NONNEGATIVE MATRIX FACTORIZATION

April 29, 2019

Author: Maxim Bolshakov

Supervisor: Dmitry Chistikov

Year of study: 3

Abstract

This project explores the problem of nonnegative matrix factorization (NMF). Given a nonnegative matrix V, it is a problem of finding matrices W, H such that V is equal to, or is approximated by the product of W and H.

This problem has a wide range of applications. First, this report explores the applications of NMF in greater details. Then, the complexity analysis of NMF is presented. For the part which focuses on complexity, I have produced a number of proofs which clarify certain points from the related works on the complexity of NMF, which were either missed or unclear. In addition, I have created Python scripts which illustrate certain theoretical ideas related to the complexity of NMF. Finally, approximation algorithms are analysed. For this project, several NMF were chosen to be implemented in Python. The performance of these algorithms is evaluated both on the random and real datasets. The benefit of running the algorithms on a GPU was analysed as well.

This work can be useful for people who work on the complexity of NMF and related problems, as well as for people who work on the practical applications of NMF.

Keywords:

Linear algebra, NMF, matrix factorization, universality theorem, the existential theory of the reals, complexity.

Contents

1 Introduction

This project focuses on the problem of nonnegative matrix factorization (NMF).

Nonnegative matrix factorization refers to the decomposition of $V \in \mathbb{R}_+^{m \times n}$ into $W \in \mathbb{R}_+^{m \times k}$, $H \in \mathbb{R}_+^{k \times n}$, such that V is equal to or approximated by the product of W and H. Alternatively, nonnegative matrix factorization can refer to the decomposition of V into the sum of k nonnegative matrices $V_1 \dots V_k$. Such factorization can be referred to as a factorization of size k.

It is easy to see that one representation can be transformed into the other. Given W, H, each V_i term from the corresponding sum of rank 1 matrices is equal to $W_{:i}H_{i:}$, the product of the i^{th} column of W and the i^{th} row of H.

The nonnegative rank of $V \in \mathbb{R}_+^{m \times n}$, denoted $rank_+(V)$, is the smallest number k such that there exist 2 nonnegative matrices $W \in \mathbb{R}_+^{m \times k}$, $H \in \mathbb{R}_+^{k \times n}$, such that $V = WH$. Equivalently, $rank_+(V)$ is the smallest k such that there exist k rank 1 nonnegative matrices $V_1 \dots V_k$ such that $V = \sum_{i=1}^{k} V_i$.

For the practical applications of the NMF problem, we are interested in finding W and H such that WH approximates V. The quality of the approximation is assessed by some function which takes V and WH as 2 parameters. This function can be different from task to task, but the two most commonly used are Frobenius Norm, which equals to $\sqrt{\sum_{i,j}(V - WH)_{i,j}^2}$, and Kullback-Leibler divergence function, which is equal to $\sum_{i,j}(V_{ij}log\frac{V_{ij}}{(WH)_{ij}} - V_{ij} + (WH)_{ij})$. The nonnegativity restriction often leads to a straightforward interpretation of the results.

2 Applications

This section presents some background information on the application of NMF to various practical and theoretical tasks.

2.1 Expectation Maximisation [Kubjas et al., 2015]

For the analysis of a joint distribution of 2 random variables X and Y, it is possible to use a model where (X, Y) has a distribution of a mixture of r joint distributions, and for each underlying distribution, X and Y are conditionally independent. The correlation in observations is explained by the underlying choice of one of the r distributions form the mixture. The intuition is that the common cause should factor out the observed correlations. Let Z be the hidden random variable, corresponding to the choice of the joint distribution form the mixture.

Consider stochastic matrices A, B and L, such that the k^{th} column of A gives the distribution of $X|Z = k$, the k^{th} row of B gives the distribution of $Y|Z = k$, and L is a diagonal matrix which entries give the distribution of Z. Then the distribution of (X, Y) is given by $P = ALB$, where P_{ij} is the probability to observe $X = i$ and $Y = j$. P can be estimated from observations, and, if one is interested in A, L, B, it is possible to use nonnegative matrix factorization to estimate them from P. Statistical expectation maximisation, in this case, is essentially the NMF algorithm from [Lee and Seung, 1999].

In quantum mechanics, one can be interested if a common cause can be used to explain the results of an experiment. If the results can be described by the mixture model above, then the $rank_+(P)$ is the smallest size of the support of Z for which common cause explanation is possible [Suppes and Zanotti, 1981] [Cohen and Rothblum, 1993].

2.2 Text Analysis [Ding et al., 2008]

Probabilistic latent semantic analysis(PLSA) is a statistical technique to analyse text documents [Hofmann, 1999]. It matches the model described in the expectation maximisation section, where a hidden variable Z is a topic, X and Y represent the occurrence of the document(observing the i^{th} of the given documents is treated as a distinct outcome $X = i$) and the occurrence of the word respectively. X and Y are assumed to be independent given the topic.

Let $V = V_{ij}$ be the word-to-document matrix: V_{ij} is the frequency of word w_i in document d_j. It was shown in [Ding et al., 2008] that the NMF error minimisation algorithm described in [Lee and Seung, 2001a] and PLSA optimise the same objective function.

It coincides with the connection between expectation maximisation, mixture distribution models and NMF which was described in [Kubjas et al., 2015].

2.3 Combinatorial Optimisation [Yannakakis, 1991]

For a polytope P in the n-dimensional space, with f facets and v vertices, slack matrix S is an f by v nonnegative matrix such that $S_{ij} = b_i a_i^T v_j$, where v_j is the j^{th} vertex and a_i, b_i are the terms of an inequality corresponding to the i^{th} facet, $a_i^T x \leq b_i$. Then if m =$rank_+(SK)$, then the minimal number of (values + constraints) over all linear programs which have P as a set of feasible solutions is $\Theta(m + n)$

2.4 Nested Polytope Problem [Gillis and Glineur, 2012]

Restricted nonnegative matrix factorization(RNMF) is a constrained version of NMF, which requires that for the factorization $V = WH$ the column space of V and W coincide. Restricted nonnegative rank $rrank_+(V)$ is defined as the smallest possible number of columns in W(and rows in H).

RNMF problem: Given a matrix M and a natural number k, is restricted nonnegative rank of M less or equal to k?

Nested polytopes problem (NPP): Given a full-dimensional bounded polyhedron P described by m inequalities in $\mathbb{R}^{r-1}$, $P = \{x : Ax + b \geq 0\}$, and a set S of n points, $S \subseteq P$ such that convex hull of S is also full-dimensional, find the minimum number

k of points in P whose convex hull T contains convex hull of S ($conv(S) \subseteq T \subseteq P$).

The RNMF problem can be reduced to the NPP problem in polynomial time, and vice versa:

1. Given $M \in \mathbb{Q}_{+}^{m\times n}$ of rank r, we can in polynomial time compute m rational points which span polytope S, compute $A \in \mathbb{Q}^{n\times(r-1)}$ and $b \in \mathbb{Q}^{n}$ such that the polytope $P = \{x : Ax + b \geq 0\}$ is full dimensional and $S \subseteq P$.
2. Given $A \in \mathbb{Q}^{n\times(r1)}$ and $b \in \mathbb{Q}$ such that $P = \{x : Ax + b \geq 0\}$ is a full-dimensional polytope, and given a full-dimensional polytope $S \subseteq P$ spanned by m points $s_1 \ldots s_m \in \mathbb{Q}r - 1$, $[s_1 \ldots s_m] \in \mathbb{Q}^{r-1\times m}$, we in polynomial time can construct matrix $M = A[s_1 \ldots s_m] + b$.

I both of the cases above, it is possible to show that the restricted nonnegative rank of M is less or equal to d if and only if there exist d points which span polytope T such that $S \subseteq T \subseteq P$.

2.5 Probabilistic Automata [Chistikov et al., 2016]

A labelled Markov chain (LMC) is a tuple $\mathcal{M} = (n, \Sigma, \mu)$ where n is the number of states, Σ is a finite alphabet of labels, and function μ specifies the transition matrices such that when $\mathcal{M}$ is in state i, it emits label σ and moves to state j, with probability $\mu(\sigma)_{i,j}$

For the state i and for $w \in \Sigma$, we write $p_i^{\mathcal{M}}(w)$ for the probability that, starting in state i, $\mathcal{M}$ emits word w. Similarly, we can label the probability that, starting with the initial distribution π, $\mathcal{M}$ emits word w as $p_\pi^{\mathcal{M}}(w)$.

We say that an LMC $\mathcal{M}$ is covered by an LMC $\mathcal{M}_0$, written as $\mathcal{M}_0 \geq \mathcal{M}$, if for every initial distribution π for $\mathcal{M}$ there exists a distribution π_0 for $\mathcal{M}_0$ such that $p_\pi^{\mathcal{M}}(w) = p_{\pi_0}^{\mathcal{M}_0}(w)$.

The backward matrix of $\mathcal{M}$ is a matrix $Back(\mathcal{M}) \in \mathbb{R}_+^{n \times \Sigma^*}$ where $(Back(\mathcal{M}))_{i,w} = p_i^{\mathcal{M}}(w)$. (It is infinite, but since it has n rows, its rank is at most n)

Given nonnegative matrix M in $\mathbb{Q}^{n \times m}$ of rank r, one can compute in polynomial time a labelled Markov chain(LMC) $\mathcal{M}$ with $m+2$ states such that:

1. Any restricted factorization of M of size d in a form $M = WH$ determines another labelled Markov chain $\mathcal{M}'$ with $d+2$ states s.t. $\mathcal{M}$ is covered by $\mathcal{M}'$ and $rank(Back(\mathcal{M}')) = rank(W) + 2$

2. Any $\mathcal{M}'$ with $d+2$ states which covers $\mathcal{M}$ determines a restricted factorization of size d, $M = WH$ such that $rank(Back(\mathcal{M}')) = rank(W) + 2$.

In particular, for all d the inequality $rrank_+(M) \leq d$ holds if and only if $\mathcal{M}$ is covered by some $(d+2)$-state LMC $\mathcal{M}_0$ such that $Back(\mathcal{M}_0)$ and $Back(\mathcal{M})$ have the same rank.

2.6 Computational Biology [Devarajan, 2008]

Gene expression data from a set of experiments is typically presented as a matrix in which the rows correspond to expression levels of genes, the columns to samples (which may represent distinct tissues, experiments, or time points), and each entry to the expression level of a given gene in a given sample. For gene expression studies, the number of genes, p, is typically in the thousands; the number of samples, n, is typically less than 100.

For a gene expression matrix $V \in \mathbb{R}_+^{p \times n}$, consisting of observations on p genes from n samples.

Consider the NMF $V = WH$. Each column of W defines a metagene (a pattern of gene expression), and each column of H represents the metagene expression pattern of the corresponding sample. NMF can be used to find a small number of metagenes which can model the whole gene expression data. The whole gene data is then expressed as a nonnegative linear combination of metagene patterns, and columns of H represent gene samples in the space defined by metagenes.

2.7 Image Analysis [Lee and Seung, 1999]

Consider an image database as a matrix $V \in \mathbb{R}_+^{n \times m}$, each column of which contains n non-negative pixel values of one of the m facial images. Consider nonnegative factorisation of V, $V = WH$. The r columns of W are called basis images and represent the features extracted from the database. Each column of H consists of the coefficients which represent a face as a linear combination of basis images.

Compared to principal components analysis (PCA) and vector quantization (VQ), the features extracted by NMF correspond better with intuitive notions of the parts of faces.

2.8 Clustering [Kim and Park, 2008b]

If a dataset is represented as a nonnegative matrix V, where columns of V correspond to data entries, then the nonnegative factorization $V = WH$ can be used to assign each data entry corresponding to the i^{th} column to the cluster $argmax_k(H_{ki})$. It was shown that, when the sparsity of W and H is enforced, the performance of such clustering is better compared to the K-means method.

2.9 Privacy-preserving Data Mining [Wang et al., 2006]

One of the ways to preserve data privacy is to transform original data into protected, publishable data by introducing distortions. If the data can be represented as nonnegative matrix V, then it is possible to run the following algorithm to achieve data distortion.

Starting from some $r = k$, to $r = 1$, compute $V^{(r)} = WH$, with the inner dimension r. Then choose $V^{(r)}$ which satisfies privacy and accuracy conditions. In particular, if original data entries have some class labels, an accuracy metric can be the accuracy of the classifier trained on the distorted data.

2.10 Audio Data Analysis [Févotte et al., 2009]

Given the audio data in the time domain, the short-time Fourier transform X of the audio data can be computed. X can be represented as a nonnegative matrix, where frequency changes over rows and time changes over columns. Consider a nonnegative matrix of squared entries $V = |X|^2$. For the NMF $V = WH$, W represents a dictionary of power distribution over frequencies for different sources, H represents their mixture at each time step. A particular non-standard loss function, called Itakura-Saito divergence, is used for this task.

2.11 Community Detection [Ozer et al., 2016]

Nonnegative matrix factorisation can be applied to social media data for the task of community detection.

Data of the frequencies at which users use different words, hashtags and link domains are represented by matrices X_{uw}, X_{uh}, X_{ud}. Matrix U represents the assignments of users into communities, W, H, D represent the words, hashtags and domains used by these communities. The task to assign users into communities can be represented as the task to approximate $[X_{uw}, X_{uh}, X_{ud}]$ by $U[W^T, H^T, D^T]$, minimising a custom cost function. ([ ] correspond to a row-wise concatenation)

User i is assigned to the community j where $j = argmax_j(U_{ij})$ Users from community j are more likely to use words, hashtags and domains corresponding to the largest values of column j in W, H, D respectively.

3 Complexity Analysis

3.1 Introduction

To discuss the complexity of the nonnegative matrix factorization, we first need to introduce the concept known as the existential theory of reals($\exists\mathbb{R}$).

The existential theory of the reals ($\exists\mathbb{R}$) is the set of all true sentences of the form $\exists x_1 \dots x_n \ s.t. \ F(x_1 \dots x_n)$, where F is a well-formed quantifier-free formula(QFF) involving equalities and inequalities of real polynomials.

For the complexity analysis, consider an $\exists\mathbb{R}$ decision problem, which is a problem of deciding, given a sentence, if this sentence belongs to $\exists\mathbb{R}$ set (if the given formula is satisfiable in $\mathbb{R}$). The $\exists\mathbb{R}$ complexity class is then defined as a set of all decision problems which have a polynomial time reduction to $\exists\mathbb{R}$ decision problem.

NMF decision problem is a problem of deciding if for a given nonnegative matrix V and an integer k, $rank_+(V) \leq k$. It is straightforward to see that this problem is a particular instance of the $\exists\mathbb{R}$-decision problem.

Determining the nonnegative rank of $V \in \mathbb{R}^{m\times n}$ can be reduced to $O(log(min(m,n)))$ instances of the NMF decision problems. It is straightforward that for V, $rank(V) \leq rank_+(V) \leq min(m,n)$. Hence, to compute the nonnegative rank of V it suffices to check for $k = rank(V), rank(V)+1, \dots, min(m,n)$ whether $rank_+(V) \leq k$. By using a bisection procedure, the number of required tests is of the order $log(min(m,n) - rank(V))$. [Cohen and Rothblum, 1993]

The above problem is complete for the $\exists\mathbb{R}$ complexity class. The proof of this fact was presented in [Shitov, 2016]. However, it would be beneficial to discuss and clarify the proof, as certain steps are either skipped or hard to understand.

Using results from [Matousek, 2014] and [Grigor'ev and Vorobjov, 1988], it is possible to show that the $\exists\mathbb{R}$-decision problem remains complete for the $\exists\mathbb{R}$ complexity class if F is restricted to one polynomial, presented as a sum of monomials, and roots are restricted to a unit cube $[0, 1]^n$. While the ideas form [Matousek, 2014] and [Grigor'ev and Vorobjov, 1988] may be complex, assuming their correctness, the conclusion that $\exists\mathbb{R}$ decision problem can be restricted and still remain $\exists\mathbb{R}$-complete is quite straightforward. This conclusion is presented at the end of [Shitov, 2016].

It follows that to prove the fact that NMF decision problem is $\exists\mathbb{R}$-complete, it is enough to show that the restricted $\exists\mathbb{R}$ decision problem reduces to it.

3.2 Important Concepts

To discuss the proof, first, several concepts should be introduced.

The [Shitov, 2016] introduced 2 important concepts, the incomplete matrix and the strong equivalence between subsets $U \in \mathbb{R}^n$ and $V \in \mathbb{R}^m$. Another important concept, which was introduced in a different paper [Cohen and Rothblum, 1993] is the concept of independent elements. First, these concepts must be explained.

3.2.1 Independent Elements

For a matrix $A = (a_{ij})$, two elements $a_{i_1j_1}, a_{i_2j_2}$ are independent if and only if $a_{i_1j_1} \cdot a_{i_2j_2} > 0$ and $a_{i_1j_2} \cdot a_{i_2j_1} = 0$.

It was stated in [Cohen and Rothblum, 1993] without a proof that for a nonnegative matrix, $rank_+(A)$ is bounded below by the size of a set of pairwise independent elements of A. Later in this work, I will provide the proof and expand the concept of independent elements to independent sets of elements.

3.2.2 Strong Equivalence

Consider two sets, $U \in \mathbb{R}^n$ and $V \in \mathbb{R}^m$, $n \leq m$.

Let π be a natural projection $\mathbb{R}^m \to \mathbb{R}^n$. If there exists a set of rational functions $\phi_{n+1} \dots \phi_m$ such that π is invertible, $\pi^{-1}(u)$ is unique and equals to $(u_1 \dots u_n,$

$\phi_{n+1}(u) \dots \phi_m(u))$ $\forall u \in U$, then call this π a rational projection(a projection which can be inverted by rational functions).

A permutation function for $\mathbb{R}^n$ is defined as $(x_1 \dots x_n) \mapsto (x_{\sigma(1)} \dots x_{\sigma(n)})$, where σ is a permutation on n elements.

U and V are strongly equivalent(denoted $U \sim V$) if they belong to an equivalence relation generated by rational projections and permutations, i.e.

For $V \in \mathbb{R}^n$ $U \in \mathbb{R}^m$
$V \sim U \iff \exists f : U \mapsto V$
such that f is a bijection and
$f = g_1 \circ \dots \circ g_n$ where
$g_i \in \{$
rational functions, invertible by projections,
projections, invertible by rational functions,
permutations
$\}$

In simple terms, this means that and V and U are strongly equivalent if there is a 1-1 correspondence between elements in U and V which can be represented as a composition of projections, rational functions and permutations. In practice, this means that there is an algorithm to convert value in U to the corresponding value in V and vice versa. Below, the example of such sets is presented.

$$\{(x,y) : x, y \in [0,1]\} \sim \{(y, \frac{x+1}{y+1}, x) : x, y \in [0,1]\}$$

$$\{(x,y) : x, y \in [0,1]\} \xrightleftharpoons[\text{projection}]{\text{rational func}} \{(x, y, \frac{x+1}{y+1}) : x, y \in [0,1]\}$$

$$\{(x, y, \frac{x+1}{y+1}) : x, y \in [0,1]\} \xrightleftharpoons[\text{permutation}]{\text{permutation}} \{(y, \frac{x+1}{y+1}, x) : x, y \in [0,1]\}$$

3.2.3 Incomplete Matrix

Incomplete matrix is the matrix which may contain not only elements of $\mathbb{R}_+$ but also variables whose ranges are subsets of $\mathbb{R}_+$

Let A be such matrix; a *completion* of A is a matrix that can be obtained from A by assigning some value to every instance of each variable within the range of this variable. The factorization of the matrix A is the factorization of any of its possible completions.

Here is an example:

V is an incomplete matrix.

$$V = \begin{bmatrix} 1 & 1 & x \\ x & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \; x \in [0,2]$$

V_1, V_2 are the possible completions for $x = 0$, $x = 1$ respectively

$$V_1 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} V_2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

We can consider the set of factorizations of V of size k as $fact_+(V, k)$. It is possible to show that $rank_+(V_1) = 3$, so the set of factorizations of size 2 does not include the factorization of V_1, but the set of factorizations of size 3 will include factorizations of both V_1 and V_2

$$\left(\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \right) \in fact_+(V, 2)$$

$$\left(\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 0 & 0 & 0\end{bmatrix}, \begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 1 & 0 & 0\end{bmatrix}, \begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 1\end{bmatrix}\right) \in fact_+(V, 3)$$

$$\left(\begin{bmatrix}1 & 1 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}, \begin{bmatrix}0 & 0 & 0\\ 0 & 1 & 1\\ 0 & 0 & 0\end{bmatrix}, \begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 1 & 0 & 1\end{bmatrix}\right) \in fact_+(V, 3)$$

3.3 The Proof of the Complexity

Note, that for a matrix $V \in \mathbb{R}_+^{n \times m}$, the set of factorization into k nonnegative rank 1 matrices can be viewed as a set of vectors size $n \cdot m \cdot k$ in $\mathbb{R}^{n \cdot m \cdot k}$, if we consider a set of matrices unravelled into one vector. We can call such factorizations as the factorization of size k. In addition, we can restrict the factorizations such that the terms(rank 1 nonnegative matrices) are ordered in lexicographic order. This is useful because it removes the factorizations which are different only up to the order of the terms. Having defined the factorization in this way, we can talk about the set of factorizations being strongly equivalent to the roots of a certain polynomial, as both are the sets of real vectors.

3.3.1 Overview

Given a polynomial, it can be transformed into a set of simple conditions, such that the set of values which satisfies these conditions is strongly equivalent to the roots of the polynomial restricted to a unit cube. For each condition, it is possible to construct a nonnegative incomplete matrix whose set of factorizations(where we consider the terms of factorization to be in lexicographic order and presented as an unravelled vector) is strongly equivalent to the set of values which satisfy the conditions. These incomplete matrices can be combined into a block-diagonal incomplete matrix, whose factorization is strongly equivalent to the set of all values for which all conditions are satisfied. This block-diagonal incomplete matrix has a corresponding (complete) nonnegative matrix such that their factorizations are equivalent. Therefore, the set of roots of the polynomial is strongly equivalent to the set of factorizations of a particular nonnegative matrix. One set is empty if and only if the other is empty. NMF decision problem asks about the emptiness of one set, and the $\exists\mathbb{R}$ decision problem asks about the emptiness of the other. Given that transformation steps take polynomial time, restricted $\exists\mathbb{R}$ decision problem reduces to NMF decision problem and so NMF decision problem is complete for the $\exists\mathbb{R}$ complexity class.

Here is the scheme of the proof.

$$\boldsymbol{\{x_1 \dots x_n : P(x_1 \dots x_n) = 0, (x_1 \dots x_n) \in [0,1]^n\}}$$
$$\text{for polynomial } \boldsymbol{P}$$
$$\wr$$
$$\boldsymbol{\{x_1 \dots x_m : C_1(x_1 \dots x_m) \bigwedge \dots \bigwedge C_k(x_1 \dots x_m)\}}$$
$$\text{Where } \boldsymbol{C_i} \text{ is either}$$
$$\boldsymbol{x_a \in [0, \Sigma k_{b_i}], x_{b_i} \in [0,1]\ x_a = (k_0+) \sum_{i=1}^n k_i x_{b_i}}$$
$$\text{or } \boldsymbol{x_a, x_b, x_c \in [0,1]\ x_a = x_b * x_c}$$
$$\wr$$
$$\boldsymbol{fact_+(V_1, q_1) \oplus \dots \oplus fact_+(V_k, q_k)}$$
$$\text{where } \boldsymbol{V_i} \text{ are certain incomplete matrices}$$
$$\wr$$
$$\boldsymbol{fact_+(V, \sum q_i)}$$
$$\text{where } \boldsymbol{V} \text{ is a block-diagonal matrix with blocks } \boldsymbol{V_i}$$
$$\wr$$
$$\boldsymbol{fact_+(G, q')}$$
$$\text{where } \boldsymbol{G} \text{ is a variable-free matrix}$$

The proof can be illustrated by the chain of sets whose consequent strong equivalence is proven.

1. Under certain conditions, for an incomplete matrix A, there exists a (variable-free) matrix B such that, for particular integers q, q', we have $fact_+(A, q) \sim fact_+(B, q')$

2. Let incomplete matrices $M_1 \dots M_n$ be such that the nonnegative rank is bounded below by $r_1 \dots r_n$ and integers $r_1 \dots r_n$ are such that $fact_+(M_i, r_i) \sim \{x_1^{(i)} \dots x_{n_i}^{(i)} : C_i(x_1^{(i)} \dots x_{n_i}^{(i)})\}$, where $x_1^{(i)} \dots x_{k_i}^{(i)}$ correspond to variables from the incomplete matrix M_i, and C_i is a certain condition on these variables. It is possible to combine these matrices into a block-diagonal incomplete matrix M. The rank of this block-diagonal matrix is bounded below by $r := \sum_{i=1}^{n}(r_i)$.

   In this case, $fact_+(M, r) \sim \{(x_1^{(1)} \dots x_{k_1}^{(1)} \dots x_1^{(n)} \dots x_{k_n}^{(n)} : C_1(x_1^{(1)} \dots x_{k_1}^{(1)}) \wedge \dots \wedge C_n(x_1^{(n)} \dots x_{k_n}^{(n)})\}$. Under certain conditions, the first point can be applied to M.

3. It is possible to construct an incomplete matrix M such that $rank_+(M) \geq k$ for a constant k and $fact_+(M, k) \sim \{x_1 \dots x_n : C(x_1 \dots x_n)\}$, where C is one of the following conditions:

   $C(x, y, z) :=$ "$x = y \cdot z \wedge x, y, z \in [0, 1]$" or
   $C(L, x_1 \dots x_n) :=$ "$L = k_0 + \sum_{i=1}^{n} x_i k_i \wedge x_1 \dots x_n \in [0, 1] \wedge L \in [0, \sum k_i]$"
   Where k_i is a nonnegative constant

4. For a particular polynomial, it is possible to find a set of constraints of the form described in step 3, such that the set of values which satisfy the constraints is strongly equivalent to the set of roots of the polynomial in a unit cube.

The final step is straightforward. First, constraints which transform all products between 2 variables into new single variables are generated. After the generations of these constraints, the polynomial has a form of a linear combination of variables, with no products between individual variables. Finally, 2 constraints which correspond to the linear combinations with positive and negative coefficients are generated. These linear combinations are set to be equal to the new single variable. This implies that the values which satisfy these constraints are, at the same time, the roots of the polynomial, as in this case the "positive" and "negative" sides cancel each other.

It can be illustrated with the following example.

$$F(x,y,z,t) = xy + xz - 2xyz + 0.5 - t = 0 \wedge x,y,z,t \in [0,1]$$

$$\Leftrightarrow$$

$$u_1 = xy \wedge x,y,u_1 \in [0,1] \wedge$$

$$u_2 = xz \wedge x,z,u_2 \in [0,1] \wedge$$

$$u_{31} = xy \wedge x,y,u_{31} \in [0,1] \wedge$$

$$u_{32} = u_{31}z \wedge z,u_{31},u_{32} \in [0,1] \wedge$$

$$L = u_1 + u_2 + 0.5 \wedge u_1,u_2 \in [0,1] \wedge L \in [0,2.5]$$

$$L = 2u_{32} + t \wedge u_{32},t \in [0,1] \wedge L \in [0,3]$$

3.3.2 Independent Elements and Blocks

First, the statement about the independent elements which was stated in one of the previous parts must be proven.

For a matrix $A = (a_{ij})$, two elements $a_{i_1j_1}, a_{i_2j_2}$ are independent if and only if $a_{i_1j_1} \cdot a_{i_2j_2} > 0$ and $a_{i_1j_2} \cdot a_{i_2j_1} = 0$.

Let $S = \{a_{i_1j_1} \dots a_{i_kj_k}\}$ be a set of mutually independent entries of a nonnegative matrix A, $|S| = k$. Then $rank_+(A) \geq k$. Note that there is a straightforward *upper* bound, for $A \in \mathbb{R}_+^{n\times m}$ $rank_+(A) \leq min(n,m)$. Independent elements allow us to get a *lower* bound.

Proof:

One can note that several straightforward points:

1. as $a_{i_{k_1}j_{k_1}} a_{i_{k_2}j_{k_2}} > 0$, for $k_1 \neq k_2$, and as both entries are nonnegative, it follows that $a_{i_{k_1}j_{k_1}} > 0$ and $a_{i_{k_2}j_{k_2}} > 0$

2. as $a_{i_{k_1}j_{k_2}} a_{i_{k_2}j_{k_1}} = 0$, for $k_1 \neq k_2$, it follows that $a_{i_{k_1}j_{k_2}} = 0$ or $a_{i_{k_2}j_{k_1}} = 0$

3. All independent entries occupy distinct rows and columns. Assume that independent entries share the same row. If $i_{k_1} = i_{k_2}$, then $0 = a_{i_{k_1}j_{k_2}}a_{i_{k_2}j_{k_1}} = a_{i_{k_2}j_{k_2}}a_{i_{k_1}j_{k_1}} > 0$, which is a contradiction. The same argument works for columns as well

We first need to prove that if a term(which is a rank 1 nonnegative matrix) of the factorization has a positive value in the position of one independent entry, then this term has zeros in the positions of all other independent entries.

Assume that for particular factorization $A_1 \dots A_p$, where A_j is a nonnegative rank 1 matrix and $\sum_i A_i = A$, for a particular terms A' there are 2 values in the positions of independent entries which are positive.

WLOG let these values be $A'_{i_1j_1}, A'_{i_2j_2}$ (values of A' in the positions of the first 2 independent entries)

WLOG let $a_{i_1j_2} = 0$ (point 2)

$0 = a_{i_1j_2} = \sum_j((A_j)_{i_1j_2}) \geq A'_{i_1j_2} \geq 0$ where the inequality follows from the fact that all the terms are non-negative.

It follows that $A'_{i_1j_2} = 0$

As $rank_+(A') = 1$, it follows that all the distinct non-zero rows are just the rescaling of each other, and we have assumed that rows i_1, i_2 are distinct and non-zero. Then for some α:

$A'_{i_1:} = \alpha \cdot A'_{i_2:} \implies A'_{i_1j_2} = \alpha \cdot A'_{i_2j_2} \implies \alpha = \frac{A'_{i_1j_2}}{A'_{i_2j_2}} = \frac{0}{A'_{i_2j_2}} = 0$

$A'_{i_1:} = \alpha \cdot A'_{i_2:} \implies A'_{i_1j_1} = \alpha \cdot A'_{i_2j_1} \geq A'_{i_1j_1} = 0 \cdot A'_{i_2j_1} = 0$, contradiction

Therefore in each term of the factorization, there is at most one element in positions of independent entries which is positive.

Assume that $rank_+(A) < k$, i.e. there exists a set of rank 1 nonnegative matrices $A_1 \dots A_{k-1}$ such that $\sum_i(A_i) = A$

As there are only $k-1$ rank 1 nonnegative matrices and k distinct positive independent elements in their sum (note 1, A has positive values on the positions of independent entries), for some term in a sum, call it $A' = A_i$, there are at least 2 positive values in the positions of independent entries(pigeonhole principle). We have proven that it is not possible, therefore $rank_+(A) \geq k$.

□

It is possible to extend the concept of independent entries to *independent blocks*. A block is a set of entries in a matrix, and 2 blocks are independent if for each pair of entries, where the first is from the first block and the second if from the second, either the entries are independent or their product is zero.

We can make a similar observation. If a factorization term has a positive value in a position of entry inside one of the independent blocks, it is zero in the positions of entries from all the other independent blocks.

Proof:

Let A be a nonnegative matrix, such that $rank_+(A) = k$ Let $A_1 \dots A_k$ be rank 1 nonnegative matrices such that $\sum_i (A_i) = A$.

Assume a factorization term $A' = A_i$ for some i is positive in a position of entry inside one of the independent blocks. Let this position be (i_1, j_1) and so $A'_{i_1 j_1} > 0$.

Then for any other position (i_2, j_2) of entry from any other block, either these entries are independent or $A_{i_1 j_1} \cdot A_{i_2 j_2} = 0$.

In the first case, by the previous proof, $A'_{i_1 j_1} > 0$ implies that $A'_{i_2 j_2} = 0$

In the second case, as $A_{i_1 j_1} = \sum_i (A_{i_1 j_1}) \geq A'_{i_1 j_1} > 0$, it is must be the case that $A_{i_2 j_2} = 0$, as $A_{i_1 j_1} \cdot A_{i_2 j_2} = 0$

$0 = A_{i_2 j_2} = \sum_p ((A_p)_{i_2 j_2}) \geq A'_{i_2 j_2} \geq 0$ as A' is nonnegative, therefore $A'_{i_2 j_2} = 0$.

□

Corollary

Consider nonnegative matrix A. It follows that if it is known that for independent blocks $B = \{S_1 \dots S_k\}$ of entries $S_p = \{a_{p_{i_1} p_{j_1}} \dots a_{p_{i_k} p_{j_k}}\}$, for blocks $S_i, S_j, i \neq j$ there are at least r_i terms in a factorization which are positive in positions of some entries in S_i and at least r_j terms in a factorization which are positive in positions of some entries in S_j, then these terms are different for different blocks (in other words, one particular term can correspond to only one of the blocks). This implies that there are at least $\sum_j(r_j)$ such terms which correspond to different independent blocks, and so $rank_+(A) \geq \sum_j(r_j)$

We can say that the block has a (nonnegative) rank r if there must be r matrices in the whole factorization which are positive in some of the positions of this block.

3.3.3 $\mathcal{B}_0$ Matrix

It is important to consider the following nonnegative matrix $\mathcal{B}_0$

$$\mathcal{B}_0 = \begin{pmatrix} a_1 & \cdots & a_n & 1 & 1 & 1 & 1 \\ 1 & \cdots & 1 & 1 & 1 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 1 & 1 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 1 & 1 \\ 0 & \cdots & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

It is possible to show that

$rank_+(\mathcal{B}_0) = 4$ if and only if $a_1 = \ldots = a_n = a \in [0,1]$

In this case, the factorization of size 4 has the form

$$\begin{pmatrix} a\cdots a & a & a & 0 & 0 \\ 1\cdots 1 & 1 & 1 & 0 & 0 \\ 0\cdots 0 & 0 & 0 & 0 & 0 \\ 0\cdots 0 & 0 & 0 & 0 & 0 \\ 0\cdots 0 & 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0\cdots 0 & 0 & \beta & \beta & 0 \\ 0\cdots 0 & 0 & 0 & 0 & 0 \\ 0\cdots 0 & 0 & 1 & 1 & 0 \\ 0\cdots 0 & 0 & 0 & 0 & 0 \\ 0\cdots 0 & 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0\cdots 0 & 0 & 0 & a & a \\ 0\cdots 0 & 0 & 0 & 0 & 0 \\ 0\cdots 0 & 0 & 0 & 0 & 0 \\ 0\cdots 0 & 0 & 0 & 1 & 1 \\ 0\cdots 0 & 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0\cdots 0 & \beta & 0 & 0 & \beta \\ 0\cdots 0 & 0 & 0 & 0 & 0 \\ 0\cdots 0 & 0 & 0 & 0 & 0 \\ 0\cdots 0 & 0 & 0 & 0 & 0 \\ 0\cdots 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Where $\beta = 1 - a$

Proof:

From the fact that we have a particular factorization of $\mathcal{B}_0$ into 4 terms, it is straightforward that we get

$a_i = a \in [0,1]\ \forall i \in \{1 \dots n\} \implies rank_+(\mathcal{B}_0) \le 4.$

Because the following 4 elements, highlighted by the blue circles, are independent,

$$\begin{pmatrix} a_1 & \cdots & a_n & 1 & 1 & 1 & 1 \\ 1 & \cdots & 1 & \textcircled{1} & 1 & 0 & 0 \\ 0 & \cdots & 0 & 0 & \textcircled{1} & 1 & 0 \\ 0 & \cdots & 0 & 0 & 0 & \textcircled{1} & 1 \\ 0 & \cdots & 0 & 1 & 0 & 0 & \textcircled{1} \end{pmatrix}$$

we have $rank_+(\mathcal{B}_0) \ge 4$. So $rank_+(\mathcal{B}_0) \le 4 \implies rank_+(\mathcal{B}_0) = 4$, and we complete the "if" direction.

The proof for the other direction is more complex. Assume that either for some i, j such that $j \neq i$ we have $a_i \neq a_j$ or all a_i are equal to some $a > 1$

Consider the brown and orange blocks

$$\left(\begin{array}{ccc|cccc} a_1 & \cdots & a_n & 1 & 1 & 1 & 1 \\ 1 & \cdots & 1 & 1 & 1 & 0 & 0 \\ \hline 0 & \cdots & 0 & 0 & \textcircled{1} & 1 & 0 \\ 0 & \cdots & 0 & 0 & 0 & \textcircled{1} & 1 \\ 0 & \cdots & 0 & \textcircled{1} & 0 & 0 & 1 \end{array}\right)$$

They are independent, and the brown block has rank 3, because of 3 rows and 3 independent elements (circled)

Consider a particular factorization of size 4. Because there are 3 terms, which are positive in the brown block, these 3 terms must have value zero in all the positions of the orange block entries. Only one remaining term can be positive in the positions of the orange block. Call this term B_0. This leads to the fact that the values in positions of the brown block in B_0 are equal to the values of $\mathcal{B}_0$ in the same positions.

B_0 has rank 1 and its second row is non-zero, so the first row is the rescaling of the second, (potentially with a coefficient 0). This implies that for some a

$(B_0)_{1:} = a \cdot (B_0)_{2:}$

It follows that $a_i = a \cdot 1$ for all i, i.e $a_1 = \ldots = a_n = a$ for some $a \geq 0$.

From this point, WLOG we can view $\mathcal{B}_0$ simply as

$$\begin{pmatrix} \boxed{a} & 1 & 1 & 1 & 1 \\ \boxed{1} & 1 & 1 & 0 & 0 \\ 0 & 0 & \textcircled{1} & 1 & 0 \\ 0 & 0 & 0 & \textcircled{1} & 1 \\ 0 & \textcircled{1} & 0 & 0 & 1 \end{pmatrix}$$

Consider the brown block

$$\begin{matrix} 0 & \textcircled{1} & 1 & 0 \\ 0 & 0 & \textcircled{1} & 1 \\ \textcircled{1} & 0 & 0 & 1 \end{matrix}$$

By the independence of entries, highlighted by the circles, each of the 3 terms corresponding to the brown block is positive in the position of only one of the red, green or blue entries. Let these terms be called B_r, B_g, B_b respectively. The values of these terms in the red/green/blue positions must be 1, as these are the only terms which can be positive in these positions.

Red and blue entries are independent with entry 1 in the position $(2,4)$ of the brown block, so only term B_g can have a positive value in this position, which must be 1 because this is the only factorization term which is positive in this position. The same argument, based on the independence of the green and blue entries with entry 1 in positions $(3,4)$ implies that B_r has 1 in the position $(3,4)$.

Red entry and 1 in the position $(1,3)$ are independent, so the 1 in the position $(1,3)$ is the result of the sum of B_g and B_b. However, B_g has is 1 in the position $(2,4)$, which is independent with 1 in $(1,3)$. This implies that B_g has value zero in the position $(1,3)$, leading to the conclusion that B_b has value is 1 in the position $(1,3)$.

At this step, we know that the factorization must be of the form $B_0+B_r+B_g+B_b =$

$$= \left(\begin{array}{ccccc} \boxed{a} & ? & ? & ? & ? \\ \boxed{1} & ? & ? & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right) + \left(\begin{array}{c|cccc|} 0 & ? & ? & ? & ? \\ 0 & ? & ? & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \enclose{circle}{1} & 0 & 0 & \enclose{circle}{1} \\ \hline \end{array}\right) + \left(\begin{array}{c|cccc|} 0 & ? & ? & ? & ? \\ 0 & ? & ? & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \enclose{circle}{1} & \enclose{circle}{1} \\ 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}\right) + \left(\begin{array}{c|cccc|} 0 & ? & ? & ? & ? \\ 0 & ? & ? & 0 & 0 \\ \hline 0 & 0 & \enclose{circle}{1} & \enclose{circle}{1} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}\right)$$

Each of the terms B_r, B_g, B_b has positive values in the positions of some elements which are independent with values 1 in the position $(2,2)$ and $(2,3)$ of $\mathcal{B}_0$. This implies that B_r, B_g, B_b are zero in the $(2,2)$, $(2,3)$ positions. Then B_0 has values 1 in the said positions. From the fact that $(B_0)_{1:} = a \cdot (B_0)_{2:}$, we have that $(B_0)_{12} = (B_0)_{13} = a$. The fact that $(B_0)_{1:} = a \cdot (B_0)_{2:}$ also implies that $(B_0)_{14} = (B_0)_{15} = 0$

We conclude that

$$B_0 = \begin{pmatrix} a & a & a & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Out of the matrices B_0, B_r, B_g and B_b, only B_0 and B_r have no positive values in the positions of entries which are independent with entries of value 1 in the position $(1,2)$, and only B_0 and B_b have no positive values in the positions of any entries independent with the entry of value 1 in the position $(1,3)$. The following matrices show which elements, corresponding to B_g, B_b, are independent with 1 in $(1,2)$, and which elements, corresponding to B_r, B_g, are independent with 1 in $(1,3)$.

$$\begin{pmatrix} a & \enclose{circle}{1} & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & \enclose{circle}{1} & 0 \\ 0 & 0 & 0 & 1 & \enclose{circle}{1} \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a & 1 & \enclose{circle}{1} & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & \enclose{circle}{1} \\ 0 & \enclose{circle}{1} & 0 & 0 & 1 \end{pmatrix}$$

This leads to the fact that $(B_0)_{12} + (B_r)_{12} = a + (B_r)_{12} = (\mathcal{B}_0)_{12} = 1$, $(B_0)_{13} + (B_b)_{13} = a + (B_b)_{13} = (\mathcal{B}_0)_{13} = 1$

$\implies (B_b)_{12} = (B_r)_{12} = 1 - a$, which can be the case if and only if

$1 - a \geq 0 \implies a \leq 1$, which together with the fact that $a \geq 0$ implies $a \in [0, 1]$

From the fact that B_r is rank 1, and rows 5 and 1 are non-zero, it follows that for some β

$\beta \cdot (B_r)_{5:} = (B_r)_{1:} \implies \beta = \frac{(B_r)_{12}}{(B_r)_{52}} = 1 - a \implies (B_r)_{15} = (B_r)_{55} \cdot \beta = 1 \cdot (1 - a)$

Similar logic can be applied to conclude that $(B_b)_{14} = (1 - a)$.

By the same argument, it is also straightforward that $(B_r)_{13} = (B_r)_{14} = (B_b)_{12} = (B_b)_{15} = 0$. We can conclude that B_r, B_b have the following form

$$\begin{pmatrix} 0 & \enclose{circle}{\beta} & 0 & 0 & \enclose{circle}{\beta} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \enclose{circle}{1} & 0 & 0 & \enclose{circle}{1} \end{pmatrix} \begin{pmatrix} 0 & 0 & \enclose{circle}{\beta} & \enclose{circle}{\beta} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \enclose{circle}{1} & \enclose{circle}{1} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Where $\beta = 1 - a$

It remains to find the form of B_g

$\mathcal{B}_0 = B_0 + B_r + B_g + B_b \implies B_g = \mathcal{B}_0 - (B_0 + B_r + B_b)$

$$B_g = \begin{pmatrix} a & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} a & a & a & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} - \begin{pmatrix} 0 & b & 0 & 0 & b \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 0 & b & b & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$B_g = \begin{pmatrix} 0 & 0 & 0 & \textcircled{a} & \textcircled{a} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \textcircled{1} & \textcircled{1} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

We have proven that if $rank_+(\mathcal{B}_0) = 4$, $a_1 = \ldots = a_n = ain[0,1]$ and the factorization is of the form $B_0 + B_r + B_g + B_b$ which is described above.

This finishes the proof.

□

$rank_+(\mathcal{B}_0)$ is bounded below by the number of independent elements, which is 4, and above by the number of rows, which is 5. This leads to the conclusion that if the conditions of the above proof are not satisfied, $rank_+(\mathcal{B}_0) = 5$. This can be the case if either it is not the case that $a_1 = \ldots = a_n = a \in [0,1]$ or if the factorization of $\mathcal{B}_0$ does not have the particular form stated in the proof.

3.3.4 Variable Gadget

In this part, I present my work where I expand the proof presented in proposition 5 of [Shitov, 2016], with the intention to clarify it.

The proof in [Shitov, 2016] presents a "variable gadget", a tool which allows removing variables from an incomplete matrix. It considers an incomplete matrix B which contains t variables x, where $x \in [a,b]$, where no 2 instances of x share the same row or column.

Given B, consider matrix B' which is obtained by the removal of rows and columns which contain x from B. If $rank_+(B') \geq q$, it is possible to construct new(potentially still incomplete) matrix G such that $fact_+(B,q) \sim fact + (G, q + 5t + 4)$, and G does not contain the variable x. The time it takes to construct G and the final size of G is polynomial in the size of B.

The proof presented in proposition 5 of [Shitov, 2016] introduces a number of new constants which are connected with the range $[a, b]$ of x, but no definition is made. This makes the original proof more general than the one I will present, however it still proves the point which we are interested in.

It can be noted that the constants M, N, P, Q, presented in the original proof, can have the following values: $M = b + 1, N = 1, P = 1/((b - a) + 1), Q = 1$

WLOG we can state that $b > a$, otherwise, if $a = b$ x can be considered just as a constant.

WLOG we can assume that all variables x occupy the first t diagonal entries. If it is not the case, the rows and columns can be relabelled/rearranged. li Given such B, consider the partition

$$B = \left(\begin{array}{c|c} B_1 & B_2 \\ \hline B_3 & B_4 \end{array} \right)$$

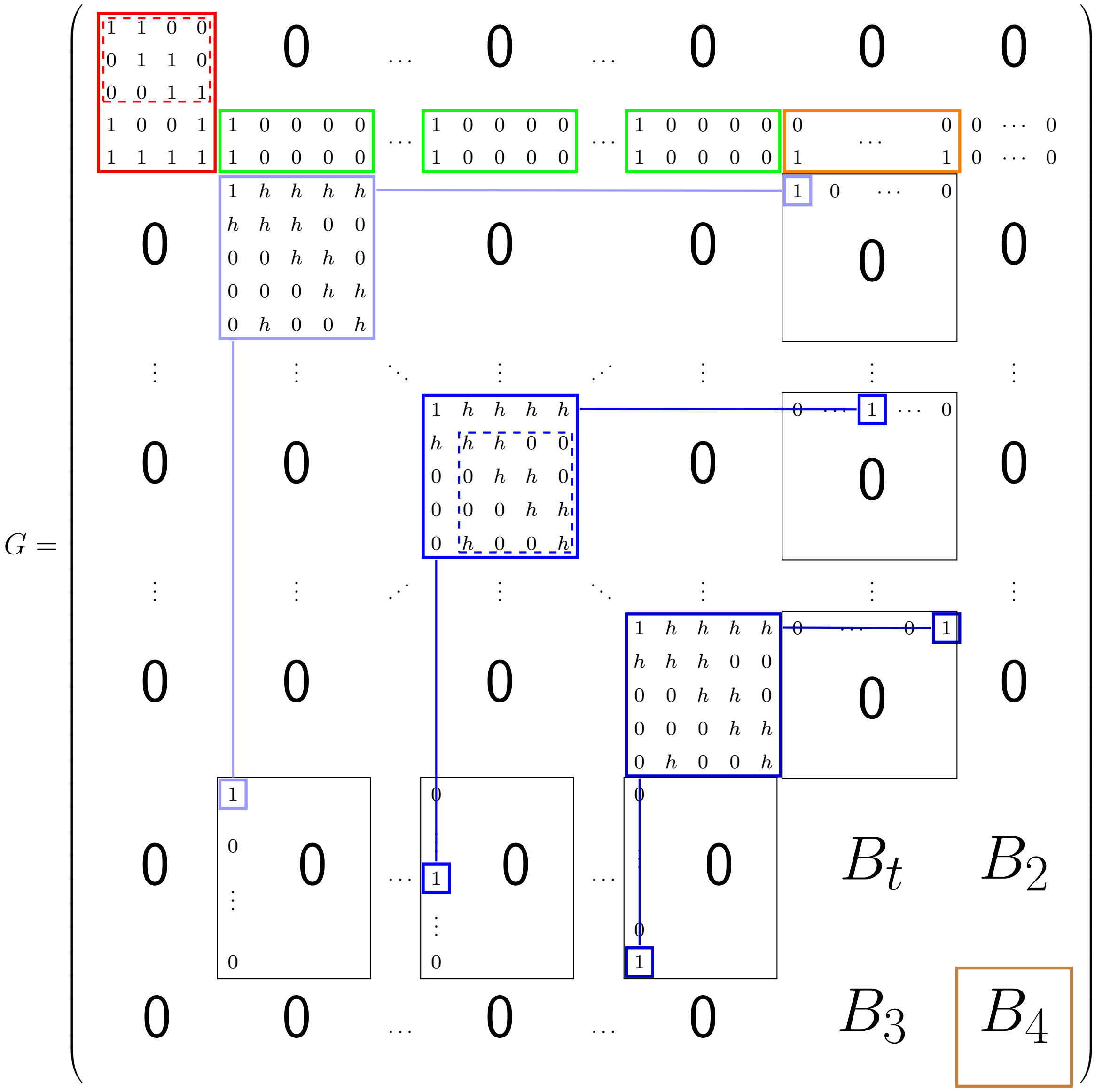
such that x instances occupy positions $(1,1) \dots (t,t)$ and B_1 is $t \times t$ matrix. B_4 is obtained by the removal of all rows and columns containing x from B.

Let q be such that $rank_+(B_4) \geq q$. It must follow that $rank_+(B) \geq rank_+(B_4) \geq q$ as B_4 is a submatrix of B.

Let $h = \frac{b-a}{b-a+1}$, note that $h \in (0,1)$.

Let B_t be B_1 where all instances of variable x(which occupy the diagonal) are replaced with a constant $b + 1$.

Then the $fact_+(B,q) \sim fact_+(G, q' = q+5t+4)$, where G is



Note that if $rank_+(B) > g$, $fact_+(B,q) = \emptyset$ and $fact_+(B,q) \sim fact_+(G,q')$ must be a trivial strong equivalence of the empty sets.

Proof:

I will show that for any factorization in $fact_+(B,q)$, there is a unique corresponding factorization in $fact(G,q')$, and for any factorization in $fact_+(G,q')$, there is a unique factorization in $fact_+(B,q)$ and there is no irrationality in the mapping from one or the other direction. (as it was mentioned before, uniqueness is up to the order of terms). The proof constructs this bijection.

Let $F=(G_1 \dots G_{q'}) \in fact_+(G,q')$ be a particular factorization.

Note that G can be partitioned into $t+2$ independent blocks, highlighted by solid red, brown and blue lines. We can note that the 4 diagonal elements of the red block are independent, and so the first block has nonnegative rank 4. Each of the t blue blocks is of the nonnegative rank 5, as each block is $h \cdot \mathcal{B}_0$, where $\mathcal{B}_0$ is from the previous section, where the value in the position $(1,1)$ is $a=\frac{1}{h}>1$.

It is important to highlight a 4×4 blue "subblock". This subblock is independent with every other block outside the blue block, so if one particular term is positive inside it, it is zero everywhere outside the blue block. Same is true for the 3×4 red subblock, which is independent with every other block outside the red block.

By the definition of q, the brown B_4 block is of the rank at least q.

We can label the terms of the factorization such that the first 4 terms in the factorization are positive in the positions of the red block, $5t$ following terms are positive in the positions of the blue blocks, and the remaining q terms are positive in the positions of the brown block. Note that this already implies that in order for F to exist, it must be the case that $rank_+(B_4) \leq q$, which leads to $rank_+(B_4)=q$ if $fact_+(G,q')$ is non-empty.

Consider the matrices $(G'_1 \dots G'_5)=(G_{4+5\tau} \dots G_{4+5\tau+5})$ for some τ which are positive in the positions of the τ^{th} blue block.

$$\begin{array}{ccccccc}
1 & h & h & h & h & \cdots & \boxed{1} \\
h & h & h & 0 & 0 & & \\
0 & 0 & h & h & 0 & & \\
0 & 0 & 0 & h & h & & \\
0 & h & 0 & 0 & h & & \\
\vdots & & & & & & \\
\boxed{1} & & & & & &
\end{array}$$

Note that it must be the case that only one matrix is positive in the positions of the two entries of value 1 which are separated from the larger block. This is because these entries of value 1 are independent with the block highlighted by the dashed line, and this block has at least 4 corresponding terms which are positive in some of its positions(as its 4 diagonal entries are mutually independent). This leaves only 1 term to be positive in the positions of the two entries of value 1 which are separated from the larger block. Let this term be G_1'. It follows that the blue block of the matrix G_1' has the form

$$G_1' = \begin{array}{ccccccc}
u_\tau & 0 & 0 & 0 & 0 & \cdots & \boxed{1} \\
0 & 0 & 0 & 0 & 0 & & \\
0 & 0 & 0 & 0 & 0 & & \\
0 & 0 & 0 & 0 & 0 & & \\
0 & 0 & 0 & 0 & 0 & & \\
\vdots & & & & & & \\
\boxed{1} & & & & & & \boxed{1/u_\tau}
\end{array}$$

for some u_τ.

This implies that the sum of the remaining terms is

$$G_2' + G_3' + G_4' + G_5' = h\cdot \begin{pmatrix} \boxed{\begin{matrix} \frac{1-u_\tau}{h} & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{matrix}} & \cdots & \boxed{0} \\ \vdots & & \\ \boxed{0} & & 0 \end{pmatrix}$$

Which, by the theorem about the matrix $\mathcal{B}_0$, implies 2 facts:

1. Let $v_\tau = 1 - u_\tau$. The 2-5 terms are of the form

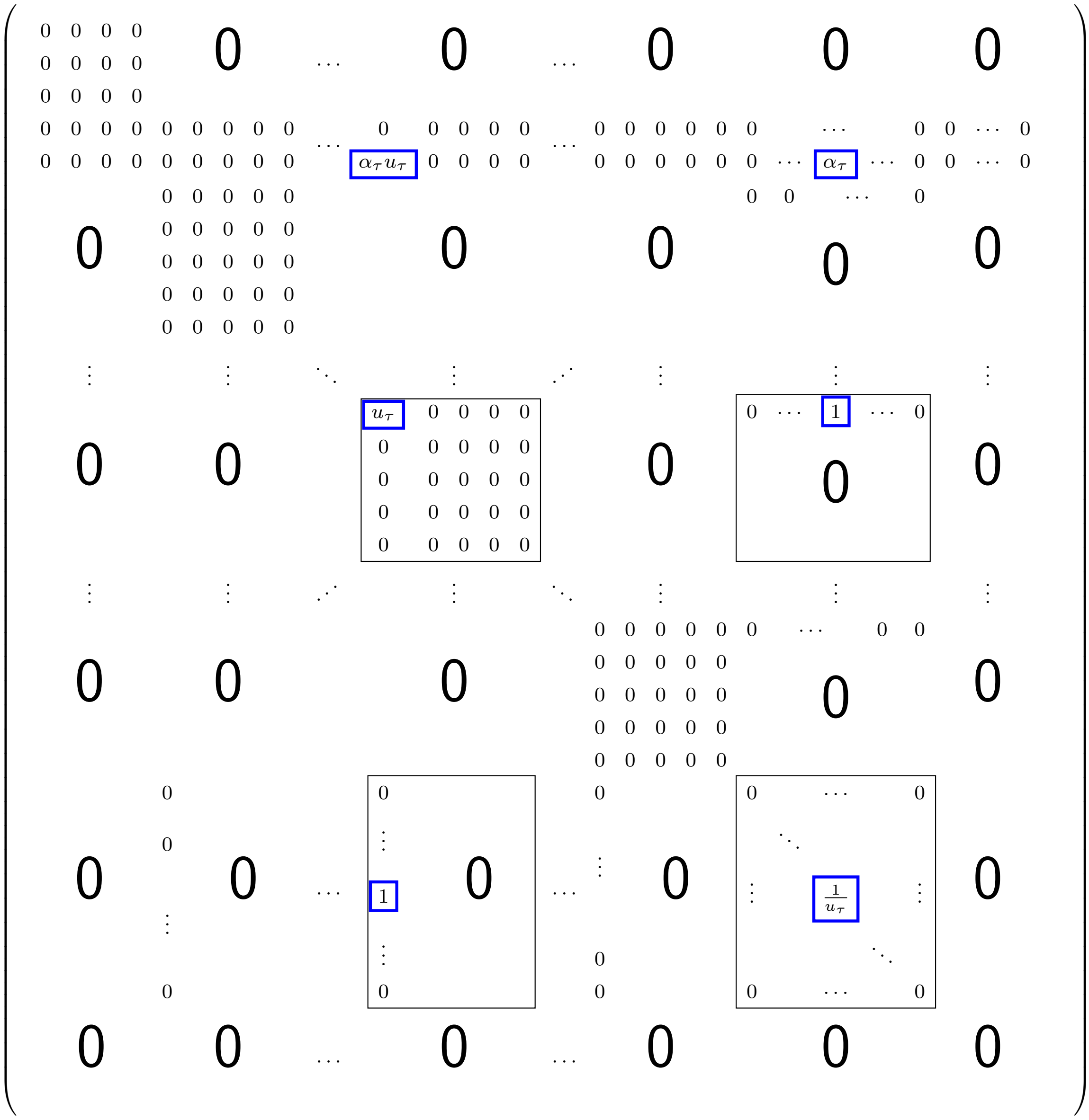
$$G_2' = \begin{pmatrix} v_\tau & v_\tau & v_\tau & 0 & 0 & \cdots & 0 \\ h & h & h & 0 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & & \\ \vdots & & & & & & \\ 0 & & & & & & 0 \end{pmatrix} \qquad G_3' = \begin{pmatrix} 0 & h - v_\tau & 0 & 0 & h - v_\tau & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & & \\ 0 & h & 0 & 0 & h & & \\ \vdots & & & & & & \\ 0 & & & & & & 0 \end{pmatrix}$$

$$G_4' = \begin{pmatrix} 0 & 0 & h - v_\tau & h - v_\tau & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & & \\ 0 & 0 & h & h & 0 & & \\ 0 & 0 & 0 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & & \\ \vdots & & & & & & \\ 0 & & & & & & 0 \end{pmatrix} \qquad G_5' = \begin{pmatrix} 0 & 0 & 0 & v_\tau & v_\tau & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & & \\ 0 & 0 & 0 & h & h & & \\ 0 & 0 & 0 & 0 & 0 & & \\ \vdots & & & & & & \\ 0 & & & & & & 0 \end{pmatrix}$$

2. $1 - u_\tau \in [0, h] \implies u_\tau \in [1 - h, 1] = [\frac{1}{b-a+1}, 1]$, and $\frac{1}{u_\tau} \in [1, b - a + 1]$

The subblock highlighted by the dashed blue line is independent with every other block of the matrix. However, it is not the case for the remaining elements of the blue block. We can conclude that the terms $G'_2 \dots G'_5$ are zero everywhere outside the blue block. But the term G'_1 can have non-zero elements in the green and orange blocks. G'_1 has the following form for some α_τ

$G'_1 =$



The B_4 brown block is independent with all the other possible blocks of G, in particular, it is independent with the ones on the 4^{th} row. This leads to the fact that the only terms which can be positive in the positions of these entries of the value one are the first 4 terms which correspond to the red block. Given the structure of the terms corresponding to the blue blocks and the independence of the brown block with other blocks, the sum of the terms corresponding to the red block is

$$\left(\begin{array}{ccccccccccccccccccccccccccc}
1 & 1 & 0 & 0 & \multicolumn{5}{c}{\mathbf{0}} & \cdots & \multicolumn{5}{c}{\mathbf{0}} & \cdots & \multicolumn{5}{c}{\mathbf{0}} & \multicolumn{3}{c}{\mathbf{0}} & \multicolumn{3}{c}{\mathbf{0}} \\
0 & 1 & 1 & 0 & & & & & & & & & & & & & & & & & & & & & & & \\
0 & 0 & 1 & 1 & & & & & & & & & & & & & & & & & & & & & & & \\
1 & 0 & 0 & 1 & \textcircled{1} & 0 & 0 & 0 & 0 & \cdots & \textcircled{1} & 0 & 0 & 0 & 0 & \cdots & \textcircled{1} & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
1 & 1 & 1 & 1 & \boxed{v_1} & 0 & 0 & 0 & 0 & \cdots & \boxed{v_\tau} & 0 & 0 & 0 & 0 & \cdots & \boxed{v_t} & 0 & 0 & 0 & 0 & \boxed{1-\alpha_1} & & \boxed{1-\alpha_t} & 0 & \cdots & 0 \\
\multicolumn{27}{c}{\boxed{\qquad\qquad\mathbf{0}\qquad\qquad}}
\end{array}\right)$$

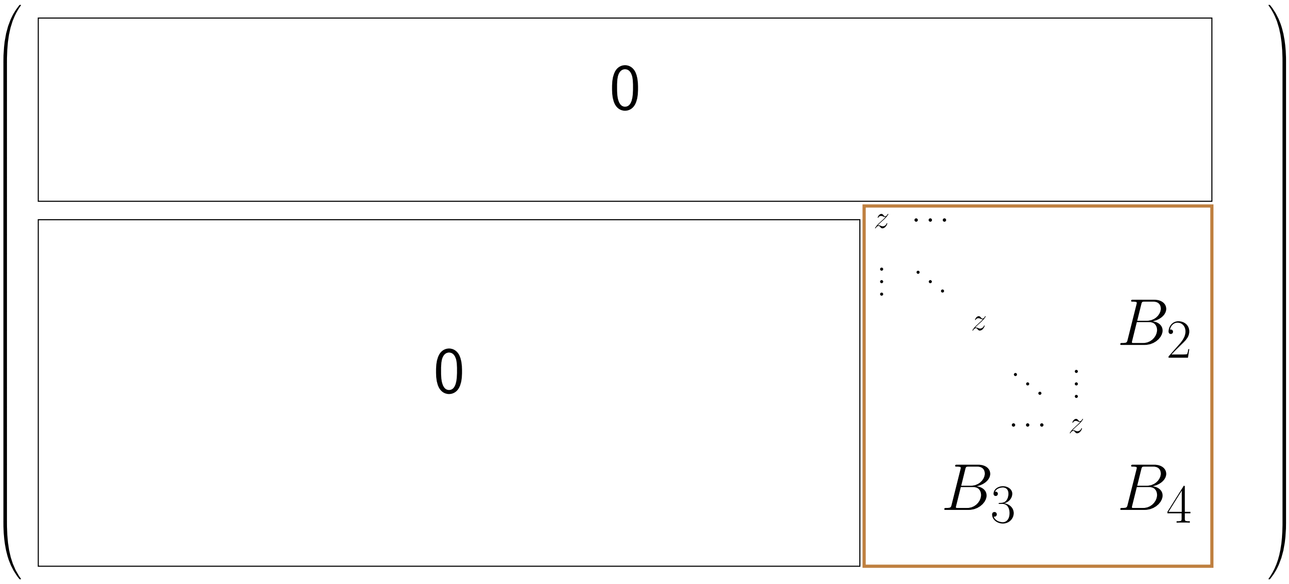
Note that the first 3 diagonal entries are independent, and there are only 4 terms corresponding to the red block. This implies that 3 of the terms are nonnegative in the positions of the first 3 rows. The entries of value 1 in the first 3 rows are independent with the entries of value 1 from the 4^{th} row which are highlighted by the red circles above, as well as with the entries in the positions of $1-\alpha_\tau$. The remaining 4^{th} term must contain both the values 1 in the positions of the red circles, as well as the entries $1-\alpha_\tau$. This is possible if and only if $1-\alpha_\tau = 0 \implies \alpha_\tau = 1\ \forall\tau$. It is easy to see that the entries of value 1, highlighted by the red circles, and the entries in the positions of $1-\alpha_\tau$ form 2 independent "subblocks".

Then the only positive elements of the red block are the ones which are insider the dashed red line in the illustration above. This block resembles the $\mathcal{B}_0$ matrix, with certain rows and columns swapped. It has the same property, that it has nonnegative rank 4 if and only if $v_1 = \ldots = v_t = v \in [0,1]$ (and so $u_1 = \ldots = u_t = 1 - v$), and the terms corresponding to the red block are uniquely determined by the value of v. (Given the previous information, we know that $v \in [0,h] \subset [0,1]$)

All the $4 + 5t$ terms sum to.

$$\left(\begin{array}{cccccccc}
\boxed{\begin{matrix}1&1&0&0\\0&1&1&0\\0&0&1&1\\1&0&0&1\\1&1&1&1\end{matrix}} &
\begin{matrix}\mathbf{0}\\ \boxed{\begin{matrix}1&0&0&0&0\\1&0&0&0&0\end{matrix}}\end{matrix} &
\begin{matrix}\cdots\\ \cdots\end{matrix} &
\begin{matrix}\mathbf{0}\\ \boxed{\begin{matrix}1&0&0&0&0\\1&0&0&0&0\end{matrix}}\end{matrix} &
\begin{matrix}\cdots\\ \cdots\end{matrix} &
\begin{matrix}\mathbf{0}\\ \boxed{\begin{matrix}1&0&0&0&0\\1&0&0&0&0\end{matrix}}\end{matrix} &
\begin{matrix}\mathbf{0}\\ \boxed{\begin{matrix}0&\cdots&0\\1&&1\end{matrix}}\end{matrix} &
\begin{matrix}\mathbf{0}\\ \begin{matrix}0&\cdots&0\\0&\cdots&0\end{matrix}\end{matrix} \\
\mathbf{0} &
\boxed{\begin{matrix}1&h&h&h&h\\h&h&h&0&0\\0&0&h&h&0\\0&0&0&h&h\\0&h&0&0&h\end{matrix}} &
& \mathbf{0} & & \mathbf{0} &
\boxed{\begin{matrix}\boxed{1}&0&\cdots&0\\ &&\mathbf{0}&\end{matrix}} &
\mathbf{0} \\
\vdots & \vdots & \ddots & \vdots & \iddots & \vdots & \vdots & \vdots \\
\mathbf{0} & \mathbf{0} & &
\boxed{\begin{matrix}1&h&h&h&h\\h&h&h&0&0\\0&0&h&h&0\\0&0&0&h&h\\0&h&0&0&h\end{matrix}} &
& \mathbf{0} &
\boxed{\begin{matrix}0&\cdots&\boxed{1}&\cdots&0\\ &&\mathbf{0}&&\end{matrix}} &
\mathbf{0} \\
\vdots & \vdots & \iddots & \vdots & \ddots & \vdots & \vdots & \vdots \\
\mathbf{0} & \mathbf{0} & & \mathbf{0} & &
\boxed{\begin{matrix}1&h&h&h&h\\h&h&h&0&0\\0&0&h&h&0\\0&0&0&h&h\\0&h&0&0&h\end{matrix}} &
\boxed{\begin{matrix}0&\cdots&0&\boxed{1}\\ &\mathbf{0}&&\end{matrix}} &
\mathbf{0} \\
\mathbf{0} &
\boxed{\begin{matrix}\boxed{0}&\\0&\mathbf{0}\\ \vdots&\\0&\end{matrix}} &
\cdots &
\boxed{\begin{matrix}0&\\ \vdots&\\ \boxed{1}&\mathbf{0}\\ \vdots&\\0&\end{matrix}} &
\cdots &
\boxed{\begin{matrix}0&\\ \vdots&\mathbf{0}\\0&\\ \boxed{0}&\end{matrix}} &
\begin{matrix}\boxed{\frac{1}{u}}&0&\cdots&0\\0&\ddots&&\vdots\\ \vdots&&\boxed{\frac{1}{u}}&\\ &&\ddots&0\\0&\cdots&0&\boxed{\frac{1}{u}}\end{matrix} &
\mathbf{0} \\
\mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0}
\end{array}\right)$$

This means that the remaining q terms corresponding to the brown block must sum to



Where $z = b + 1 - \frac{1}{v}$, and, $\frac{1}{v} \in [1, b - a + 1]$, we get that $z \in [a, b]$

Consider the factorization of G of size $4 + 5t + q$. This factorization is uniquely determined by the q terms corresponding to the brown block. They uniquely define $z = b + 1 - \frac{1}{v} \in [a, b]$, which uniquely defines $v = \frac{1}{b-z+1}$. The value of v uniquely defines the remaining $4+5t$ terms corresponding to the blue and red blocks, as described above.

For a particular factorization of G, the corresponding factorization of B can be obtained by considering the q terms corresponding to the brown block. These terms must sum to a particular completion of B, where all the t instances of x are equal to $z \in [a, b]$. The q terms, corresponding to the brown block, can be viewed as a result of permutation p, followed by a projection g applied to the particular factorization of G. (Note that any permutation is bijective). Because the q terms which correspond to the brown block uniquely define the factorization of G, g is injective. The mapping from $fact_+(G, q')$ to $fact_+(B, q)$ can be inverted by rational functions, as the (re)construction of the $4 + 5t$ terms corresponding to the red and blue blocks, based on the particular value $z \in [a, b]$, involves no irrationality, so g is a rational projection.

Consider a particular factorization of B of size q. It corresponds to a particular completion of B, where $x = x' \in [a, b]$. This particular factorization of B can be considered as the q terms which correspond to the brown block of some factorization of G, where $z = x' = b + 1 - \frac{1}{v}$. It uniquely defines the corresponding factorization of G of size q', such that this particular factorization of B can be obtained by applying permutations p and then projection g to the uniquely defined factorisation of G. We can see that any valid factorisation of B of size q can be obtained from some factorisation of G of size q'. It follows that g is surjective, and so it is bijective.

$p \circ g : fact_+(G, q') \mapsto fact_+(B, q)$ is bijective. Therefore $fact_+(G, q') \sim fact_+(B, q)$.

□

Note, that if there are several variables which occupy the region B_1, they can be removed iteratively. It is easy to see that if all the rows and columns containing block B_t are removed from G, the rank of the remaining matrix will be bounded below by $q' := 4 + 5t + q$, as the red and brown blocks are still in place, and each of the t blue blocks contains $h \cdot \mathcal{B}_0$, where the value in $(1, 1)$ is such that the block has nonnegative rank 5.

Assume that, in addition to t instances of the variable x, there are t' instances of the variable y inside the B_1 block of B. After the application of the variable gadget with respect to x, all t' instances of y occupy the block B_t of G. If all the rows and columns which contain y are removed, the nonnegative rank of the remaining matrix is bounded below by q'.

Therefore, it is possible to apply the variable gadget described above to G with respect to y and construct G' such that $fact_+(G', q'' = 4 + 5t' + q') \sim fact_+(G, q')$, and as $fact_+(B, q) \sim fact_+(G, q' = 4 + 5t + q)$, it follows that $fact_+(B, q) \sim fact_+(G, q') \sim fact_+(G', q'')$.

This way, all the variables which occupy the original B_1 block can be iteratively removed.

3.3.5 Incomplete Matrix for the Linear Combination Constraint

As it was shown before, it is possible to transform a polynomial equation into the conjunction of several constraints, which either take the form of the product or the form of the linear combination with nonnegative coefficients. This section focuses on the later.

Consider the following set of real vectors

$U := \{(L, x_2 \ldots x_n) : L = k_1 + \sum_{i=2}^{n} x_i k_i \wedge x_1 \ldots x_n \in [0,1] \wedge L \in [0, \sum k_i]\}$ where $k_1 \ldots k_n$ are nonnegative constants.

It was proven in lemma 7 of [Shitov, 2016] that the set of factorizations of the following incomplete matrix S of size $5n$ is strongly equivalent to U

$$
S=\left(\begin{array}{ccccc}
\begin{array}{c}\boxed{1}\\ \boxed{x_2}\\ \vdots\\ \boxed{x_t}\\ \vdots\\ \boxed{x_n}\\ L\end{array} & & \mathbf{0} & & \begin{array}{cccccc}\boxed{1}&0&\cdots&0&\cdots&0\\ 0&\boxed{1}&\cdots&0&\cdots&0\\ \vdots&&\ddots&&&\vdots\\ 0&&&\boxed{1}&&0\\ \vdots&&&&\ddots&\vdots\\ 0&\cdots&0&\cdots&0&\boxed{1}\\ \boxed{k_1}&\boxed{k_2}&\cdots&\boxed{k_t}&\cdots&\boxed{k_n}\end{array}\\
\begin{array}{c}1\\1\\0\\0\\0\end{array} & \begin{array}{c}\begin{matrix}1&1&1&1\end{matrix}\\ \boxed{\begin{matrix}1&1&0&0\\0&1&1&0\\0&0&1&1\\1&0&0&1\end{matrix}}\end{array} & \mathbf{0} & \mathbf{0} & \begin{array}{cccccc}\boxed{1}&0&\cdots&0&\cdots&0\\0&&&&&\\0&&&\mathbf{0}&&\\0&&&&&\\0&&&&&\end{array}\\
\vdots & \ddots & & \iddots & \\
\begin{array}{c}1\\1\\0\\0\\0\end{array} & \mathbf{0} & \begin{array}{c}\begin{matrix}1&1&1&1\end{matrix}\\ \boxed{\begin{matrix}1&1&0&0\\0&1&1&0\\0&0&1&1\\1&0&0&1\end{matrix}}\end{array} & \mathbf{0} & \begin{array}{cccccc}0&0&\cdots&\boxed{1}&\cdots&0\\0&&&&&\\0&&&\mathbf{0}&&\\0&&&&&\\0&&&&&\end{array}\\
\vdots & & \iddots & \ddots & \\
\begin{array}{c}1\\1\\0\\0\\0\end{array} & \mathbf{0} & \mathbf{0} & \begin{array}{c}\begin{matrix}1&1&1&1\end{matrix}\\ \boxed{\begin{matrix}1&1&0&0\\0&1&1&0\\0&0&1&1\\1&0&0&1\end{matrix}}\end{array} & \begin{array}{cccccc}0&0&\cdots&0&\cdots&\boxed{1}\\0&&&&&\\0&&&\mathbf{0}&&\\0&&&&&\\0&&&&&\end{array}
\end{array}\right)
$$

Where $L, x_2 \ldots x_n$ are variables, $L \in [0, \sum k_i]$, $x_1 \ldots x_n \in [0, 1]$.

Consider the set of factorizations $fact_+(S, 5n)$.

The red and blue blocks, highlighted by the solid lines, are independent. Red blocks have the nonnegative rank 4, as their diagonal entries are independent. This means that in a particular factorization, there are at least 4 terms corresponding to each red block(4 terms, which are nonnegative in the position of one particular red block). This means that there are at least $4n$ terms corresponding to the all red blocks, and at most n remaining terms correspond to the n blue blocks. At least 1 term must correspond to each blue block. This leads to the fact that at most $4n$ terms correspond to n red block, and so exactly 4 terms correspond to each red block.

In addition, dashed red lines indicate the regions, outside of which every positive element is independent with the inside blocks highlighted by the solid red line. This leads to the fact that the terms which correspond to each of the red blocks can be positive only inside the region indicated by the dashed red line around said red block.

Each green block is independent with all the red blocks and all the blue blocks except the one blue block which is in the same column with the said green block. This means that the value inside the green block is the result of the term corresponding to this one blue block only. Same is true for the brown and orange blocks, except that in the case of the orange blocks we must consider the same rows instead of the same columns.

The term corresponding to one particular blue block is of rank 1. It must have the following form, for some constant v_t (if $t = 1$, $v_t = 1$)

$$\begin{pmatrix}
0 & & & & & & & & & & & & & & & 0 & \cdots & 0 & \cdots & 0 \\
\vdots & & & & & & & & & & & & & & & \vdots & \ddots & & & \vdots \\
\boxed{v_t} & & & & & & & \mathbf{0} & & & & & & & & 0 & & \boxed{1} & & 0 \\
\vdots & & & & & & & & & & & & & & & \vdots & & & \ddots & \vdots \\
0 & & & & & & & & & & & & & & & 0 & \cdots & 0 & \cdots & 0 \\
\boxed{v_t \cdot k_t} & & & & & & & & & & & & & & & 0 & \cdots & \boxed{k_t} & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & & & & & & & & & & & 0 & \cdots & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & & & \mathbf{0} & & & & & \mathbf{0} & & & 0 & & & & \\
0 & 0 & 0 & 0 & 0 & & & & & & & & & & & 0 & & \mathbf{0} & & \\
0 & 0 & 0 & 0 & 0 & & & & & & & & & & & 0 & & & & \\
0 & 0 & 0 & 0 & 0 & & & & & & & & & & & 0 & & & & \\
\vdots & & & & & \ddots & & & & & \iddots & & & & & & & & & \\
\boxed{v_t} & & & & & & 0 & 0 & 0 & 0 & & & & & & 0 & \cdots & \boxed{1} & \cdots & 0 \\
0 & & & \mathbf{0} & & & 0 & 0 & 0 & 0 & & & \mathbf{0} & & & 0 & & \mathbf{0} & & \\
0 & & & & & & 0 & 0 & 0 & 0 & & & & & & 0 & & & & \\
0 & & & & & & 0 & 0 & 0 & 0 & & & & & & 0 & & & & \\
0 & & & & & & 0 & 0 & 0 & 0 & & & & & & 0 & & & & \\
\vdots & & & & & \iddots & & & & & \ddots & & & & & & & & & \\
0 & & & & & & & & & & & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 \\
0 & & & \mathbf{0} & & & & \mathbf{0} & & & & 0 & 0 & 0 & 0 & 0 & & & & \\
0 & & & & & & & & & & & 0 & 0 & 0 & 0 & 0 & & \mathbf{0} & & \\
0 & & & & & & & & & & & 0 & 0 & 0 & 0 & 0 & & & & \\
0 & & & & & & & & & & & 0 & 0 & 0 & 0 & 0 & & & &
\end{pmatrix}$$

The t^{th} red block then must sum to

$$\begin{pmatrix} \begin{matrix} \boxed{\begin{matrix} u_t \\ 1 \end{matrix}} \\ 0 \\ 0 \\ 0 \end{matrix} & \cdots & \boxed{\begin{matrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{matrix}} \end{pmatrix}$$

Where $u_t = 1 - v_t$

As there are only 4 terms corresponding to each red block, using the result about the $\mathcal{B}_0$ matrix, we conclude that $u_t \in [0, 1]$, and so $v_t \in [0, 1]$

The only term which can be positive in the position of x_t is the one which corresponds to one blue entry which has the value v_t in the position of x_t.

The only terms which can be positive in the position of L are the ones which correspond to the blue entries, and each of them contributes a value of $v_t \cdot k_t$.

This means that a particular factorization of S of size $5n$ is the factorization of the completion of S where $x_t = v_t \in [0, 1]$ and $L = \sum_i s_i{\cdot}v_i = \sum_i s_i{\cdot}x_i = k_1 + \sum_{i=2}^n s_i{\cdot}x_i$ (and so $L \in [0, \sum_i k_i]$). This factorization is completely determined by the values of $x_2 \dots x_n$. Obtaining the values $L, x_2 \dots x_n$ from the factorisation of S can be considered as a permutation, followed by a projection. These values form a vector which belongs to the set U which was defined at the beginning. At the same time, given the values which satisfy the condition which defines U, it is possible to construct the unique factorization of S of size $5n$ in polynomial time, and construction does not include irrationality.

We can conclude that $fact_+(S, 5n) \sim U$ □

Consider the following partition of S

$$S = \left(\begin{array}{ccccccccccccccccccc}
1 & & & & & & & & & & & & & 1 & 0 & \cdots & 0 & \cdots & 0 \\
x_2 & & & & & & & & & & & & & 0 & 1 & \cdots & 0 & \cdots & 0 \\
\vdots & & & & & & \mathbf{0} & & & & & & & \vdots & & \ddots & & & \vdots \\
x_t & & & & & & & & & & & & & 0 & & & 1 & & 0 \\
\vdots & & & & & & & & & & & & & \vdots & & & & \ddots & \vdots \\
x_n & & & & & & & & & & & & & 0 & \cdots & 0 & \cdots & 0 & 1 \\
L & & & & & & & & & & & & & k_1 & k_2 & \cdots & k_t & \cdots & k_n \\
1 & 1 & 1 & 1 & 1 & & & & & & & & & \boxed{1} & 0 & \cdots & 0 & \cdots & 0 \\
1 & 1 & 1 & 0 & 0 & & & & & & & & & 0 & & & & & \\
0 & 0 & 1 & 1 & 0 & & \mathbf{0} & & & & \mathbf{0} & & & 0 & & & \mathbf{0} & & \\
0 & 0 & 0 & 1 & 1 & & & & & & & & & 0 & & & & & \\
0 & 1 & 0 & 0 & 1 & & & & & & & & & 0 & & & & & \\
\vdots & & & & & \ddots & & & & \iddots & & & & & & & & & \\
1 & & & & & 1 & 1 & 1 & 1 & & & & & 0 & 0 & \cdots & \boxed{1} & \cdots & 0 \\
1 & & & & & 1 & 1 & 0 & 0 & & & & & 0 & & & & & \\
0 & & \mathbf{0} & & & 0 & 1 & 1 & 0 & & \mathbf{0} & & & 0 & & & \mathbf{0} & & \\
0 & & & & & 0 & 0 & 1 & 1 & & & & & 0 & & & & & \\
0 & & & & & 1 & 0 & 0 & 1 & & & & & 0 & & & & & \\
\vdots & & & & & \iddots & & & & \ddots & & & & & & & & & \\
1 & & & & & & & & & 1 & 1 & 1 & 1 & 0 & 0 & \cdots & 0 & \cdots & \boxed{1} \\
1 & & & & & & & & & 1 & 1 & 0 & 0 & 0 & & & & & \\
0 & & \mathbf{0} & & & & \mathbf{0} & & & 0 & 1 & 1 & 0 & 0 & & & \mathbf{0} & & \\
0 & & & & & & & & & 0 & 0 & 1 & 1 & 0 & & & & & \\
0 & & & & & & & & & 1 & 0 & 0 & 1 & 0 & & & & &
\end{array}\right)$$

If all rows and columns which contain variables are removed, the black block remains. The highlighted coloured blocks are independent and each of them has non-negative rank 5. It is easy to see this from the fact that following 5 elements are independent in these blocks, it follows that the rank of a single block is bounded by 5 from below and that it is bounded by 5 from above by the number of rows and columns.

$$\begin{array}{cccccc}
1 & 1 & 1 & 1 & \cdots & \textcircled{1} \\
\textcircled{1} & 1 & 0 & 0 & & 0 \\
0 & \textcircled{1} & 1 & 0 & & 0 \\
0 & 0 & \textcircled{1} & 1 & & 0 \\
1 & 0 & 0 & \textcircled{1} & & 0
\end{array}$$

The black block then has the nonnegative rank bounded below by $5n$. This implies that if for one particular variable the rows and the columns which contain the instances of this variable are removed, the nonnegative rank of the remaining matrix will be bounded below by $5n$ as well (the remaining matrix will still contain the black block). Each variable is present only once, so no 2 same variables share the same row or column. This means that they can be removed, by applying the variable gadget to S several times with respect to each variable, and obtaining the (variable-free) matrix S' such that $fact_+(S, 5n) \sim fact_+(S', q')$ for some q'.

3.3.6 Incomplete Matrix for the Product Constraint

As it was said before, it is possible to transform a polynomial equation into the conjunction of several constraints, which either take the form of the product or the form of the linear combination with nonnegative coefficients. This section focuses on the former.

Consider the following set of real vectors

$U := \{(x_1, x_2, x_3) : x_1 = x_2 \cdot x_3 \wedge x_1, x_2, x_3 \in [0, 1]\}$

It was proven in lemma 8 of [Shitov, 2016] that the set of factorizations of the following incomplete matrix P of size 9 is strongly equivalent to U

$$P = \left(\begin{array}{ccccccccccc}
x_1 & 0 & 0 & 0 & 0 & x_2 & 1 & 1 & 1 & 1 & 1 \\
0 & & & & & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & & \mathbf{0} & & & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & & & & & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & & & & & 0 & 0 & 1 & 0 & 0 & 1 \\
x_3 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & & & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & \cdots & & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & & & & \\
0 & 0 & 1 & 1 & 0 & \vdots & & & \mathbf{0} & & \\
0 & 0 & 0 & 1 & 1 & & & & & & \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & & & &
\end{array}\right)$$

Where x_1, x_2, x_3 are variables in $[0, 1]$.

Consider the set of factorizations $fact_+(P, 9)$.

The blocks which are highlighted by the red and blue solid lines are independent. Red blocks have nonnegative rank 4. This leads to the fact that in any particular factorization, there are 8 terms corresponding to the red blocks, 4 for each one, and only one remaining term corresponding to the blue block. In addition, dashed red lines indicate the regions, outside of which every block is independent with the inside block which is highlighted by the solid red line. This leads to the fact that the terms which correspond to the particular red block can be positive only inside the dashed red line around said red block.

The term which corresponds to the blue block is of rank 1. It must have the following form, for some constants a, b.

$$P = \left(\begin{array}{ccccccccccc}
\boxed{a\cdot b} & 0 & 0 & 0 & 0 & \multicolumn{2}{c}{\boxed{a \quad a}} & 0 & 0 & 0 & 0 \\
0 & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & & \multicolumn{2}{c}{\mathbf{0}} & & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\
\boxed{b} & 0 & 0 & 0 & 0 & \multicolumn{2}{c}{\boxed{1 \quad 1}} & 0 & & & 0 \\
 & & & & & & & & \multicolumn{2}{c}{\cdots} & \\
\boxed{b} & 0 & 0 & 0 & 0 & \multicolumn{2}{c}{\boxed{1 \quad 1}} & 0 & & & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & & & & \\
0 & 0 & 0 & 0 & 0 & \multicolumn{2}{c}{\vdots} & & \multicolumn{2}{c}{\mathbf{0}} & \\
0 & 0 & 0 & 0 & 0 & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & & & &
\end{array}\right)$$

The terms corresponding to the red blocks then must sum to

$$\begin{array}{ccccc}
1-a & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 1
\end{array}
\qquad \text{and} \qquad
\begin{array}{ccccc}
1-b & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 1
\end{array}$$

As there are only 4 terms corresponding to each red block, using the result about the $\mathcal{B}_0$ matrix, we conclude that $1-a, 1-b \in [0,1]$ and so $a, b \in [0,1]$, and that the form of these terms is uniquely determined by the values of a and b.

The only term which can be positive in the position of x_1, x_2, x_3 is the one which corresponds to the blue block.

This means that a particular factorization of P of size 9 is the factorization of the completion of S where $x_2 = a \in [0,1]$, $x_3 = b \in [0,1]$ and $x_1 = a \cdot b \in [0,1]$. This factorization is completely determined by the values of x_2 and x_3. Obtaining the values x_1, x_2 and x_3 can be considered as a permutation followed by a projection. These values form a vector which belongs to the set U which was defined at the beginning. At the same time, given the values which satisfy the conditions from the definition of U, it is possible to construct the unique factorization of P of size 9 in polynomial time, and the process of construction will not include irrationality.

We can conclude that $fact_+(P, 9) \sim U$

□

If all rows and columns which contain variables are removed, the following matrix remains. The 9 encircled entries are independent, so the nonnegative rank is bounded below by 9.

$$\begin{pmatrix}
 & & & & 1 & \textcircled{1} & 1 & 0 & 0 \\
 & \text{\Huge 0} & & & 0 & 0 & \textcircled{1} & 1 & 0 \\
 & & & & 0 & 0 & 0 & \textcircled{1} & 1 \\
 & & & & 0 & 1 & 0 & 0 & \textcircled{1} \\
1 & 1 & 1 & 1 & \textcircled{1} & 0 & 0 & 0 & 0 \\
\textcircled{1} & 1 & 0 & 0 & 0 & & & & \\
0 & \textcircled{1} & 1 & 0 & 0 & & \text{\Huge 0} & & \\
0 & 0 & \textcircled{1} & 1 & 0 & & & & \\
1 & 0 & 0 & \textcircled{1} & 0 & & & &
\end{pmatrix}$$

This implies that if for one particular variable the rows and the columns which contain this variable are removed, the nonnegative rank of the remaining matrix will be bounded below by 9 as well(the matrix presented above will still be contained inside the remaining matrix). Each variable is present only once, so no 2 same variables share the same row or column. This means that they can be removed, by applying the variable gadget to P 3 times with respect to each variable, and the (variable-free) matrix P' can be obtained, such that $fact_+(P, 9) \sim fact_+(P', q')$ for some q'.

3.3.7 The Final Step

As it was stated before, a particular polynomial equation, restricted to the unit cube, can be transformed into the set of constraints, where one variable is either the product or the linear combination of the other variables. Values, which satisfy the constraints, form the set which is strongly equivalent to the roots of the polynomial in the unit cube.

Each of these constraints has a corresponding nonnegative incomplete matrix. These matrices can be combined together into the one nonnegative incomplete matrix.

Let these matrices be $V_1 \dots V_n$. As it was described in the previous sections, for some constants $q_1 \dots q_n$ $fact_+(V_i, q_i)$ is strongly equivalent to the values which satisfy the i^{th} constraint. We also know that if all rows and columns which contain all variables are removed, the nonnegative rank of the remaining submatrix will still be bounded below by q_i.

Let V be the block-diagonal matrix which contains $V_1 \dots V_n$ as diagonal blocks. All blocks in V are independent. This implies 3 following facts about V

1. $fact_+(V, \sum_i q_i)$ must be strongly equivalent to the set of values which satisfy all the constraints.

2. It must be the case that if all rows and columns which contain variables are removed from V, its rank is still bounded below by $\sum_i q_i$ (as it is the property of each particular V_i block and all blocks of the block-diagonal matrix are independent). This implies that if rows and columns which contain the instances of just one variable are removed, the nonnegative rank will be bounded by $sum_i q_i$ as well.

3. As every block contains just 1 instance of each variable, and blocks occupy distinct rows and columns, no 2 instances of the same variable share a row or a column.

This implies that we can iteratively apply the variable gadget to create variable-free nonnegative matrix V' such that for some q' that fact $fact_+(V, q) \sim fact_+(V', q')$. This proves the fact that checking the existence of the roots of a given polynomial reduces to the task of checking the existence of the nonnegative factorization of a particular size. And because we know the nature of the strong equivalence in this particular case, it is possible to construct particular factorization given particular root and vice versa.

3.4 Examples

For this project, I have created a number of Python scripts which illustrate the proof which was discussed before. These scripts can perform the following tasks.

1. Given a polynomial, create an incomplete nonnegative matrix, such that its factorization is strongly equivalent to the roots of the given polynomial inside the unit cube.

2. Given an incomplete matrix, apply a variable gadget with respect to a particular variable.

3. Given a particular root of the polynomial, create the corresponding factorization of the nonnegative matrix for this polynomial.

4. Given particular exact factorisation of the corresponding nonnegative matrix, deduce the corresponding root of the polynomial.

The code is published at https://github.com/ninextycode/finalYearProjectNMF

In this section, I am going to present matrices which were created by said scripts. (Zero entries are omitted for clarity)

Example 1, simple equation

Consider the polynomial equation $2 \cdot x - 1 = 0$.

For a simple equation like this, it is possible to present both a corresponding incomplete matrix M and a corresponding variable-free matrix G, obtained by applying a variable gadget to M.

The matrices are constructed such that

$fact_{+}(M, 10) \sim fact_{+}(G, 33) \sim \{x : 2x - 1 = 0 \land x \in [0, 1]\}$

Note that, as the proof dictates, the polynomial equation was initially transformed into a condition $L = 2x \land L = 1$ where a new variable $L \in [0, 3]$ was introduced.

Incomplete matrix:

$$M = \begin{pmatrix} \boxed{\begin{matrix} x & & & & & 1 \\ \scriptstyle L & & & & & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & & & \\ & & 1 & 1 & & \\ & & & 1 & 1 & \\ & 1 & & & 1 & \end{matrix}} & \\ & \boxed{\begin{matrix} 1 & & & & & 1 \\ \scriptstyle L & & & & & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & & & \\ & & 1 & 1 & & \\ & & & 1 & 1 & \\ & 1 & & & 1 & \end{matrix}} \end{pmatrix}$$

The red block corresponds to the condition $L = 2x$ and the blue block corresponds to the condition $L = 1$.

Variable-free matrix:

$$G = \begin{pmatrix}
\boxed{2} &&&&& 1 &&&&& &&&&& &&&&& &&&&& &&&&& 1 \\
\boxed{4} &&&&& 2 &&&&& &&&&& & 1 \\
1&1&1&1&1&1 \\
1&1&1 \\
&&1&1 \\
&&&1&1 \\
&1&&&1 \\
&&&&& & 1 &&&&& 1 \\
&&&&& & \boxed{4} &&&&& 1 &&&&& &&&&& 1 \\
&&&&& & 1&1&1&1&1&1 \\
&&&&& & 1&1&1 \\
&&&&& &&& 1&1 \\
&&&&& &&&& 1&1 \\
&&&&& && 1 &&& 1 \\
&&&&& &&&&& && 1&1 \\
&&&&& &&&&& &&& 1&1 \\
&&&&& &&&&& &&&& 1&1 \\
&&&&& &&&&& && 1 &&& 1 & 1 &&&&& 1 \\
1 &&&&& & 1 &&&&& & 1&1&1&1&1 &&&&& 1 \\
1 &&&&& &&&&& &&&&& & 1 & \frac{3}{4}&\frac{3}{4}&\frac{3}{4}&\frac{3}{4} \\
&&&&& &&&&& &&&&& & \frac{3}{4}&\frac{3}{4}&\frac{3}{4} \\
&&&&& &&&&& &&&&& &&& \frac{3}{4}&\frac{3}{4} \\
&&&&& &&&&& &&&&& &&&& \frac{3}{4}&\frac{3}{4} \\
&&&&& &&&&& &&&&& && \frac{3}{4} &&& \frac{3}{4} \\
&&&&& & 1 &&&&& &&&&& &&&&& 1 & \frac{3}{4}&\frac{3}{4}&\frac{3}{4}&\frac{3}{4} \\
&&&&& &&&&& &&&&& &&&&& & \frac{3}{4}&\frac{3}{4}&\frac{3}{4} \\
&&&&& &&&&& &&&&& &&&&& &&& \frac{3}{4}&\frac{3}{4} \\
&&&&& &&&&& &&&&& &&&&& &&&& \frac{3}{4}&\frac{3}{4} \\
&&&&& &&&&& &&&&& &&&&& && \frac{3}{4} &&& \frac{3}{4} \\
&&&&& &&&&& &&&&& &&&&& &&&&& & 1&1 \\
&&&&& &&&&& &&&&& &&&&& &&&&& && 1&1 \\
&&&&& &&&&& &&&&& &&&&& &&&&& &&& 1&1 \\
&&&&& &&&&& &&&&& &&&&& &&&&& & 1 &&& 1 & 1 \\
1 &&&&& &&&&& &&&&& &&&&& &&&&& & 1&1&1&1&1 \\
1 &&&&& &&&&& &&&&& &&&&& &&&&& &&&&& 1 & \frac{1}{2}&\frac{1}{2}&\frac{1}{2}&\frac{1}{2} \\
&&&&& &&&&& &&&&& &&&&& &&&&& &&&&& \frac{1}{2}&\frac{1}{2}&\frac{1}{2} \\
&&&&& &&&&& &&&&& &&&&& &&&&& &&&&& && \frac{1}{2}&\frac{1}{2} \\
&&&&& &&&&& &&&&& &&&&& &&&&& &&&&& &&& \frac{1}{2}&\frac{1}{2} \\
&&&&& &&&&& &&&&& &&&&& &&&&& &&&&& & \frac{1}{2} &&& \frac{1}{2}
\end{pmatrix}$$

By applying the variable gadget, the blue block was constructed and variables were replaced by the constants indicated by the black blocks. The variable gadget allows us to remove variables, as it "encodes" / "hides" the variables in itself.

Example 2, longer equation

Consider the polynomial equation $x \cdot y - 3x + 1 = 0$.

It is possible to use Python scripts to construct an incomplete matrix M and a variable-free nonnegative matrix G such that

$fact_{+}(M, 24) \sim fact_{+}(G, 75) \sim \{(x, y) : x \cdot y - 3x + 1 = 0 = 0 \land x, y \in [0, 1]\}$

Note that, as the proof dictates, the polynomial equation was initially transformed into a condition $L = x \cdot y = v \land L = v + 1 \land L = 3x$ where new variable $v \in [0, 1]$, $L \in [0, 5]$ were introduced.

For an equation like this, a corresponding variable-free matrix would be too large to be presented, so only an incomplete matrix is presented below.

Incomplete matrix:

$$M = \left(\begin{array}{cccccccccccccccccccccccccccc}
v & & & & & x & 1 & 1 & 1 & 1 & 1 & & & & & & & & & & & & & & & & & \\
& & & & & & 1 & 1 & 1 & & & & & & & & & & & & & & & & & & & \\
& & & & & & & & 1 & 1 & & & & & & & & & & & & & & & & & & \\
& & & & & & & & & 1 & 1 & & & & & & & & & & & & & & & & & \\
& & & & & & & 1 & & & 1 & & & & & & & & & & & & & & & & & \\
y & & & & & 1 & 1 & & & & & & & & & & & & & & & & & & & & & \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & & & & & & & & & & & & & & & & & & & & & \\
1 & 1 & 1 & & & & & & & & & & & & & & & & & & & & & & & & & \\
& & 1 & 1 & & & & & & & & & & & & & & & & & & & & & & & & \\
& & & 1 & 1 & & & & & & & & & & & & & & & & & & & & & & & \\
& 1 & & & 1 & & & & & & & & & & & & & & & & & & & & & & & \\
& & & & & & & & & & & v & & & & & & & & & 1 & & & & & & & \\
& & & & & & & & & & & 1 & & & & & & & & & & 1 & & & & & & \\
& & & & & & & & & & & L & & & & & & & & & 1 & 1 & & & & & & \\
& & & & & & & & & & & 1 & 1 & 1 & 1 & 1 & & & & & 1 & & & & & & & \\
& & & & & & & & & & & 1 & 1 & 1 & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & 1 & 1 & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & 1 & 1 & & & & & & & & & & & & \\
& & & & & & & & & & & & 1 & & & 1 & & & & & & & & & & & & \\
& & & & & & & & & & & 1 & & & & & 1 & 1 & 1 & 1 & & 1 & & & & & & \\
& & & & & & & & & & & 1 & & & & & 1 & 1 & & & & & & & & & & \\
& & & & & & & & & & & & & & & & & 1 & 1 & & & & & & & & & \\
& & & & & & & & & & & & & & & & & & 1 & 1 & & & & & & & & \\
& & & & & & & & & & & & & & & & 1 & & & 1 & & & & & & & & \\
& & & & & & & & & & & & & & & & & & & & & & x & & & & & 1 \\
& & & & & & & & & & & & & & & & & & & & & & L & & & & & 3 \\
& & & & & & & & & & & & & & & & & & & & & & 1 & 1 & 1 & 1 & 1 & 1 \\
& & & & & & & & & & & & & & & & & & & & & & 1 & 1 & 1 & & & \\
& & & & & & & & & & & & & & & & & & & & & & & & 1 & 1 & & \\
& & & & & & & & & & & & & & & & & & & & & & & & & 1 & 1 & \\
& & & & & & & & & & & & & & & & & & & & & & & 1 & & & 1 &
\end{array}\right)$$

The red block corresponds to the condition $v = xy$, the blue block corresponds to the condition $L = v + 1$ and the green block corresponds to the condition $L = 3x$

4 Approximating Algorithms

4.1 Introduction

In this section, I will provide an overview of some of the algorithms which, given the nonnegative matrix V and an integer n, construct an approximate nonnegative factorization of V of size n. For this project, I have implemented and analysed the performance of several algorithms. This analysis is presented in the next section.

Algorithms, which are used to find an approximate solution, generally have the following form:

- initialise W and H
- for each iteration
  - minimise cost function with respect to W, update W
  - minimise cost function with respect to H, update H

There is a tradeoff between fast iteration steps and a good optimisation inside one iteration step. Because of this reason, comparing the performance of algorithms with respect to iterations and not with respect to time may be meaningless.

First, I present some background information on the algorithms which I have implemented.

4.2 Background Information

4.2.1 Multiplicative Algorithm [Lee and Seung, 2001a]

The multiplicative method was one of the first algorithms published[Lee and Seung, 1999]. Later, a slightly modified version of this algorithm, together with the proof of the fact that at each update step the objective function does not increase, was published[Lee and Seung, 2001a].

The multiplicative method updates elements of W and H, multiplying them by coefficients which are computed from V and W, H from the previous step.

This method is similar to the expectation maximisation algorithm from statistics. It also can be presented as a gradient descent with a varying step size.

4.2.2 Projected Gradient Descent Method [Lin, 2007]

The paper [Lin, 2007] suggested a projective gradient descent method. It is based on the application of the gradient descent to minimise the cost function with respect to W, then changing the negative entries to zero(project the result to the set of nonnegative matrixes) and update W, then do the same for H.

This algorithm tries different gradient step and chooses the optimal to achieve faster convergence.

4.2.3 Nesterov Optimal Gradient Descent Method [Guan et al., 2012]

The paper [Guan et al., 2012] suggested an algorithm which is similar to the projective gradient descent. But in this case, instead of just using simple gradient descent and then projecting the result to iteratively find optimal W and H, it uses Nesterov gradient descent method [Nesterov, 1983]. The idea of this algorithm is similar to adding "inertia" to the gradient descent(but not exactly the same).

In theory, Nesterov gradient descent algorithm achieves the error decrease rate at $O(\frac{1}{k^2})$, which is the optimal rate for the first-order optimization[Bubeck et al., 2015].

4.2.4 Bayesian Method [Schmidt et al., 2009]

The Bayesian approach to the NMF was proposed by [Schmidt et al., 2009].

Elements V_{ij} are assumed to be independent and normally distributed. The variance parameter is assumed to follow the inverse gamma distribution. Assumed distribution for W and H is initially exponential, but this does not persist after parameters are updates.

The paper suggests using either iterated conditional modes to find a maximum a posteriori estimation, or to use the Gibbs sampler. In both cases, the algorithm iteratively updates the parameters of the distribution of W, then the variance, and then the parameters of the distribution of H.

The advantage of this approach is that it becomes possible to compute the likelihood function, confidence intervals and to use the Bayesian information criterion to compare the models with different inner dimensions of WH.

After implementing this algorithm, a major downside was discovered. This method lacks robustness. It often produces solutions where some columns of W(or some rows of H) are zero. For this reason, I have not included this algorithm into the complexity analysis.

4.2.5 Active Set Method [Kim and Park, 2008a]

I have considered this algorithm too complex to be implemented for the scope of this project. However, it is suggested in [Kim and Park, 2008a] that this algorithm has a good performance, therefore I have included its overview here.

It was stated by [Kim and Park, 2008a] that the gradient methods may suffer from slow convergence due to a possible zigzag phenomenon. It was suggested to use the alternating nonnegativity constrained least squares and the active set method. For each iteration step, this method finds the W and H which minimise the objective function.

It considers each column of H and of W^T separately. To find the solution for the column, it uses (Karush-Kuhn-Tucker) necessary conditions [Kuhn, 1951] for nonnegative least squares.

Initially, for each column, the elements which are 0 considered to form the "active" set, and positive elements form the "passive" set.

Iteratively, an element from an "active" set, which, if changed, can minimise the objective function, is moved to the "passive" set. Then, the solution to the unconstrained(no nonnegativity constraint) problem with respect to the elements of the "passive" set is computed. If the obtained solution is the minimum of the constrained problem, it should have all the elements positive. If it is not the case, then the solution is updated, negative elements are removed and sets are rearranged until the "active" and "passive" sets contain elements with certain properties, defined in the cited paper. Then algorithm makes another iteration, as long as the "active" set is not empty.

5 Performance Analysis of Algorithms

5.1 Introduction

This section compares the performance of the multiplicative algorithm [Lee and Seung, 2001b], projected gradient algorithm [Lin, 2007] and Nesterov gradient algorithm [Guan et al., 2012]. The comparison is done on 5 sets of random data and 3 sets of real-life data.

5.2 Development

Due to the fact that Python programming language is relatively simple but at the same time contains all the necessary libraries to implement algorithms on CPU and GPU, development was done in this language. Development was conducted according to the iterative software development model.

Algorithms were initially implemented for a CPU using the following plan

For each algorithm

- Study relevant paper
- Plan the structure

- Implement the functions

- Test the correctness

- Modify if necessary

After the correctness on CPU was established, algorithms were modified so that they can be run on GPU. PyTorch library was used to make it possible to run algorithms on GPU.

As the final step, the performance of algorithms was compared. The Bayesian algorithm was implemented, but due to the poor quality of the produced solutions, it was not included in the comparison.

5.3 Hardware

Performance was tested on the Amazon p2.xlarge instance with the following characteristics:

CPU: Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz

GPU: Tesla K80 12 GB

5.4 Performance Comparison

5.4.1 Random Data

Random data was generated by the following algorithm:

To generate a random matrix of size $(m \times n)$, for which the factorization of size k will be considered, the following steps were taken.

1. Generate matrices $A \in \mathbb{R}_{+}^{m\times k}$, $B \in \mathbb{R}_{+}^{k\times n}$, where each entry is sampled from a uniform distribution over the interval $[0, 1]$.

2. Multiply A by B to obtain the final random matrix $V = AB$, which, together with k, will be used as an input to the nonnegative factorization algorithm.

Five pairs of datasets were generated. The matrices had the following dimensions: $(150, 30)$, $(500, 100)$, $(1500, 300)$, $(5000, 1000)$, $(15000, 3000)$. For each of these shapes, 2 matrices were generated, one with nonnegative rank $\approx 0.1\cdot$ number of columns, and one with nonnegative rank $\approx 0.8\cdot$ number of columns.

As algorithms only find approximate solutions, improving approximation with every step, they can theoretically run forever. To compare their performance, we can compare the time it takes algorithms to compute an approximation of a particular quality. Frobenius norm was used as a cost function to evaluate the quality of the approximation.

From the comparison of performance on the random data, the following conclusions were obtained

- In general, Nesterov algorithm is faster than projected gradient algorithm, which is, in turn, faster than the multiplicative

- During the first iterations, the multiplicative algorithm shows the best performance, but its rate of improvement slows down quickly

- The advantage of a faster algorithm is greater for a larger data

- For the large matrices, GPU versions show better performance, and the performance increase from the GPU neglects the corresponding overhead. This is not the case for smaller matrices(for them, however, the solution is obtained relatively quickly in any case)

- The performance increase from the GPU is more significant for a smaller factorization size

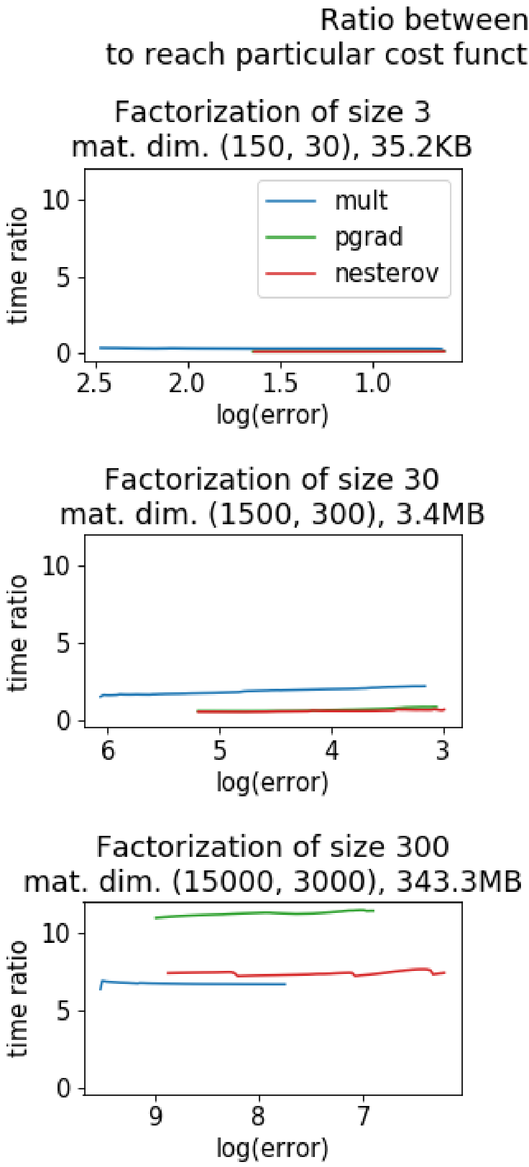
- The multiplicative algorithm benefits the most from being run on GPU

The results from testing on real-life data were in line with these findings.

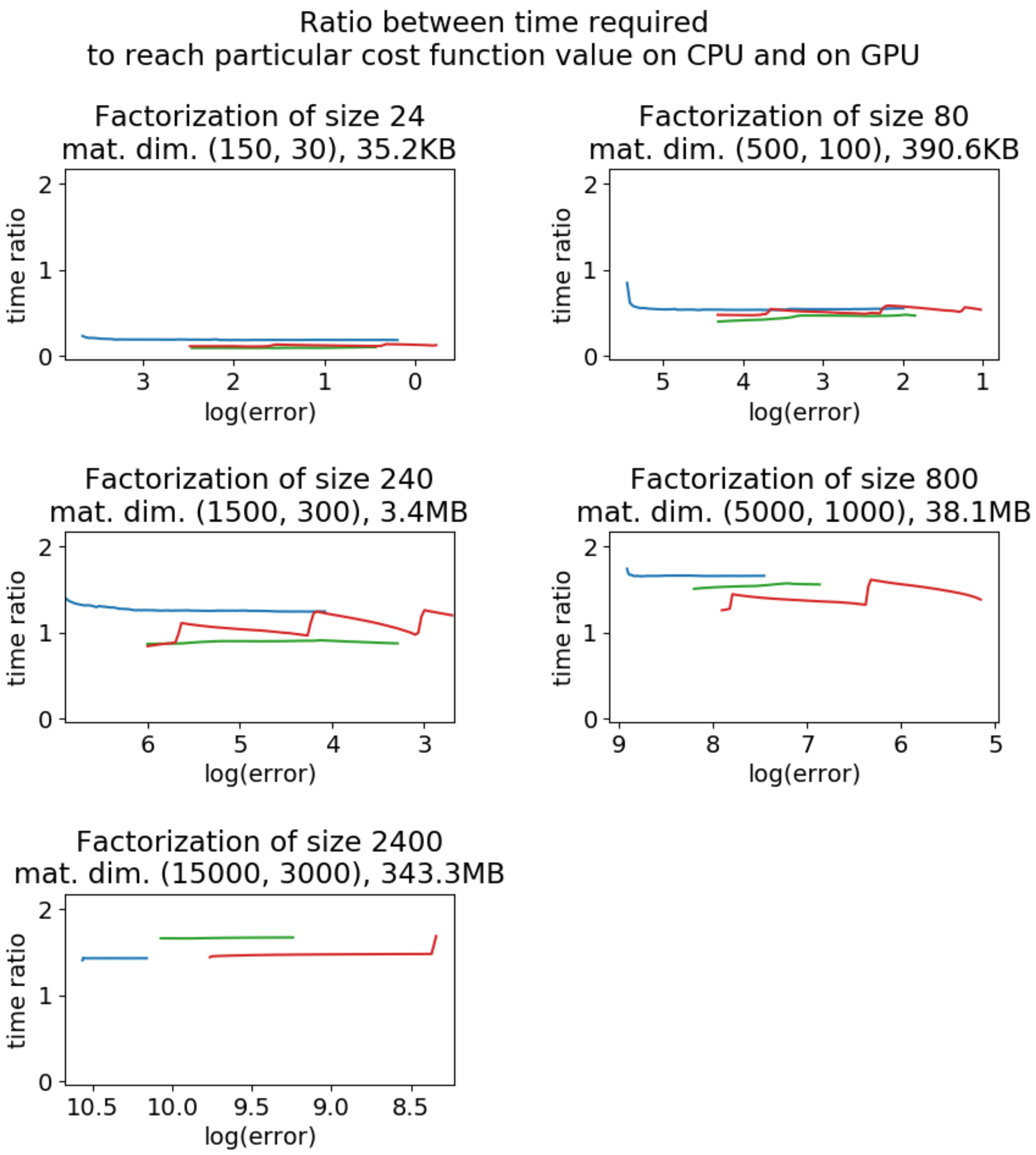
CPU vs GPU

The following graphs show how faster are algorithms on GPU compared to algorithms on CPU. The y-axis corresponds to the logarithm of the cost function value. The x-axis corresponds to the ratio between the time it took an algorithm on CPU to achieve a particular value of the cost function and the time it took an algorithm on GPU to achieve the same value of the cost function. Labels *mult*, *pgrad*, *nesterov* correspond to the multiplicative, projected gradient and Nesterov gradient algorithms.

First 5 graphs correspond to factorizations of the (relatively) small size.



Following 5 graphs correspond to factorizations of the large size.



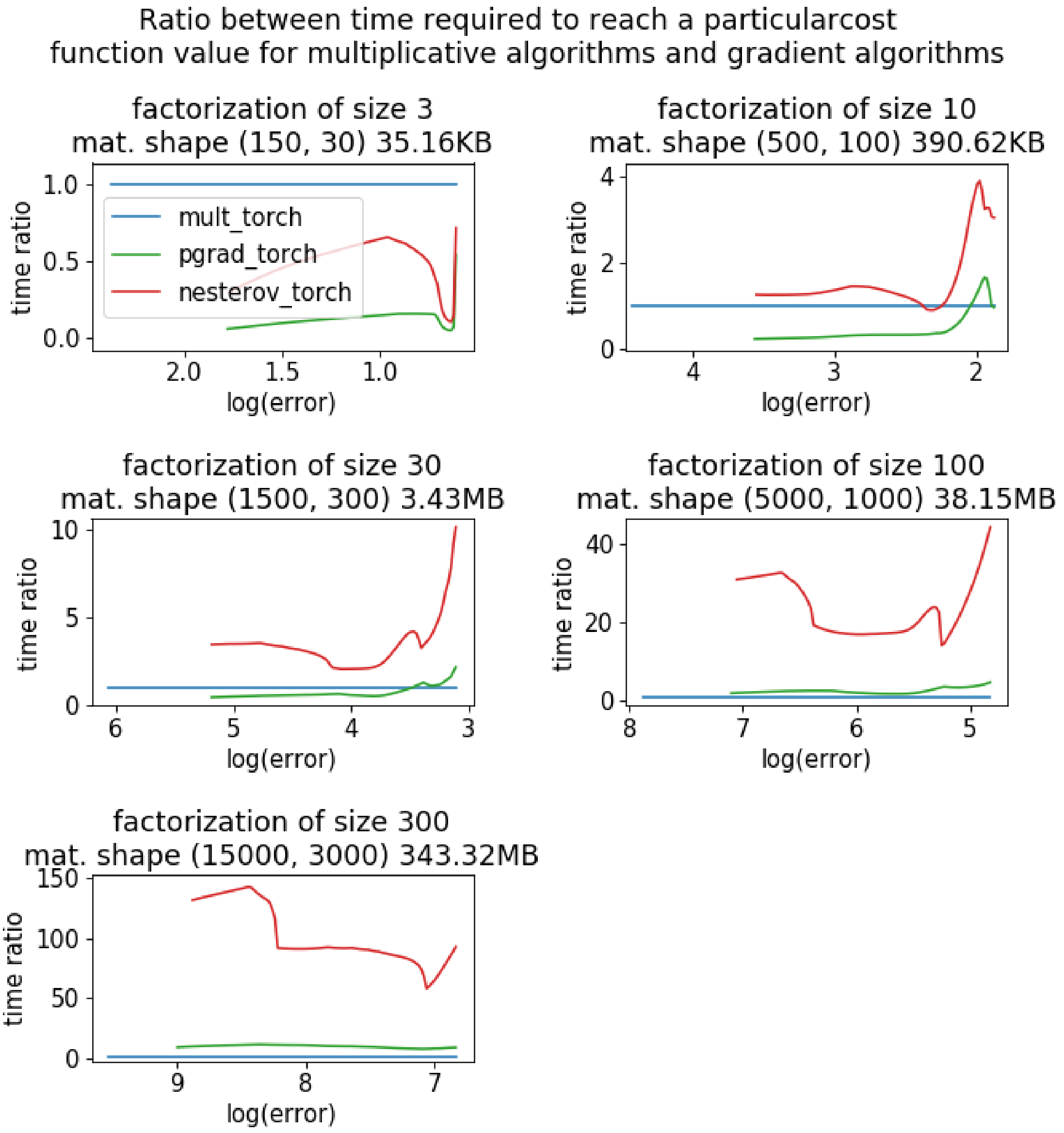
For the small input, the ratio is less than 1, which means that algorithms on CPU are faster. This can be explained by the overhead of working with the PyTorch library. However, it is not the case with larger input. For larger input matrices, GPU versions outperform CPU ones.

It can be seen from the previous graphs that the multiplicative algorithms benefit the most from being run on GPU, and projected/Nesterov gradient descent algorithms follow.

It is also possible to compare the performance gain provided by GPU for smaller and larger factorisation sizes. It can be seen that performance gain for the factorizations of the small size (relative to the number of columns) is greater than for the larger size.

Multiplicative algorithm vs gradient methods

Now compare the projected and Nesterov gradient descent algorithms to the multiplicative algorithm. On the following graphs, y-axis corresponds to the ratio between the time it took for a multiplicative algorithm to achieve a particular value of the cost function and the time it took every other algorithm to achieve the same value of the cost function. (so the line corresponding to the multiplicative algorithm is a constant 1). No patterns related to the factorisation size or to the choice between CPU/GPU were discovered. Because of this fact, only results obtained for small ($0.1\cdot$ number of columns) factorisation sizes are presented here. Postfix *_torch* indicates the fact that an algorithm was run on GPU with the help of PyTorch library.

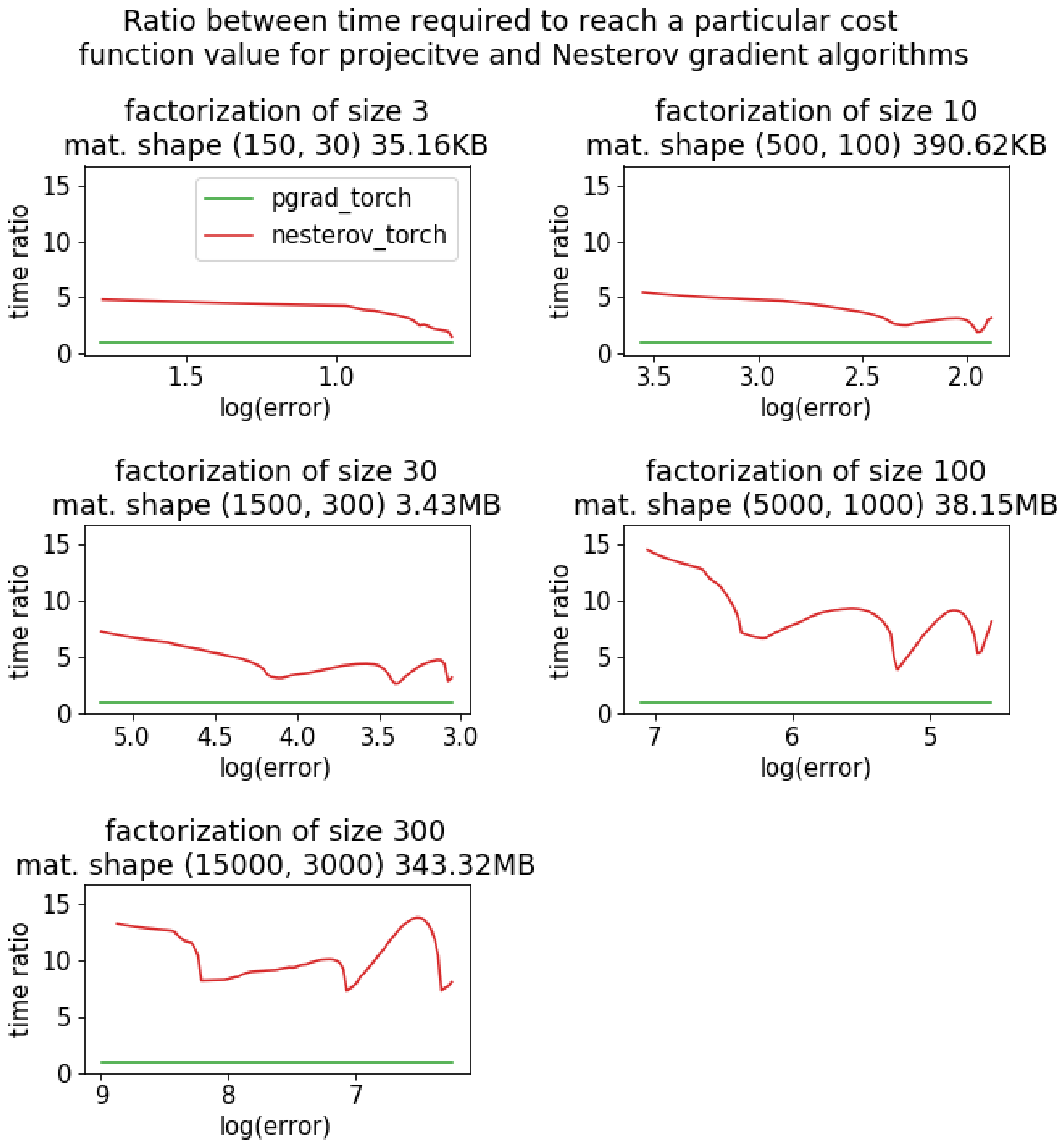


It can be noticed that the advantage of the gradient algorithms over multiplicative grows as the value of the cost function gets smaller, which means that the advantage of the gradient algorithms is greater if an approximation of greater accuracy is required. (This corresponds to the upward trends on the graphs).

It is also the case that the advantage of the gradient algorithms over the multiplicative algorithm is larger for the lager inputs.

Projected gradient algorithm vs Nesterov gradient algorithm

Finally, compare the projected and Nesterov gradient descent algorithms. As in the previous graph, y-axis corresponds to the ratio between the time it took for the Nesterov gradient descent algorithm to achieve a particular value of the cost function and the time it took the projected gradient descent algorithm to achieve the same value of the cost function.



Similar to the previous comparison, it is the case that the advantage of the Nesterov algorithms over the multiplicative algorithm is larger for the lager input, as it can be seen from the following graphs. In contrast to the relationship between multiplicative and gradient algorithms, there seems to be no connection between the quality of factorization(expressed by the value of the cost function) and the advantage of the

Nesterov algorithm.

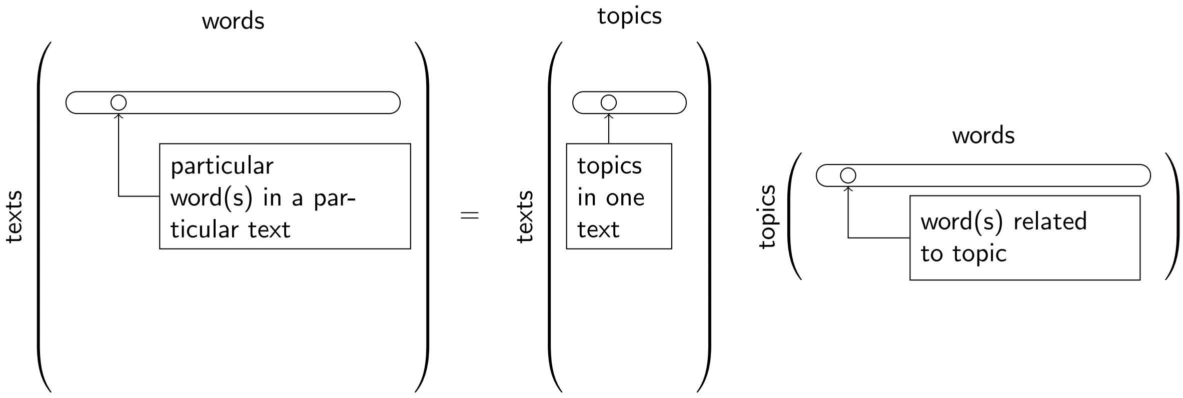
5.4.2 Text Database

For this project, the text database Reuters-21578, Distribution 1.0 was used. The resulting matrix takes 1.9GB of memory. Scikit-learn library was used to transform the text into a matrix

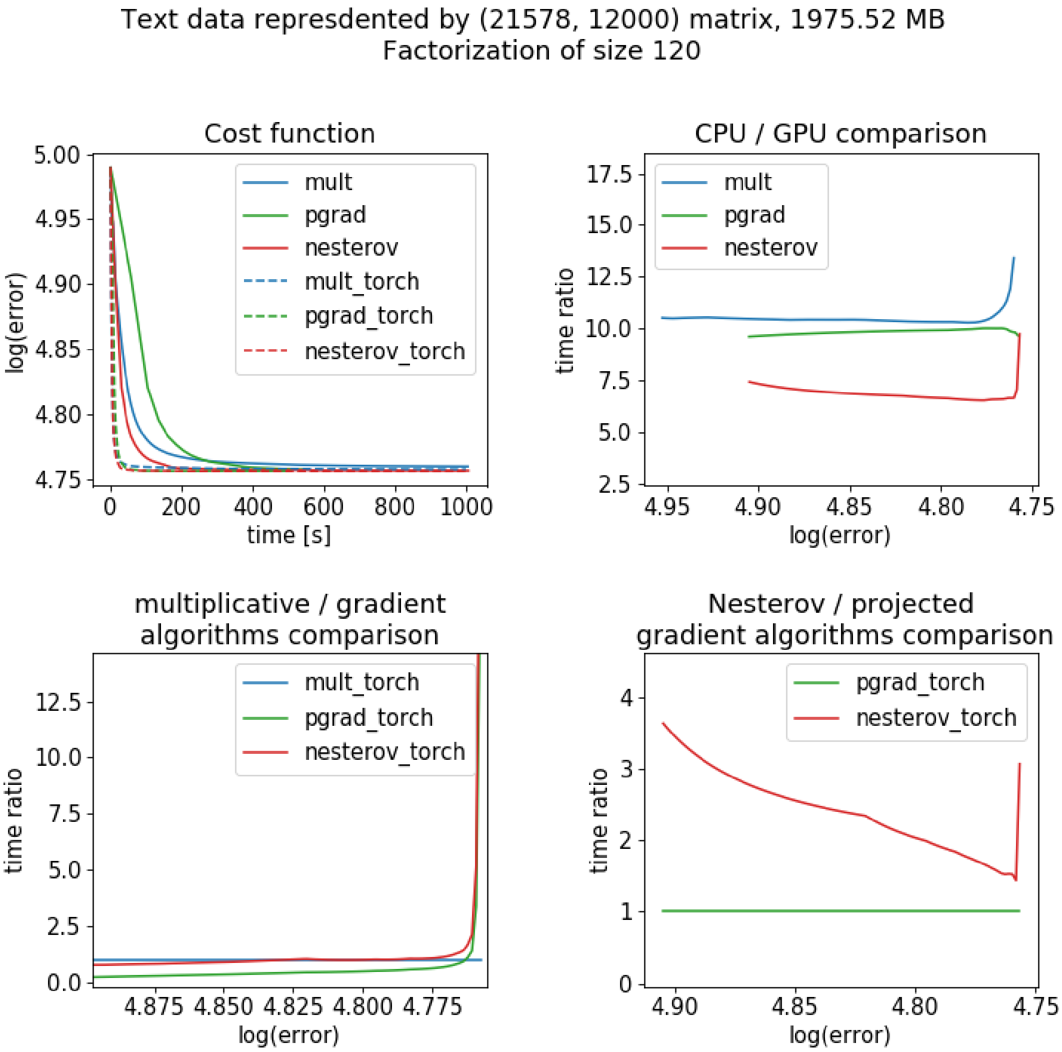
The text database is represented as $V \in \mathbb{R}^{n \times m}$, where each row corresponds to a particular text, each column corresponds to a particular word(or a set of words, is hashing was used), and the value in the position (i, j) is proportional to the number of words corresponding to j in an article i, (and it can be inversely proportional to the number of articles where such words occur).

For the nonnegative matrix factorization of the form $V = WH$ of size r, r columns of W and r rows of H correspond to r latent topics.

Rows of W correspond to the distribution of words in texts, and columns of H correspond to the distributions of words in topics.

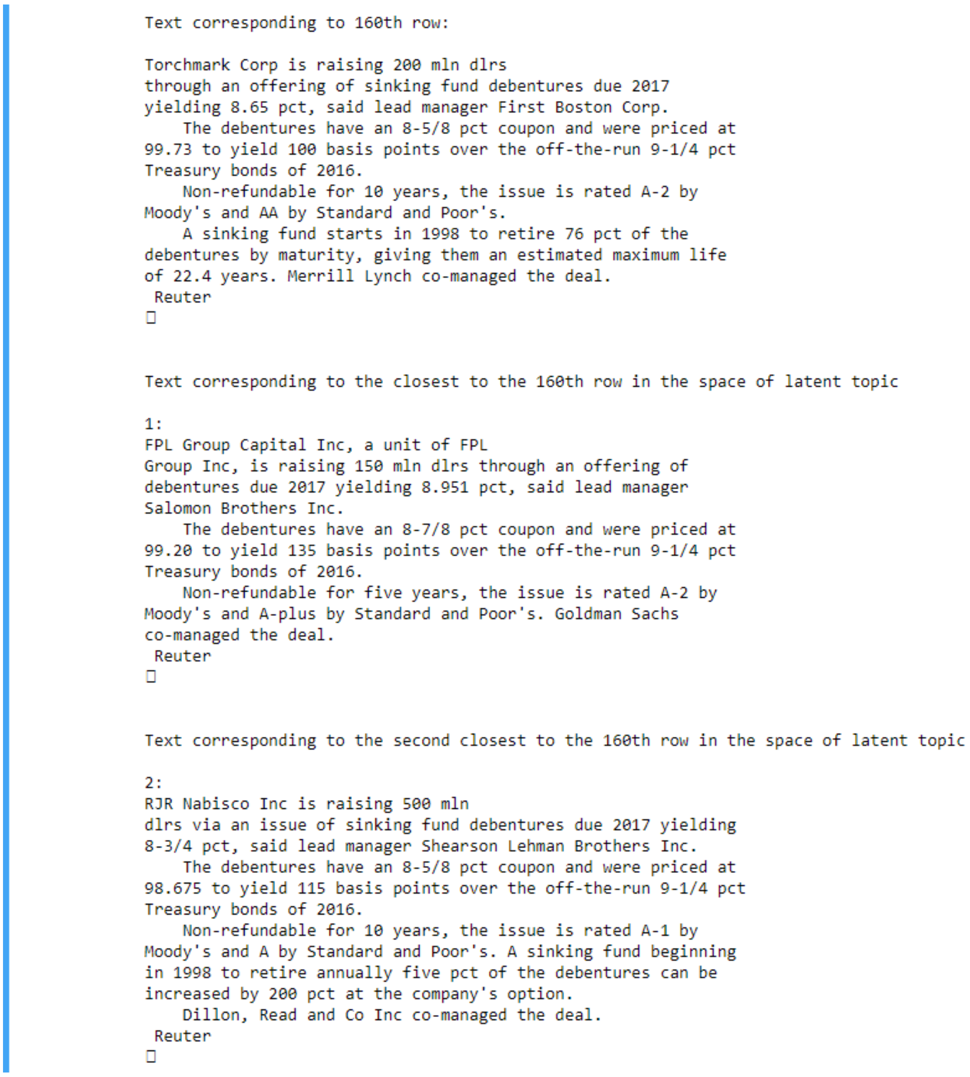


Following graphs show the performance of algorithms on the text data.



It matches the patterns which were seen on the random data. In particular, gradient algorithms increase their advantage over the multiplicative one as the value of the cost function decreases. It is also the case that the Nesterov gradient algorithm has an advantage over the projected gradient algorithm. Every algorithm gets a speedup from being run on the GPU. The multiplicative algorithm gets a greater speedup than gradient algorithms. This will be the case for other 2 sets of real-life data as well.

To illustrate the factorization, we can take a particular text, and consider the corresponding row of matrix W. Let this row be the 160^{th} row w. This row represents the text in the space of topics. If we search for other rows which have a small Euclidian distance to w, the texts corresponding to these rows will be about the same or similar topics as the text which corresponds to w. This can be seen in the following picture.



```
Text corresponding to 160th row:

Torchmark Corp is raising 200 mln dlrs
through an offering of sinking fund debentures due 2017
yielding 8.65 pct, said lead manager First Boston Corp.
    The debentures have an 8-5/8 pct coupon and were priced at
99.73 to yield 100 basis points over the off-the-run 9-1/4 pct
Treasury bonds of 2016.
    Non-refundable for 10 years, the issue is rated A-2 by
Moody's and AA by Standard and Poor's.
    A sinking fund starts in 1998 to retire 76 pct of the
debentures by maturity, giving them an estimated maximum life
of 22.4 years. Merrill Lynch co-managed the deal.
 Reuter
□


Text corresponding to the closest to the 160th row in the space of latent topic

1:
FPL Group Capital Inc, a unit of FPL
Group Inc, is raising 150 mln dlrs through an offering of
debentures due 2017 yielding 8.951 pct, said lead manager
Salomon Brothers Inc.
    The debentures have an 8-7/8 pct coupon and were priced at
99.20 to yield 135 basis points over the off-the-run 9-1/4 pct
Treasury bonds of 2016.
    Non-refundable for five years, the issue is rated A-2 by
Moody's and A-plus by Standard and Poor's. Goldman Sachs
co-managed the deal.
 Reuter
□


Text corresponding to the second closest to the 160th row in the space of latent topic

2:
RJR Nabisco Inc is raising 500 mln
dlrs via an issue of sinking fund debentures due 2017 yielding
8-3/4 pct, said lead manager Shearson Lehman Brothers Inc.
    The debentures have an 8-5/8 pct coupon and were priced at
98.675 to yield 115 basis points over the off-the-run 9-1/4 pct
Treasury bonds of 2016.
    Non-refundable for 10 years, the issue is rated A-1 by
Moody's and A by Standard and Poor's. A sinking fund beginning
in 1998 to retire annually five pct of the debentures can be
increased by 200 pct at the company's option.
    Dillon, Read and Co Inc co-managed the deal.
 Reuter
□
```

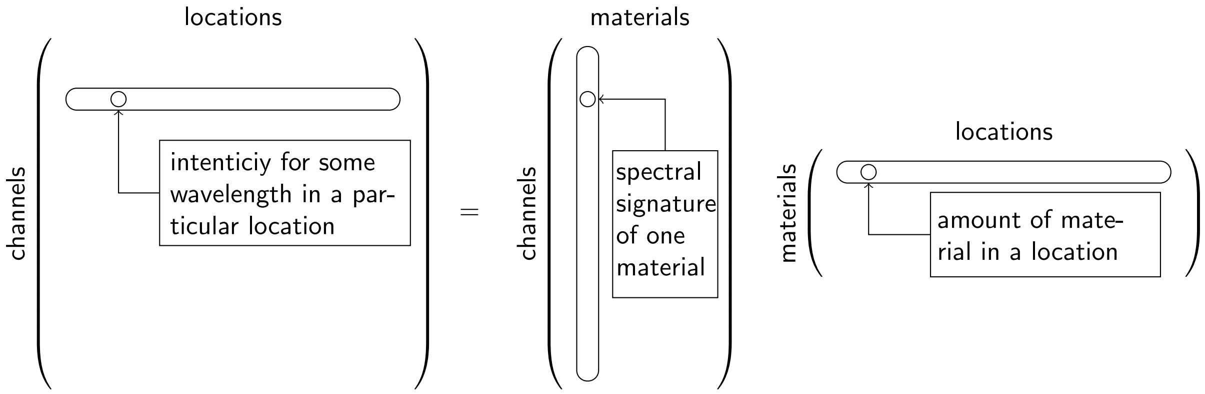
5.4.3 Hyperspectral Image

For this project, the dataset Indian Pine, Test Site 3 from Purdue University Research Repository was used. The image is the hyperspectral photo of the crop fields, where different fields are sown with different crops. The photo contains 220 channels corresponding to different wavelengths. Each channel is represented as a nonnegative matrix M_i with a shape $(145, 145)$. The total image size is 35.3MB.

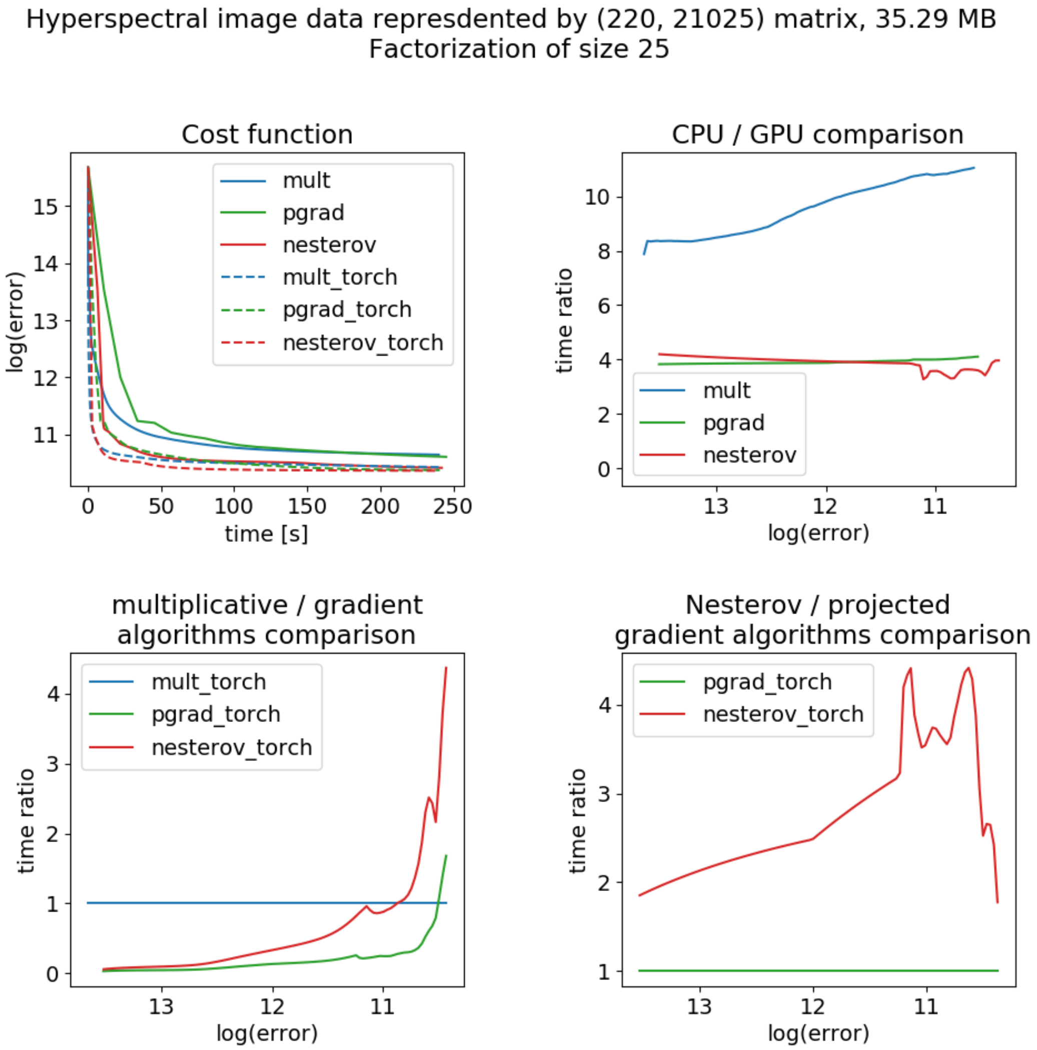
A spectral trace can be modelled as a linear combination of a set of endmembers, weighted by corresponding material quantities.

Given observed spectral traces over an area, $M_1 \dots M_n$, where M_i is a rectangular matrix with m entries which represent intensity values for a certain i^{th} wavelength, we can reshape these matrices into row vectors and concatenate them into matrix $V \in \mathbb{R}_+^{n \times m}$.

This data can be modelled as $V = WH(+noise)$, where the columns of W are the endmembers, and the values of H represent the material quantities in particular locations. The row of H then represents a distribution of a particular material over an area. In the case of the hyperspectral photo of the fields, different crops are corresponding to some of the different materials.

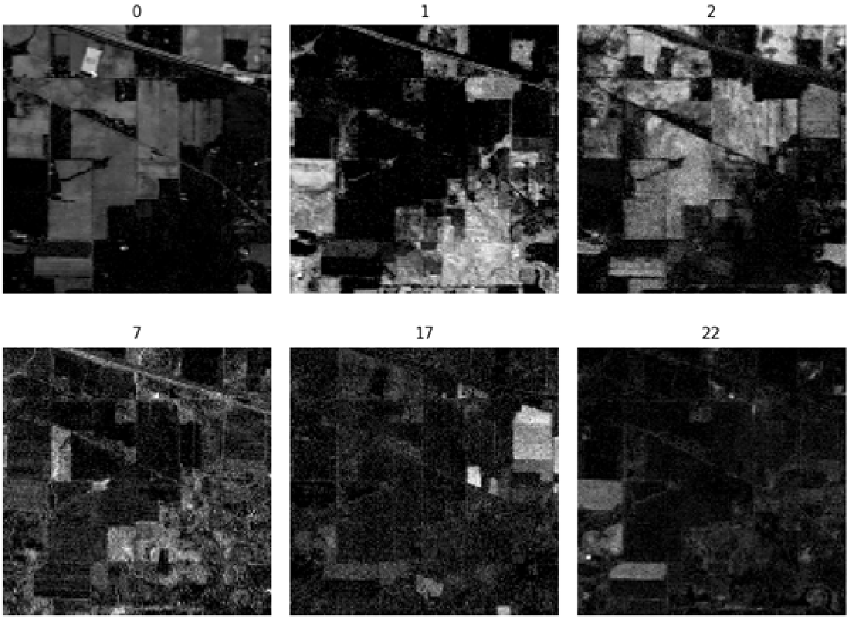

locations
channels
intenticiy for some wavelength in a particular location
=
materials
channels
spectral signature of one material
locations
materials
amount of material in a location

The following graphs show the performance of algorithms on the hyperspectral image data. It matches the patterns which were seen on the random data and on the text database.



To illustrate the results of the algorithms, rows of H can be reshaped into the shape of the original matrices M_i. Then they can be viewed as maps of the distributions of particular materials.

The following images are the results of running a factorization algorithm. They were obtained from 6(out of 25) rows of H. I have selected them such that the distinction between fields of different crops can be easily observed and the noise is minimal.



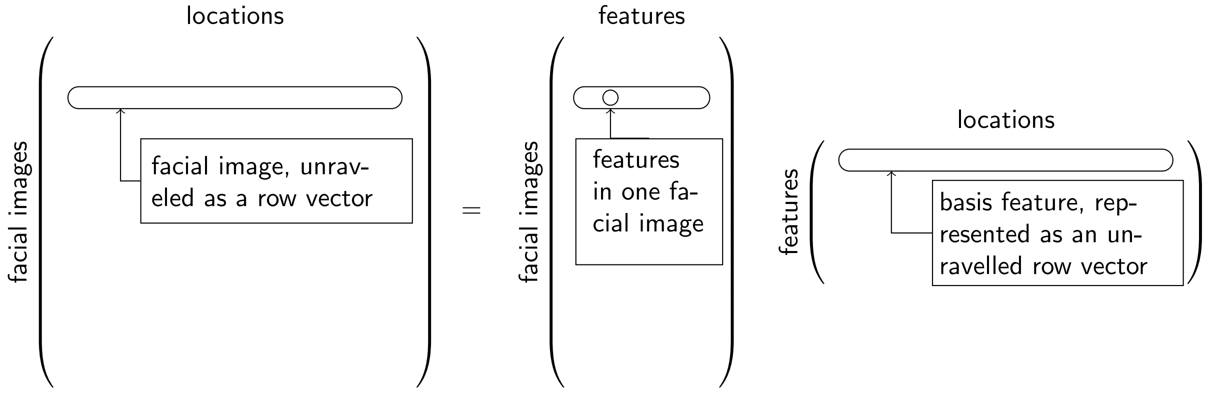
5.4.4 Facial Database

For this project, the database of faces from At&t Laboratories Cambridge was used. The facial images are represented as matrices of shape $(112, 92)$, and the total database size is 31.45MB.

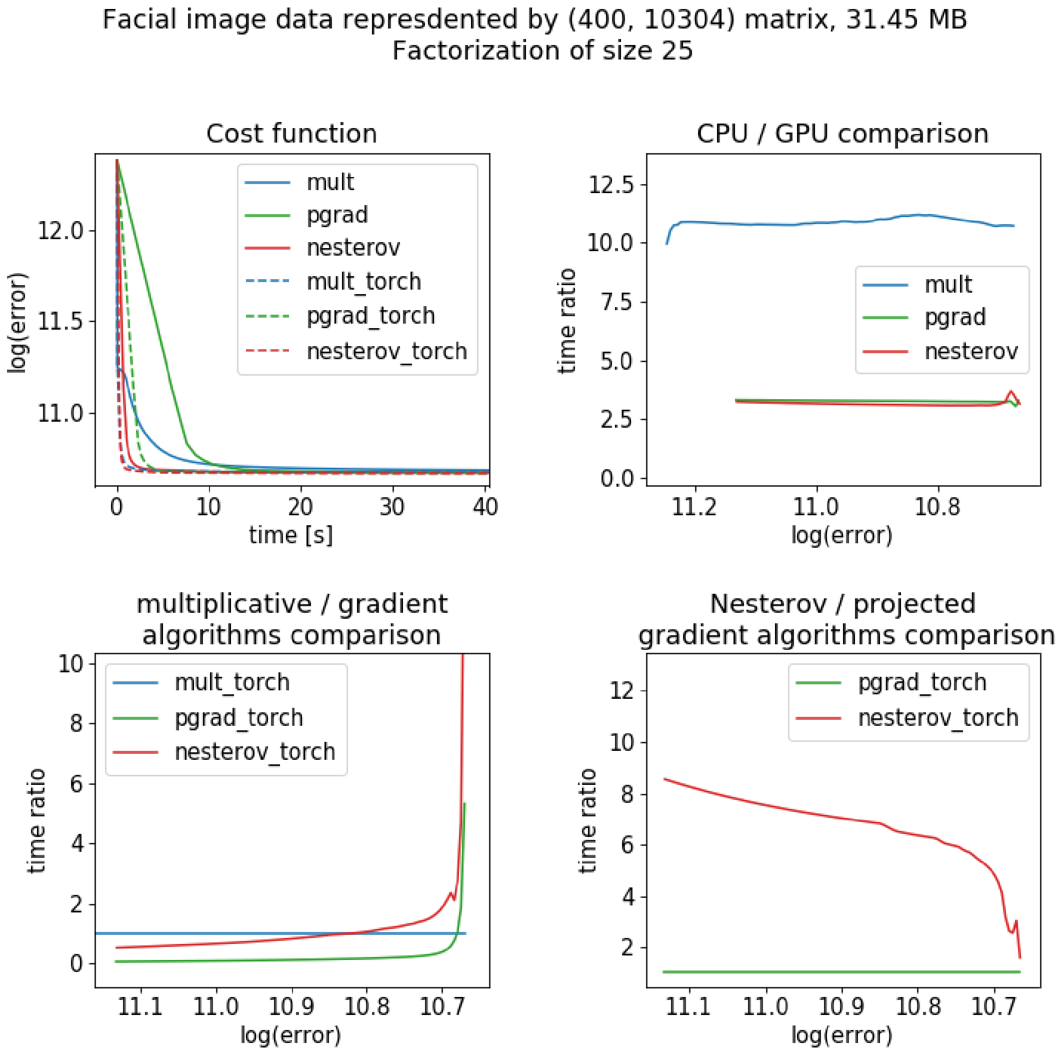
The interpretation of the factorization is similar to the one for the hyperspectral images.

The image database is represented as $V \in \mathbb{R}_+^{n\times m}$. Each row of V corresponds to one of the n facial images.

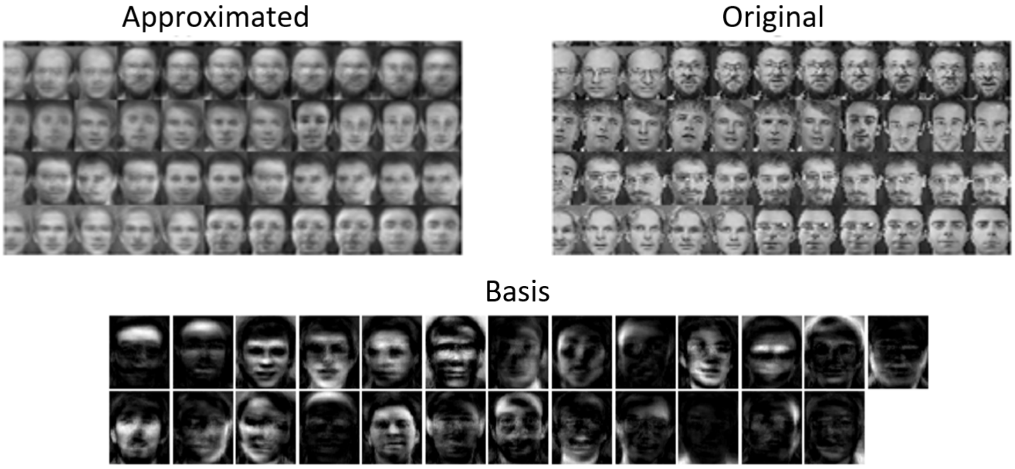
For the nonnegative matrix factorization $V = WH$ of size r, the r columns of H form the basis images. Each basis image corresponds to a particular feature of the facial image. Each row of H is the representation of a particular facial image in the space of the basis images. Facial images and basis images are represented by the rectangular matrices which are unravelled into row vectors.


locations
facial images
facial image, unraveled as a row vector
=
features
facial images
features in one facial image
locations
features
basis feature, represented as an unravelled row vector

The following graphs show the performance of algorithms on the facial image data. It matches the patterns which were seen on the random data and on the text database.



The following image illustrates the factorization. A sample of the images from the database is presented, together with the same facial images as they were reconstructed from their representation of the space of feature images. Images of the extracted features are presented at the bottom. In this case, nonnegative matrix factorization can be considered as a lossy image compression algorithm.



It can be noted that at least some of the basis features correspond to the distinct parts of the faces, which is in agreement with the similar observation made in [Lee and Seung, 1999].

6 Summary

In the scope of this project, the following goals were achieved:

- Nonnegative matrix factorization was introduced. Several applications of the nonnegative matrix factorization were reviewed. This part can help to get a general idea of what is the nonnegative matrix factorization and how it can be used.

- The proof of the complexity of deciding the nonnegative rank was reviewed. Based on the original work [Chistikov et al., 2016], an expanded version of the proof was produced, where unclear or skipped steps were completed. This part can help to understand the details of the original proofs and the connection between nonnegative matrix factorization and roots of polynomials.

- Scripts, illustrating the ideas of the complexity proof [Chistikov et al., 2016] were created. They can be helpful to understand the connection between nonnegative matrix factorization and finding roots of a polynomial. These scripts can as well be a part of a different work which focuses on the nonnegative matrix factorization and/or polynomials. The source code is available online.

- A number of algorithms were successfully implemented to be run on both CPU and GPU. Their performance was compared on a number of real-life and random datasets. Several performance trends were discovered. These trends were consistent over random and real-life datasets. This analysis can be used by people who need to choose which approximate algorithm to use for a practical task. The source code of the algorithms is available online.

References

[Bubeck et al., 2015] Bubeck, S. et al. (2015). Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357.

[Chistikov et al., 2016] Chistikov, D., Kiefer, S., Marušić, I., Shirmohammadi, M., and Worrell, J. (2016). On restricted nonnegative matrix factorization. *arXiv preprint arXiv:1605.07061*.

[Cohen and Rothblum, 1993] Cohen, J. E. and Rothblum, U. G. (1993). Nonnegative ranks, decompositions, and factorizations of nonnegative matrices. *Linear Algebra and its Applications*, 190:149–168.

[Devarajan, 2008] Devarajan, K. (2008). Nonnegative matrix factorization: an analytical and interpretive tool in computational biology. *PLoS computational biology*, 4(7):e1000029.

[Ding et al., 2008] Ding, C., Li, T., and Peng, W. (2008). On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics & Data Analysis*, 52(8):3913–3927.

[Févotte et al., 2009] Févotte, C., Bertin, N., and Durrieu, J.-L. (2009). Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis. *Neural computation*, 21(3):793–830.

[Gillis and Glineur, 2012] Gillis, N. and Glineur, F. (2012). On the geometric interpretation of the nonnegative rank. *Linear Algebra and its Applications*, 437(11):2685–2712.

[Grigor'ev and Vorobjov, 1988] Grigor'ev, D. Y. and Vorobjov, N. N. (1988). Solving systems of polynomial inequalities in subexponential time. *Journal of symbolic computation*, 5(1-2):37–64.

[Guan et al., 2012] Guan, N., Tao, D., Luo, Z., and Yuan, B. (2012). Nenmf: An optimal gradient method for nonnegative matrix factorization. *IEEE Transactions on Signal Processing*, 60(6):2882–2898.

[Hofmann, 1999] Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc.

[Kim and Park, 2008a] Kim, H. and Park, H. (2008a). Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM journal on matrix analysis and applications*, 30(2):713–730.

[Kim and Park, 2008b] Kim, J. and Park, H. (2008b). Sparse nonnegative matrix factorization for clustering. Technical report, Georgia Institute of Technology.

[Kubjas et al., 2015] Kubjas, K., Robeva, E., Sturmfels, B., et al. (2015). Fixed points of the em algorithm and nonnegative rank boundaries. *The Annals of Statistics*, 43(1):422–461.

[Kuhn, 1951] Kuhn, H. (1951). Aw tucker, nonlinear programming. In *Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492.

[Lee and Seung, 1999] Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788.

[Lee and Seung, 2001a] Lee, D. D. and Seung, H. S. (2001a). Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562.

[Lee and Seung, 2001b] Lee, D. D. and Seung, H. S. (2001b). Algorithms for non-negative matrix factorization. In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information Processing Systems 13*, pages 556–562. MIT Press.

[Lin, 2007] Lin, C.-J. (2007). Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779.

[Matousek, 2014] Matousek, J. (2014). Intersection graphs of segments and $\exists\mathbb{R}$. *arXiv preprint arXiv:1406.2636*.

[Nesterov, 1983] Nesterov, Y. (1983). A method for unconstrained convex minimization problem with the rate of convergence o $(1/\text{k}\hat{}\ 2)$. In *Doklady AN USSR*, volume 269, pages 543–547.

[Ozer et al., 2016] Ozer, M., Kim, N., and Davulcu, H. (2016). Community detection in political twitter networks using nonnegative matrix factorization methods. In *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 81–88. IEEE Press.

[Schmidt et al., 2009] Schmidt, M. N., Winther, O., and Hansen, L. K. (2009). Bayesian non-negative matrix factorization. In *International Conference on Independent Component Analysis and Signal Separation*, pages 540–547. Springer.

[Shitov, 2016] Shitov, Y. (2016). A universality theorem for nonnegative matrix factorizations. *arXiv preprint arXiv:1606.09068*.

[Suppes and Zanotti, 1981] Suppes, P. and Zanotti, M. (1981). When are probabilistic explanations possible? *Synthese*, 48(2):191–199.

[Wang et al., 2006] Wang, J., Zhong, W., and Zhang, J. (2006). Nnmf-based factorization techniques for high-accuracy privacy protection on non-negative-valued datasets. In *null*, pages 513–517. IEEE.

[Yannakakis, 1991] Yannakakis, M. (1991). Expressing combinatorial optimization problems by linear programs. *Journal of Computer and System Sciences*, 43(3):441–466.