathbf{E} \end{bmatrix} \text{ subject to } \mathbf{D} \in \mathcal{K}^{CP}, \mathbf{E} \in \mathcal{K}^{CP}.$$

Since minimizing the rank is non-convex, we can use its convex envelope, which is the trace of the matrix (or the nuclear norm) to relax the problem as:

$$\min_{\mathbf{D},\mathbf{E}} \text{Tr} \begin{bmatrix} \mathbf{D} & \mathbf{V} \\ \mathbf{V}^\top & \mathbf{E} \end{bmatrix} \text{ subject to } \mathbf{D} \in \mathcal{K}^{CP}, \mathbf{E} \in \mathcal{K}^{CP}.$$

After determining $\mathbf{D}$, $\mathbf{E}$, $\mathbf{W}$ and $\mathbf{H}$ can be recovered by CP factorization of $\mathbf{D}$, $\mathbf{E}$, which is hard in itself. The authors point out that there is no practical barrier function known for the CP cone so this means that Primal-Dual interior point methods or Barrier/Penalty methods can't be applied. So, even though this relaxation with the nuclear norm seems convex, there wouldn't be any practical algorithm to solve this problem. However, this result is quite nice and intriguing.

D. Convex relaxations of the NMF problem

The authors introduce several convex relaxations of the convex formulation of the NMF problem, which are practically realizable. Some of these are described below:

1) **Convex upper bound with Singular Value Decomposition** – Briefly speaking, compute the SVD of a matrix $\mathbf{A} = \mathbf{UV}$, then we project the matrices $\mathbf{U}$ and $\mathbf{V}$ to the non-negative orthant. This is called Clipped SVD (CSVD). In general, this might not be a good approximation, but practically speaking the authors say that this seemed to be a strong baseline.
2) **Relaxation with a positive semidefinite cone** – In general, people solve the following optimization for NMF

$$\min_{\mathbf{W}\geq 0,\mathbf{H}\geq 0} ||\mathbf{V} - \mathbf{WH}||_2 .$$

One approach could be to unfold the matrices $\mathbf{W}, \mathbf{H}$ into a large vector $z = [\mathsf{vec}(\mathbf{W}^\top); \mathsf{vec}(\mathbf{H})]$. Now, if $z \geq 0$ and $\mathbf{Z} = zz^\top$, then the terms in $||\mathbf{V} - \mathbf{WH}||_2$ are linear in $\mathbf{Z}$. Now, we could write the optimization problem in terms of the entries in $\mathbf{Z}$. So, we end up solving the following problem:

$$\min \mathsf{Tr}(\mathbf{Z}) \quad \text{s.t.}$$
$$\mathbf{A}\cdot\mathbf{Z} = [\mathbf{V}]_{ij}, \quad \mathbf{Z} \succeq 0, \quad \mathbf{Z} - zz^\top = 0, \quad \mathbf{Z} \geq 0 .$$
$$\min \mathsf{Tr}(\mathbf{Z}) \quad \text{s.t.}$$
$$\mathbf{A}\cdot\mathbf{Z} = [\mathbf{V}]_{ij}, \quad \mathbf{Z} \succeq 0, \quad \mathbf{Z} - zz^\top \succeq 0, \quad \mathbf{Z} \geq 0 .$$

Here $\mathbf{A}$ is the matrix, that masks out relevant entries in the matrix $\mathbf{Z}$, that is entries which sum to $[\mathbf{V}]_{ij}$. The rank 1 constraint is relaxed to $\mathbf{Z} - zz^\top \in \mathbb{S}_+$ (similar to the SDP relaxation of max-cut). After solving this problem, the solution could be found on the first eigenvector of $\mathbf{Z}$ (by construction). Due to this, the degree of approximation is also determined by the ratio of the first eigenvalue of $\mathbf{Z}$ to the sum of all others. Positivity of $\mathbf{Z}$ would guarantee that the first eigenvector will have elements of the same sign, if they are not, we can still flip signs of the negative entries. This method seems interesting, but the complexity of this method is too high. $\mathbf{Z}$ is a $(N+m)k \times (N+m)k$ matrix and then the number of non-negativity constraints are of the same order $\mathcal{O}((N+m)k \times (N+m)k)$ as well. But a lot of these non-negativity constraints are not really required as they don't relate to the entries in $\mathbf{V}$ in any manner. So, in order to make this tractable, the next approach talks about how can the number of non-negativity constraints be reduced.

3) **Approximating the SDP cone with smaller ones** – As described in the previous point, the complexity of the previous relaxed approach is too much and one way to cut down would be to reduce the number of non-negativity constraints. So, instead of having a big SDP cone on $(N+m)k \times (N+m)k$ sized matrices, we can have smaller SDP cones. Analogously to the way we constructed the vector z earlier, we define

$$z_{ij} = [\mathsf{vec}(W_i^\top); \mathsf{vec}(H_j)],$$

which is a $2k$ dimensional vector, and then define a $2k \times 2k$ matrix

$$\mathbf{Z}_{ij} = \begin{bmatrix} \mathbf{W}_i^\top \mathbf{W}_i & \mathbf{W}_i^\top \mathbf{H}_j \\ \mathbf{W}_i^\top \mathbf{H}_j & \mathbf{H}_j^\top \mathbf{H}_j \end{bmatrix}.$$

Now the optimization problem becomes the following:

$$\min \sum_{i=1}^{N} \sum_{j=1}^{m} \mathsf{Tr}(\mathbf{Z}_{ij}) \quad \text{s.t.} \quad \mathbf{Z}_{ij} \geq 0, \quad \mathbf{Z}_{ij} \in \mathbb{S}_+^{2k \times 2k},$$
$$\mathbf{A}_{ij} \cdot \mathbf{Z}_{ij} = \mathbf{V}_{ij}, \quad \mathbf{Z_{ij}} = z_{ij} z_{ij}^T.$$

The number of constraints in this optimization problem (SDP) are of the order of $N \times m$ and hence, in terms of storage it needs $(N+m)$, $k \times k$ sized PSD matrices for each row/column of $\mathbf{W}$ and $\mathbf{H}$ and Nm symmetric positive definite $k \times k$ matrices for the constraints. The storage complexity hence is $\mathcal{O}((Nm+N+m)k^2)$ which is significantly cheaper than $\mathcal{O}((N+m)k)^2)$ complexity of the previous method.

In terms of solving this SDP practically, an interior point method that would require inverting a symmetric matrix would require $(N+m)^3 k^3$ steps in the previous formulation but would require inverting only Nm many $2k \times 2k$ matrices (= $Nm \times 8k^3$, which is significantly lesser than the last approach. In terms of performance, this method is similar to the bigger cone and comparable to the CSVD approach.

4) **NMF as a convex multi-objective program:** In this approach the authors define a convex set in which the solution to the NMF problem lies and search for solutions over this set. More formally, say we want to match each entry of the matrix $\mathbf{V}_{ij}$ to $\mathbf{W}_i \mathbf{H}_j = \sum_{l=1}^{m} \mathbf{W}_{il} \mathbf{H}_{lj}$ and let $\mathbf{V}_{ij,l} = \mathbf{W}_{il} \mathbf{H}_{lj}$. Now, it can be shown that this can be done by controlling the ratio of the $L2/L1$ norms of W and H. Consider the following matrix, that is required to be PSD.

$$\begin{bmatrix} 1 & \mathbf{W}_{il} & \mathbf{H}_{ij} \\ \mathbf{W}_{il} & t_{il} & \mathbf{V}_{ij,l} \\ \mathbf{H}_{ij} & \mathbf{V}_{ij,l} & t_{jl} \end{bmatrix} \succeq 0$$

On using Schur's complement, we can show that we need the following,

$$\begin{bmatrix} t_{il} - \mathbf{W}_{il}^2 & \mathbf{V}_{ij,l} - \mathbf{W}_{il}\mathbf{H}_{lj} \\ \mathbf{V}_{ij,l} - \mathbf{W}_{il}\mathbf{H}_{lj} & t_{jl} - \mathbf{H}_{lj}^2 \end{bmatrix} \succeq 0$$

Now, we get the following to hold true by applying Schur's complement again:

$$t_{il} \geq \mathbf{W}_{il}^2, \quad t_{jl} \geq \mathbf{H}_{lj}^2$$
$$(t_{il} - \mathbf{W}_{il}^2)(t_{jl} - \mathbf{W}_{lj}^2) \geq (\mathbf{V}_{ij,l} - \mathbf{W}_{il}\mathbf{H}_{lj})^2$$

Now, the L2-error $\sum_{i=1}^{N} \sum_{j=1}^{m} \sum_{l=1}^{k} (\mathbf{V}_{ij,l} - \mathbf{W}_{il}\mathbf{H}_{lj})^2$ becomes 0 if $t_{il} = \mathbf{W}_{il}^2$ and $t_{jl} = \mathbf{H}_{lj}^2$ for all i, j, l. This can be cast as a multi-objective optimization problem, using the $L1 - norm$, and we point the reader to the main paper [11] for more details regarding vector optimization.

5) **NMF as a Generalized Geometric Program:** The authors cast NMF as a generalised geometric program, which we describe in this section. Consider the objective,

$$\min ||\mathbf{V} - \mathbf{W}\mathbf{H}||_2 = \min \sum_{i=1}^{N} \sum_{j=1}^{m} \sum_{l=1}^{k} (\mathbf{V}_{ij} - \mathbf{W}_{il}\mathbf{H}_{lj})^2$$

(Note the 2-norm instead of Frobenious norm) Now, we make the transformation that $\mathbf{W}_{il} = \exp(w_{il})$ and $\mathbf{H}_{lj} = \exp(h_{lj})$, and so the objective would simplify to the following:

$$||\mathbf{V} - \mathbf{W}\mathbf{H}||_2 = \sum_{i=1}^{N} \sum_{j=1}^{m} \mathbf{V}_{ij}^2$$
$$+ \sum_{i=1}^{N} \sum_{j=1}^{m} \Big(\sum_{l=1}^{k} \exp(w_{il} + h_{lj}) \Big)^2$$
$$-2 \sum_{i=1}^{N} \sum_{j=1}^{m} \mathbf{V}_{ij} \sum_{l=1}^{k} \exp(w_{il} + h_{lj})$$

The first term can be ignored and the other two terms called $f(\mathbf{W}, \mathbf{H})$ and $g(\mathbf{W}, \mathbf{H})$ are convex form of posynomials. There are known algorithms for solving for the global optimum of the optimization problem

$$\min_{\mathbf{W}, \mathbf{H}} f(\mathbf{W}, \mathbf{H}) - g(\mathbf{W}, \mathbf{H})$$

Take for instance [5] which talks about global optimization of this form. The authors compared this with the preceding methods, and showed that this global optimization method gave orders of magnitude less error as compared to the earlier approaches.

E. Isometric NMF

In order to describe a build up to isometric NMF, the authors first describe various methods to the problem of manifold learning. Manifold learning requires representing data in a low dimensional space while preserving local distances. The main difference in various methods for manifold learning is the modelling of distances between points that are not in the local neighbourhood of a point. Isometric NMF is a hybrid of NMF techniques and isometric manifold modeling techniques. Imposing this property to be able to generate isometric embeddings is helpful in interpretation of the kind of solutions NMF gives us. The goal is to be able to interpret solutions that NMF gives them under added constraints.

1) Convex Maximum Furthest Neighbour Unfolding (MFNU): MFVU is a variant of MVU (Maximum Variance Unfolding) that tries to preserve local distances and maximizes distance between furthest neighbours. Formally, this means the following problem: Given a data matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$, and the kernel gram matrix $\mathbf{G} = \mathbf{X}\mathbf{X}^\top$, find a new gram matrix $\mathbf{K}$ such that $\text{rank}(\mathbf{K}) < \text{rank}(\mathbf{G})$. Now this gram matrix factorizes into data points of dimensionality $d' < d$. Now, Isometric Unfolding requires us to have euclidean distances in $\mathbb{R}^{d'}$ for a neighbourhood around every point be preserved from $\mathbb{R}^d$, so, we can express this using the gram matrices:

$$\mathbf{K}_{ii}+\mathbf{K}_{jj}-\mathbf{K}_{ij}-\mathbf{K}_{ji} = \mathbf{G}_{ii}+\mathbf{G}_{jj}-\mathbf{G}_{ij}-\mathbf{G}_{ji} \;\; \forall i,j \in I_i$$

I_i is defined as the neighbourhood set of the point i. Now, MVFU maximizes the distances between furthest neighbours not in the neighbourhood set. So, we can write it as the following problem:

$$\max_{\mathbf{K}} \sum_{i=1}^{N} \mathbf{B}_i \cdot \mathbf{K} \quad \text{s.t.} \quad \mathbf{A}_{ij} \cdot \mathbf{K} = d_{ij} \;\; \forall j \in I_i, \quad \mathbf{K} \succeq 0$$

where the matrices $\mathbf{A}_{ij}$ is such that $\mathbf{A}_{ij} \cdot \mathbf{K} = \mathbf{K}_{ii}+\mathbf{K}_{jj}-\mathbf{K}_{ij}-\mathbf{K}_{ji}$. Matrix $\mathbf{B}_i$ computes the distance between the point i and it's furthest neighbour. The new projections are the eigenvectors of the solution $\mathbf{K}$. In general, MFNU gives matrices that have more compact spectrums as compared to traditional PCA.

2) Non-convex MFNU: We can make the problem non-convex by adding a rank constraint to the last optimization problem. This means that we explicitly parameterize the rank of the matrix $\mathbf{K}$. This can be done by parameterizing the matrix $\mathbf{K} = \mathbf{R}\mathbf{R}^T$. The authors say that the global minimum of this optimization problem is the same as the convex one. This can be solved by the ADMM (Augmented Lagrangian Method, using dual gradient descent or using L-BFGS algorithm).

3) Convex IsoNMF: Using the relation between NMF and CP matrices (as discussed in the previous sections), we can cast isoNMF (= NMF+ MFNU) in the following manner:

$$\max_{\tilde{\mathbf{W}},\tilde{\mathbf{H}}} \sum_{i=1}^{N} \mathbf{B_i} \cdot \mathbf{Z} \quad \text{s.t.} \quad \mathbf{A}_{ij} \cdot \tilde{\mathbf{W}} = d_{ij},$$
$$\mathbf{Z} = \begin{bmatrix} \tilde{\mathbf{W}} & \mathbf{V} \\ \mathbf{V}^\top & \tilde{\mathbf{H}} \end{bmatrix}, \quad \mathbf{Z}, \tilde{\mathbf{W}}, \tilde{\mathbf{H}} \in \mathcal{K}^{\mathcal{CP}}$$

In the end, we can recover $\mathbf{W}$ and $\mathbf{H}$, by using the fact that $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{W}^\top$ and $\tilde{\mathbf{H}} = \mathbf{H}\mathbf{H}^\top$. This formulation is again not practically realizable, as it involves checking beloningness to the cone of completely positive matrices which is hard, but then all relaxations discussed to this constraint discussed so far still remain analogously applicable.

4) Non-Convex IsoNMF: Analogous to the non-convex formulation of MFNU, we can define the non-convex formulation of isoNMF, where we parameterize the matrix $\mathbf{Z}$ as $\mathbf{W}\mathbf{W}^\top$.

F. Some more (global) convex approximations to NMF

In the previous section, we described various ways of relaxing the NMF problem which mostly converted the NMF problem into an SDP. Apart from that in practical scenarios, we often haqve sparsity/rank constraints and we can exploit them to get simpler ways for solving the NMF problem. In the following section, we talk about approximations we can make to the NMF problem under these conditions, and give a few examples.

In its most general form, NMF can be cast in the following template.

$$\min_{\substack{\mathbf{U} \in \mathbb{R}^{m \times k}, \\ \mathbf{V} \in \mathbb{R}^{n \times k}}} \mathsf{loss}(\mathbf{M}, \mathbf{U}\mathbf{V}^\top) \text{ subject to } \mathbf{U}, \mathbf{V} \geq 0$$

In general, we could choose this $\mathsf{loss}(\mathbf{X}, \mathbf{Y})$ function to be the Mean-Squared Error (MSE) :

$$||\mathbf{X} - \mathbf{Y}||_F^2$$

or a generalized version of the Kullback-Leibler (KL) Divergence:

$$\sum_{i,j \in [1,m] \times [1,n]} [\mathbf{X}]_{ij} \log \frac{[\mathbf{X}]_{ij}}{[\mathbf{Y}]_{ij}} + [\mathbf{Y}]_{ij} - [\mathbf{X}]_{ij} \, .$$

The authors first discuss the case when the data matrix A is symmetric and then exetend it to the non-symmetric setting.

G. Global Approximations for NMF with synmmetric data matrices

We start with the most basic case when the factorization is supposed to be symmetric. In this case, the optimization problem is given by:

$$\min_{\mathbf{U}\in\mathbb{R}^{n\times k}} \mathsf{loss}(\mathbf{A}, \mathbf{U}\mathbf{U}^\top) \text{ subject to } \mathbf{U} \geq 0 \qquad (1)$$

We first introduce some definitions in order to finally describe a relaxation to this problem.

So using IV.7, we reparameterize the matrix to be $\mathbf{A}$ such that we assume $[\mathbf{A}]_{ij} = \exp([\mathbf{X}]_{ij})$ (where $\mathbf{X}$ is any positive semidefinite matrix – Theorem IV.7 says that Hadamard exponential of a PSD matrix is completely positive, and we used it here). Now, we get a new sufficient convex condition on $\mathbf{X}$. The convex restriction of the problem can be interpreted as the following:

$$\min_{\mathbf{X}\in\mathcal{S}_+^n} \mathsf{loss}(\mathbf{A}, \exp_H(\mathbf{X}))$$

If the $\mathsf{loss}(\cdot,\cdot)$ is the KL divergence, then the problem is convex in $\mathbf{X}$. If we use the MSE, then the problem can be made convex by adding another constraint $[\mathbf{A}]_{ij}/2 \leq \exp([\mathbf{X}]_{ij})$. The next important question is how we could factorize $\exp(\mathbf{X})$. We answer this in the next subsection.

1) Factorizing $\exp_H(\mathbf{X})$ *when* $\mathbf{X}$ *is PSD:* Lets start by considering the Hadamard product for two matrices which are completely positive.

So, lets assume $\mathbf{A} = \sum_{i=1}^k a_i a_i^\top$ and $\mathbf{B} = \sum_{i=1}^l b_i b_i^\top$, where $a_i \geq 0, b_i \geq 0$. Then, the Hadamard product of the two matrices can be written as:

$$\mathbf{A}\circ\mathbf{B} = \sum_{i=1}^k\sum_{j=1}^l (a_i \circ b_j)(a_i\circ b_j)^\top$$

This means that the Hadamard product of two completely positive matrices is also completely positive (as it is a sum of rank 1 completely positive matrices).

Now, from Theorem IV.7, which says that $\exp_H(\mathbf{X})$ is completely positive when X is PSD, we can say that there is a matrix $\mathbf{U}$ such that $\exp_H(\mathbf{X}) = \mathbf{U}\mathbf{U}^\top, \mathbf{U} \in \mathbb{R}^{n\times k}$. Now, we can also bound the size of $\mathbf{U}$ or the cp-rank of $\exp_H(\mathbf{X})$. Now, the cp-rank of a matrix $\mathbf{X} = \mathbf{U}\mathbf{U}^\top$ is such that, $r \leq$ cp-rank(X) $\leq \frac{r(r+1)}{2} - 1$, where $r =$ Rank($\mathbf{X}$). This comes directly from Theorem IV.4. Note that cp-rank($\mathbf{X}$) is also the rank of $\mathbf{U}$ in this case.

Now, as $\mathbf{X}$ is Positive semi-definite, we can write the following:

$$\exp_H(\mathbf{X}) = \exp_H\left(\sum_{i=1}^n \lambda_i x_i x_i^\top\right) = \prod_{i=1}^n \exp_H(\lambda_i x_i x_i^\top)$$

So essentially, we broke down the summation into product of the Hadamard exponentials of the Rank-1 matrices. Now, consider evaluating $\exp_H(vv^\top)$ for some vector $v = \sqrt{\lambda_i}x_i$. Let

$$M = \max_{i=1,\cdots,n} |v_i|\,,$$

$$\exp_H(vv^\top)_{ij} = \exp(v_i v_j) = \exp(-M^2 + (M+x_i)(M+x_j) - M(x_i+x_j))\,.$$

So, now, what we can do is that, we can write: $\exp_H(vv^\top) = \exp(-M^2)\exp_H(yy^\top)\circ zz^\top$ where $y = M\mathbf{1} + v$ and $z = \exp_H(-Mv)$ are both non-negative vectors.

Now, as y is non-negative, we can write:

$$\exp_H(yy^\top) = \sum_{i=1}^\infty \frac{(yy^\top)^{\circ i}}{i!}$$

with $(yy^\top)^{\circ k}_{ij} = (y_i y_j)^k$, the matrix $\exp_H(yy^\top)$ is completely positive. So, this gives us a way to factorize $\exp_H(\mathbf{X})$ when $\mathbf{X}$ is PSD. Also, note that the above argument shows that $\exp_H(\mathbf{X})$ is completely-positive when $\mathbf{X}$ is PSD. The steps are summarized in the algorithm below:

```
Algorithm 1: Factorizing $\exp_H(\mathbf{X})$
1 Compute the eigvenvalue decomposition of $\mathbf{X}$,
    $\mathbf{X} = \sum_i \lambda_i x_i x_i^\top$
2 Decompose each factor, $\exp_H(v_i v_i^\top) =$
    $\exp(-M^2)\exp_H(y_i y_i^\top)\circ z_i z_i^\top; v_i = \sqrt{\lambda_i}x_i$ and
    $y_i, z_i$ are nonnegative vectors.
3 Approximate $\exp_H(y_i y_i^\top) = \sum_{i=1}^\infty \frac{(yy^\top)^{\circ k}}{k!}$.
4 Collect all the terms above using the chain rule to
    get $\exp_H(\mathbf{X}) = \mathbf{U}\mathbf{U}^\top$.
```

However, this algorithm will still give us $\mathbf{U}$ which can have large sizes, this is because of the cp-rank bound is quite loose. But in practice, we are fine with sparse solutions, as usually sparse solutions are more interpretable. So, in the next section we discuss how practically, we could achieve sparse decompositions.

2) Sparse Decomposition: When the matrix $\mathbf{A}$ itself is sparse, we try to find a sparse decomposition of $\mathbf{A}$, $\mathbf{A} = \mathbf{U}\mathbf{U}^\top$, where $\mathbf{U}$ is a sparse matrix. In this case, the change of variables to $\mathbf{A} = \exp(\mathbf{X})$ is not the most apt, something else should be done. The authors say that instead, one could directly find a low dimensional $\mathbf{X}$ which is positive semi-definite and has non-negative entries, in which case, we can see that the following property holds: If $\mathbf{X} \in \mathbb{S}_+^n$, and the entries in $\mathbf{X}$ are all non-negative, and Rank($\mathbf{X}$) = 2, then $\mathbf{X}$ is completely positive and rank is a good approximation to cp-rank. This follows from Theorem IV.5.

So, then solving the following optimization problem makes sense:

$$\min_{\mathbf{X}} ||\mathbf{A}-\mathbf{X}||^2 + \gamma|\mathbf{X}| + \nu\mathrm{Tr}(\mathbf{X}) \quad \text{subject to} \quad \mathbf{X}\geq 0, \mathbf{X}\in\mathbb{S}_+^n$$

where the Tr($\cdot$) controls the rank of $\mathbf{X}$ and the L1-norm penalty controls its sparsity. This suggests that when low-rank solutions are desired, it is easy to let-go of the cp-rank constraint and instead optimize for the rank constraint as in that regime, cp-rank is highly likely to be very close to the rank of the matrix for low ranks.

3) Recursive decomposition: Until now, the authors developed techniques for solving the optimization problem mentioned in Equation 1 under specific conditions, whose solutions might not turn out to be very accurate. So, the authors then propose a recursive scheme, where the same optimization for getting a factorization is iteratively solved over the residuals. We mention it briefly here but then discuss this scheme in detail in the next Chapter V on Non-negative matrix Underapproximation (NMU).

More formally, set $\mathbf{A}_0 = \mathbf{A}$ and let $\mathbf{A}_{k+1} = \mathbf{A}_k - \mathbf{U}_k\mathbf{U}_k^\top$, then in the $k-th$ step we try to solve for a factorization of $\mathbf{A}_k$. We might need to add a constraint which says that $\mathbf{A}_k \geq \mathbf{U}_k\mathbf{U}_k^\top$ holds elementwise, just to make sure the next residual is non-negative as well.

H. Global approximations to NMF with Non-Symmetric Data Matrices

In the previous section, we talked about how with symmetric data matrices, the hadamard exponential of a PSD matrix could be a good approximation for modelling the factorization. Additionally, we talked about how sparsity can be induced and how under low rank regimes, rank optimization could be good heuristic for cp-rank optimization. Next, we start by restating the NMF problem in the case of a non-symmetric factorization. The optimization problem (analogous to 1) in this case becomes the following

$$\min_{\substack{\mathbf{U}\in\mathbb{R}^{m\times k},\\ \mathbf{V}\in\mathbb{R}^{n\times k}}} \mathsf{loss}(\mathbf{A}, \mathbf{U}\mathbf{V}^\top) \text{ subject to } \mathbf{U} \geq 0, \mathbf{V} \geq 0\,. \tag{2}$$

As we discussed before in Sections 4.2 and 4.3, a matrix $\mathbf{A}$ has a non-negative matrix factorization if and only if there exist matrices $\mathbf{B}$ and $\mathbf{C}$ such that, the symmetric block matrix:

$$\begin{bmatrix} \mathbf{B} & \mathbf{A} \\ \mathbf{A}^\top & \mathbf{C} \end{bmatrix}$$

is completely positive. To avoid getting pathological solutions like a factorization where one of the matrices is Identity, we end up constraining the cp-rank of the solution.

It is hard to bound the cp-rank of the decomposition, as it is a non-convex constraint, but instead we could use a proxy for cp-rank, which is given by the rank of the matrix. This is justified from the bound on the cp-rank (Theorem IV.4) as described previously. In order to constrain the rank, we use the nuclear norm or the trace norm (Tr($\mathbf{X}$)) of a matrix as a convex lower bound proxy to the rank. Note that the following theorem holds true:

Theorem IV.10. *The nuclear norm of a matrix $\mathbf{X}$ is such that: $||\mathbf{X}||_* \leq t$ if and only if there exist symmetric matrices $\mathbf{Y}$ and $\mathbf{Z}$ such that the block matrix*

$$\begin{bmatrix} \mathbf{Y} & \mathbf{X} \\ \mathbf{X}^\top & \mathbf{Z} \end{bmatrix} \succeq 0$$

and Tr($\mathbf{Y}$) + Tr($\mathbf{Z}$) $\leq 2t$. *Proof:* *(Sketch) This can be proved by using the fact that the nuclear norm of a matrix is the dual of the L2-norm (not Frobenious norm). This means that ¡Still incomplete¿*

Using Theorem IV.10, the problem of finding a low cp-rank factorization of $\mathbf{A}$, can hence be cast as finding $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ such that the block matrix is PSD while constraining the rank of $\mathbf{X}$. The optimization problem is given as follows:

$$\min_{\mathbf{X}\in\mathbb{R}^{m\times k}} \mathsf{loss}(\mathbf{A}, \mathbf{X}) + \gamma(\mathrm{Tr}(\mathbf{Y}) + \mathrm{Tr}(\mathbf{Z}))$$

$$\text{s.t.} \quad \mathbf{Y} \succeq 0, \mathbf{Z} \succeq 0, \begin{bmatrix} \mathbf{Y} & \mathbf{X} \\ \mathbf{X}^\top & \mathbf{Z} \end{bmatrix} \geq 0$$

1) Algorithms: Noe this problem could be solved by using a projected gradient descent method. For example, in the symmetric case, we can cast this problem as the following optimization:

$$\min_{\mathbf{X}} ||\mathbf{A} - \mathbf{X}\mathbf{X}^T||_F \quad \text{s.t.} \quad \mathbf{X} \geq 0$$

I. Experiments

In this section, we reproduce the graph partitioning/clustering experiment from the paper [10] which we talked about in the previous few sections. We first define the graph partitioning problem and then put in some of the results.

Let $\mathbf{A} \in \{0,1\}^{n\times n}$ be the adjacency matris of a graph G such that $\mathbf{A}_{ij} = 1$ if the edge (i,j) is present in G, else 0. We want to partition the graph into k clusters while minimizing the number of edges between clusters and maximizing the number of edges inside a cluster. In order to do that, we seek an indicator matrix $\mathbf{X} \in \{0,1\}^{n\times k}$ such that $\mathbf{X}_{ik}$ is 1 if the node i is in cluster C_k, and 0 otherwise. This problem can be formulated as:

$$\min ||\mathbf{A} - \mathbf{X}\mathbf{X}^T|| \quad \text{s.t.} \quad \mathbf{X}\mathbf{1} = \mathbf{1}$$

This is an NP-hard optimization problem (as it is combinatorial) and hence, we relax it to the symmetric NMF approximation we discussed in Section 4.7, to get the following problem:

$$\min_{\mathbf{X}} ||\mathbf{A} - \mathbf{X}\mathbf{X}^T|| \quad \text{s.t.} \quad \mathbf{X} \geq 0$$

Once we have a continuous solution $\mathbf{X}*$, we can then get the indicator matrix by putting in 1 in the $\arg\max$ entry of each row in $\mathbf{X}*$ and setting all other entries to 0.

The performance of the optimization is measured in terms of how well A_{ij} can be predicted from X. Formally, the authors define the performance as:

$$\mathsf{perf}(C) = 1 - \frac{\#\left\{(i,j)|\mathbf{A}_{ij} \neq \sum_{l=1}^{k} \mathbf{X}_{il}\mathbf{X}_{lj}\right\}}{n^2}$$

Data Generation: We followed the same approach as the authors [10] for the data generation process. We generated random matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ – with **extra** choices apart from the random generation of only symmetric matrices that the authors tried their approach on. We tried their projected gradient descent algorithm on also non-symmetric matrices and we report our numbers in this case. The matrices are generated using this rule:

$$\mathbf{M} = \begin{bmatrix} 1_{A_{ij} \geq \alpha} & 1_{C_{ij} \geq \beta} \\ 1_{C_{ij} \geq \beta} & 1_{B_{ij} \alpha} \end{bmatrix}$$

Noe that this also tests how good the symmetric approximation works on non-symmetric data matrices, which is more as compared to the base paper.

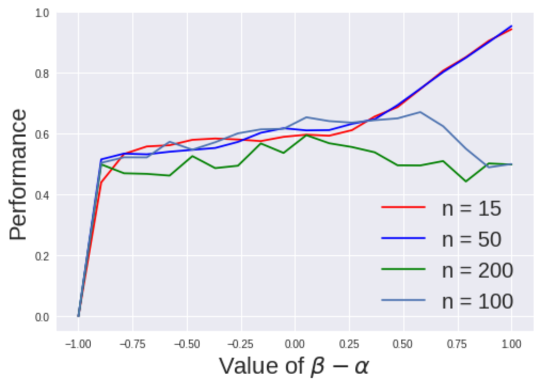

Fig. 2. Performance vs. the value of $\beta - \alpha$ in the case of graphs of different sizes, the value of $\beta - \alpha$ controls the separability of the graph into 2 clusters.

Results: We compared the performance of the projected gradient descent algorithm in terms of first of all, a trend of performance vs. the ambiguity of existence of 2 distinct clusters, this value is reflected in the value of $\beta - \alpha$. We find that for small sized graphs, the performance increases almost monotonically as the value of $\beta - \alpha$ increases which makes the graph more easily clusterable into two clusters. However, for large sized graphs their algorithm doesn't work very well. The number of projected gradient descent steps were fixed to 1000 for each of these cases. The plot is shown in Figure 2.

Regarding the Number of steps vs Number of nodes trend, we used Adam gradient descent algorithm [9] and hence, within a number of steps ($\sim$ 10), we found that the algorithm nearly converged to the convergent value. This is hence different from the results reported in the paper where they use vanilla descent. Adam also avoids the suboptimal solutions to the best extent.

We provide some samples of real adjacency matrix and generated adjacency matrix to show how good the learned approximation is. This is shown in Figure as heatmaps of the true and predicted adjacency matrices.

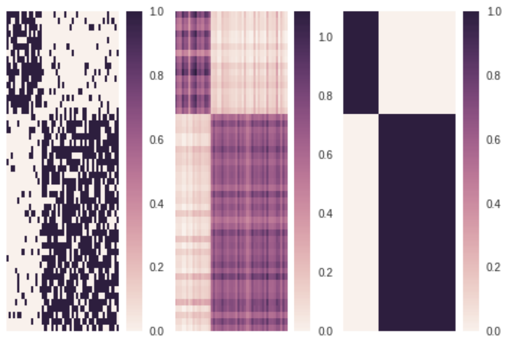

Fig. 3. A sample predicted from the model, Left: The original adjacency matrix, Middle: The predicted soft adjacency matrix, Right: Predicted Hard adjacency matrix.

V. Nonnegative Matrix Underapproximation

Sparsity is important in many applications: for example, the sparse information extracted from image data can be explained as the minor variations between two similar images, which will be analyzed in our later experiment section. Although solutions of NMF typically display some level of sparsity, sometimes it is still not enough. In [7] the authors introduced an approach to solve NMF problem based on the use of an underapproximation technique and show its effectiveness to obtain sparse solutions. The definition of NMU is shown as follows:

Definition V.1. *Given* $\mathbf{M} \in \mathbb{R}_{+}^{m\times n}$ *and* $1 \leq r < \min(m,n)$*, the* ***Nonnegative Matrix Underapproximation*** *(NMU) problem is defined as:*

$$\min_{\substack{\mathbf{V}\in\mathbb{R}^{m\times r},\\ \mathbf{W}\in\mathbb{R}^{r\times n}}} ||\mathbf{M} - \mathbf{VW}||_F$$

$$\textit{s.t.} \quad \mathbf{V} \geq 0, \mathbf{W} \geq 0, \ \textit{and} \ \mathbf{VW} \leq \mathbf{M}.$$

A. Sparsity of NMU

Intuitively, NMU can generate sparser solutions, because the constraint $\mathbf{VW} \leq \mathbf{M}$ pushed more elements in the matrices being closer to 0. It can also be explained based on the following example: since the zero entries of $\mathbf{M}$ can only be underapproximated by zeros, we have

$$\mathbf{M}_{ij} = 0 \rightarrow (\mathbf{VW})_{ij} = 0 \rightarrow \mathbf{V}_{ik} = 0 \text{ or } \mathbf{W}_{kj} = 0,$$

which shows that when the input matrix is sparse, many components of the NMU factors will have to be equal to zero. In the paper, the authors define the sparsity of a matrix $\mathbf{M}$ as

$$s(\mathbf{M}) = \frac{\#zeros(\mathbf{M})}{mn} \in [0,1],$$

and showed several theorems linking the sparsity of the decomposed matrices and the sparsity of the input matrix. First, they show that when the rank of the decomposition is equal to 1, the sum of the sparsity of the two decomposed matrices is larger than that of the input matrix, which is shown in Theorem V.2,

Theorem V.2. *For any nonnegative rank-one underapproximation $(v, w) \in \mathbb{R}^m_+ \times \mathbb{R}^n_+$ of $\mathbf{M} \in \mathbb{R}^{m\times n}_+$, $s(v) + s(w) \geq s(\mathbf{M})$.*
Proof: For a rank-one matrix $vw^\top$, the number of nonzeros is exactly equal to the product of the number of nonzeros in v and w, which means that $(1 - s(vw^\top)) = (1 - s(v))(1 - s(w))$, which implies $s(vw^\top) = s(v) + s(w) - s(v)s(w) \leq s(v) + s(w)$. Since underapproximation $vw^\top$ satisfies $0 \leq vw^\top \leq \mathbf{M}$, it must have more zeros than $\mathbf{M}$ and we have $s(\mathbf{M}) \leq s(vw^\top) \leq s(v) + s(w)$, proving the claim.

Using Theorem V.2, it is easy to prove that for the rank-r decomposition, the sum of the sparsity of the two decomposed matrices is larger than that of the input matrix.

Theorem V.3. *For any nonnegative underapproximation $(\mathbf{V}, \mathbf{W}) \in \mathbb{R}^{m\times r}_+ \times \mathbb{R}^{r\times n}_+$ of $\mathbf{M} \in \mathbb{R}^{m\times n}_+$, for each factor*

$$s(\mathbf{V}_{:k}) + s(\mathbf{W}_{k:}) \geq s(\mathbf{R}_k) \geq s(\mathbf{M}), 1 \leq k \leq r,$$

and $s(\mathbf{V}) + s(\mathbf{W}) \geq s(\mathbf{M})$.
Proof: Because $0 \leq \mathbf{V}_{:k}\mathbf{W}_{k:} \leq \mathbf{R}_k \leq \mathbf{M}$, which implies by the previous theorem the first set of inequalities. Observing that $s(\mathbf{V}) = \frac{1}{r}\sum_k s(\mathbf{V}_{:k})$ and $s(\mathbf{W}) = \frac{1}{r}\sum_k s(\mathbf{W}_{k:})$ is sufficient to prove the second inequality.

B. Algorithm for NMU based on Lagrangian relaxation

It is proved in the literature that same as NMF, NMU is a NP-hard problem and we can not expect to solve it up to guaranteed global optimality in a polynomial computational time. In the literature, the authors propose a nonlinear optimization scheme based on Lagrangian relaxation in order to compute approximate solutions of NMU.

The procedure is as follow: after writing the underapproximation constraints $\mathbf{VW} \leq \mathbf{M}$ of NMU into the objective function with the corresponding Lagrange multipliers $\mathbf{\Lambda} \in \mathbb{R}^{m\times n}_+$, we obtain the Lagrangian function $\mathcal{L}(\mathbf{V}, \mathbf{W}, \mathbf{\Lambda})$

$$\mathcal{L}(\mathbf{V}, \mathbf{W}, \mathbf{\Lambda}) = \frac{1}{2}||\mathbf{M} - \mathbf{VW}||^2_F + \text{Tr}(\mathbf{\Lambda}^\top(\mathbf{VW} - \mathbf{M})),$$

Lagrangian duality tells us that:

$$\min_{\mathbf{V},\mathbf{W}\geq 0} \sup_{\mathbf{\Lambda}\geq 0} \mathcal{L}(\mathbf{V}, \mathbf{W}, \mathbf{\Lambda}) \geq$$
$$\sup_{\mathbf{\Lambda}\geq 0} \min_{\mathbf{V},\mathbf{W}\geq 0} \mathcal{L}(\mathbf{V}, \mathbf{W}, \mathbf{\Lambda}) = \sup_{\mathbf{\Lambda}\geq 0} f(\mathbf{\Lambda}).$$

A general solution technique consists in repeatedly applying the following two steps:

1) Given multipliers $\mathbf{\Lambda}$, compute $(\mathbf{V}, \mathbf{W})$ to (approximately) minimize $\mathcal{L}(\mathbf{V}, \mathbf{W}, \mathbf{\Lambda})$;
2) Given solution $(\mathbf{V}, \mathbf{W})$, update multipliers $\mathbf{\Lambda}$.

One approach which does follow this two-step procedure is **Dual Gradient Descent**. As to the first step, because $\mathcal{L}(\mathbf{V}, \mathbf{W}, \mathbf{\Lambda})$ can also be written as follows,

$$\mathcal{L}(\mathbf{V}, \mathbf{W}, \mathbf{\Lambda}) = \sum_{i,j} \frac{1}{2}(\mathbf{M} - \mathbf{VW})^2_{ij} + \sum_{i,j} \Lambda_{ij}(\mathbf{VW} - \mathbf{M})_{ij}$$
$$= \frac{1}{2}||\mathbf{M} - \mathbf{\Lambda} - \mathbf{VW}||^2_F - \frac{1}{2}||\mathbf{\Lambda}||^2_F,$$

which shows that minimizing $\mathcal{L}(\mathbf{V}, \mathbf{W}, \mathbf{\Lambda})$ for a fixed $\mathbf{\Lambda}$ is equivalent to minimizing $||\mathbf{M} - \mathbf{\Lambda} - \mathbf{VW}||^2_F$. The problem of finding $\mathbf{V} \geq 0$ and $\mathbf{W} \geq 0$ such that $\mathbf{N} \approx \mathbf{VW}$ is a Nonnegative Factorization (NF) problem, with the input matrix having no sign constraint. It can be formulated as

$$\min_{\substack{\mathbf{V}\in\mathbb{R}^{m\times r},\\ \mathbf{W}\in\mathbb{R}^{r\times n}}} ||\mathbf{N} - \mathbf{VW}||^2_F \text{ such that } \mathbf{V} \geq 0 \text{ and } \mathbf{W} \geq 0,$$

with $\mathbf{N} \in \mathbb{R}^{m\times n}$ and $1 \leq r < \min(m, n)$ and is shown to be NP-hard for any factorization rank.

Some standard algorithms for NMF, such as Hierarchical Alternating Least Squares (HALS), can easily adapted to handle the NF problem. In this literature, they use the HALS and alternatively updates each column of $\mathbf{V}$ and each row of $\mathbf{W}$ with the following rules:

$$\mathbf{V}^*_{:k} = \text{argmin}_{\mathbf{V}_{:k}\geq 0}\|(\mathbf{M} - \mathbf{\Lambda}) - \mathbf{VW}\|_2,$$
$$\mathbf{W}^*_{k:} = \text{argmin}_{\mathbf{W}_{k:}\geq 0}\|(\mathbf{M} - \mathbf{\Lambda}) - \mathbf{VW}\|_2.$$

The second step for the algorithm presented in the paper is to update $\mathbf{\Lambda}$ in order to find better solutions for the Lagrangian dual problem. Considering that the complementary slackness must be satisfied, if $(\mathbf{M} - \mathbf{VW})_{ij} > 0$, $\mathbf{\Lambda}_{ij}$ should be decreased and eventually reach zero. If $(\mathbf{M} - \mathbf{VW})_{ij} < 0$, $\mathbf{\Lambda}_{ij}$ should be increased to try to get a feasible solution with $(\mathbf{M} - \mathbf{VW}) > 0$. The algorithm is presented in Algorithm 2.

The authors discussed different implementations of L-NMU: one way is to apply L-NMU directly to the rank-r problem, which is also called *Global-NMU* (G-NMU), and the other way is to run algorithm L-NMU successively r times to compute r rank-one approximations, with each approximation from the input matrix subtracted before computing the next one. This method is called *recursive-NMU* (R-NMU).

```
Algorithm 2: Lagrangian NMU (L-NMU)
Input  : $\mathbf{M} \in \mathbb{R}_+^{m\times n}$, $r > 0$, $\mathbf{V} \in \mathbb{R}_+^{m\times r}$,
         $\mathbf{W} \in \mathbb{R}_+^{r\times n}$, maxiter, $T$.
Output: $\mathbf{V}$, $\mathbf{W}$.
1 $\mathbf{\Lambda} = 0$
2 for $k$ = 1 : maxiter do
3     Update $(\mathbf{V}, \mathbf{W})$ using $T$ iterations of HALS
4     Update $\mathbf{\Lambda} \leftarrow \max(0, \mathbf{\Lambda} - \frac{1}{k}(\mathbf{M} - \mathbf{VW}))$
5 end
```

C. Experimental results

We did experiments on both G-NMU and R-NMU based on the Kuls image dataset, which is the dataset used in the reference and consists of 20 images (64×64 pixels) of a face illuminated from many directions with a moving light source. The images are very similar and most of the information can be expressed with only one factor. The remaining information, which is the variation between these similar images, resides in the different orientations of the lighting, and well-performed decomposition method needs to extract the information of both the similar parts and the variational parts among these images. Decomposition results with the same settings as that in the reference for a rank-5 factorization are given by Figure 4 and Table V-C. Same as the results in the literature, we observe that even though NMF is able to extract several faces with different lighting orientations, their results are not sparse and they do not extract the minor variations in these pictures. G-NMU is much a little better than NMF; we can see from Table V-C that they produces larger sparsity than the NMF. R-NMU produces the best result and it first extracts a face that contains almost all the information, and then complementary parts representing different orientations of the lighting. This shows that R-NMU, which uses both the recursive extraction algorithm and the underapproximation constraints, gives a better result on the sparsity demand.

	Error	$s(\mathbf{V})$	$s(\mathbf{W})$
NMF	10.07	0.0	0.0
G-NMU	11.32	0.047	0.12
R-NMU	14.40	0.41	0.47

TABLE II
COMPARISON OF THE RELATIVE APPROXIMATION ERROR AND SPARSITY FOR THE KULS IMAGE DATASET.

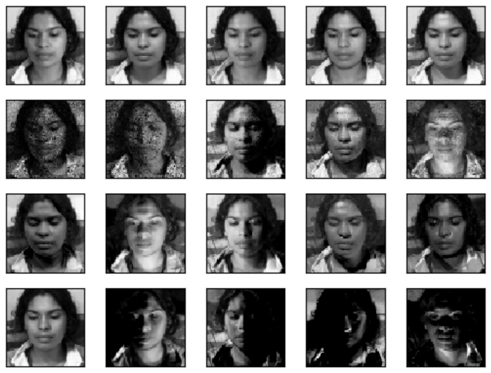

Fig. 4. Basis for the Kuls image dataset, from top to bottom: original images, NMF, G-NMU, R-NMU. The first line shows 5 of the 20 images in the original dataset. All the other images are from the decomposed matrices $\mathbf{W}$, where the ith image from left to right is the information in the ith row of $\mathbf{W}$. The dark pixels in the images show that the corresponding element in $\mathbf{W}$ is approaching 0. From the images, we can see that the images in the final row have a lot of black pixels, meaning that R-NMU produce more sparse results that others.

REFERENCES

[1] Gautam Das and Michael T. Goodrich. On the complexity of approximating and illuminating three-dimensional convex polyhedra (preliminary version). In WADS, volume 955 of Lecture Notes in Computer Science, pages 74–85. Springer, 1995.

[2] Chris HQ Ding, Tao Li, and Michael I Jordan. Convex and semi-nonnegative matrix factorizations. IEEE transactions on pattern analysis and machine intelligence, 32(1):45–55, 2010.

[3] Mirjam Dür. Copositive programming – a survey. In Moritz Diehl, Francois Glineur, Elias Jarlebring, and Wim Michiels, editors, Recent Advances in Optimization and its Applications in Engineering. Springer Berlin Heidelberg, 2010.

[4] Cédric Févotte and Jérôme Idier. Algorithms for nonnegative matrix factorization with the beta-divergence. CoRR, abs/1010.1763, 2010.

[5] Christodoulos A. Floudas. Deterministic Global Optimization: Theory, Methods and (NONCONVEX OPTIMIZATION AND ITS APPLICATIONS Volume 37) (Nonconvex Optimization and Its Applications). Springer-Verlag, Berlin, Heidelberg, 2005.

[6] Nicolas Gillis. Introduction to nonnegative matrix factorization. arXiv preprint arXiv:1703.00663, 2017.

[7] Nicolas Gillis and François Glineur. Using underapproximations for sparse nonnegative matrix factorization. Pattern recognition, 43(4):1676–1687, 2010.

[8] Ronan Hamon, Valentin Emiya, and Cédric Févotte. Convex nonnegative matrix factorization with missing data. In Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th International Workshop on, pages 1–6. IEEE, 2016.

[9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. CoRR, abs/1412.6980, 2014.

[10] V. Krishnamurthy and A. d'Aspremont. Convex Algorithms for Nonnegative Matrix Factorization. ArXiv e-prints, July 2012.

[11] Nikolaos Vasiloglou, Alexander G. Gray, and David V. Anderson. Non-negative matrix factorization, convexity and isometry. CoRR, abs/0810.2311, 2008.

[12] Stephen A. Vavasis. On the complexity of nonnegative matrix factorization. CoRR, abs/0708.4149, 2007.