2009/6

■

Using underapproximations for sparse nonnegative matrix factorization

Nicolas Gillis and François Glineur

CORE
Voie du Roman Pays 34
B-1348 Louvain-la-Neuve, Belgium.
Tel (32 10) 47 43 04
Fax (32 10) 47 43 01
E-mail: corestat-library@uclouvain.be
http://www.uclouvain.be/en-44508.html

CORE DISCUSSION PAPER
2009/6

Using underapproximations for sparse nonnegative matrix factorization

Nicolas GILLIS[1] and François GLINEUR[2]

February 2009

Abstract

Nonnegative Matrix Factorization (NMF) has gathered a lot of attention in the last decade and has been successfully applied in numerous applications. It consists in the factorization of a nonnegative matrix by the product of two low-rank nonnegative matrices:. $M \approx VW$. In this paper, we attempt to solve NMF problems in a recursive way. In order to do that, we introduce a new variant called Nonnegative Matrix Underapproximation (NMU) by adding the upper bound constraint $VW \leq M$. Besides enabling a recursive procedure for NMF, these inequalities make NMU particularly well-suited to achieve a sparse representation, improving the part-based decomposition.
Although NMU is NP-hard (which we prove using its equivalence with the maximum edge biclique problem in bipartite graphs), we present two approaches to solve it: a method based on convex reformulations and a method based on Lagrangian relaxation. Finally, we provide some encouraging numerical results for image processing applications.

Keywords: nonnegative matrix factorization, underapproximation, maximum edge biclique problem, sparsity, image processing.

[1] Université catholique de Louvain, CORE, B-1348 Louvain-la-Neuve, Belgium.
E-mail: nicolas.gillis@uclouvain.be.
[2] Université catholique de Louvain, CORE, B-1348 Louvain-la-Neuve, Belgium.
E-mail: francois.glineur@uclouvain.be. This author is also member of ECORE, the newly created association between CORE and ECARES.

The first author is a research fellow of the Fonds de la Recherche Scientifique (F.R.S.-FNRS).

This paper presents research results of the Belgian Program on Interuniversity Poles of Attraction initiated by the Belgian State, Prime Minister's Office, Science Policy Programming. The scientific responsibility is assumed by the authors.

1 Introduction

Low-rank approximation is a well-known data analysis technique and has been widely used for dimensionality reduction, noise filtering, classification, interpretation, ... More precisely, given a data matrix M, we would like to express it as the product of two low-rank matrices V and W

$$M \approx VW. \tag{1.1}$$

For example, if the noise on the data is assumed to be Gaussian, the sum of squares of the entries of the error should be minimized :

$$\min_{V,W} \quad ||M - VW||_F,$$

where $||.||_F$ is the Frobenius norm i.e. $||A||_F^2 = \sum_{i,j} A_{ij}^2$. This problem has been extensively studied and can be solved efficiently using the *Singular Value Decomposition* (SVD) [24].

It can be interpreted as a linear model: assuming each column of M represents an element of the data set and rewriting (1.1) as[1]

$$M_{:j} \approx \sum_k V_{:k} W_{kj}, \quad \forall j \ , \tag{1.2}$$

we observe that each element of the initial data set ($M_{:j}$, a column of M) is decomposed into a linear combination (with weights W_{kj}) of basis elements ($V_{:k}$, the columns of V).

In some cases, the matrix M might be nonnegative; this is in general due to particular physical interpretations, e.g. elements are images described by pixel intensities or texts represented by vectors of word counts. Therefore, imposing V to be nonnegative allows to interpret the basis elements in the same way as the columns of M. Moreover, the nonnegativity of the weight matrix W corresponds to an essentially additive reconstruction. This leads to a *part-based representation*: basis elements will represent common parts of the columns of M.

The low-rank approximation problem with nonnegativity constraints is commonly called *Nonnegative Matrix Factorization* (NMF). This approximation problem was introduced in 1994 by Paatero and Tapper [36] but has only been extensively studied after the publication of a paper by Lee and Seung [31] in 1999. It is now well established that NMF is useful in the framework of compression and interpretation of nonnegative data. It is used for example in image processing, text mining, spectral unmixing, air emission control, computational biology and for other applications (see e.g. [4, 17, 15]). The most famous example of NMF application deals with facial images. The matrix M is built such that each column represents a face using pixel intensities. As expected, the additive nature of NMF extracts bases of facial features, such as eyes, noses and lips, see Figure 1.

Another important property that the factors should satisfy is *sparsity*: it improves the compression and allows a better part-based representation of the data which makes interpretation easier. The usual way to assess sparsity is to count the number of zero values. An alternative is proposed by Hoyer in [29] : for a nonzero n dimensional vector x, it is defined as

$$\text{sparsity}(x) = \frac{\sqrt{n} - ||x||_1/||x||_2}{\sqrt{n} - 1} \quad \in \quad [0, 1]. \tag{1.3}$$

Hence, a vector with one nonzero entry is perfectly sparse

$$\text{sparsity}([0 \dots 0 \ k \ 0 \dots 0]) = 1, \quad \forall k \neq 0,$$

while a vector with entries equal to each other is completely dense

$$\text{sparsity}([k \dots k]) = 0, \quad \forall k \neq 0.$$

There are currently two (main) techniques to enforce sparsity on NMF solutions :

[1]The following notation will be used throughout the paper: M_{ik} is the (i,k) entry of the matrix M and $M_{:k}$ (resp. $M_{k:}$) is the k^{th} column (resp. row) of M.

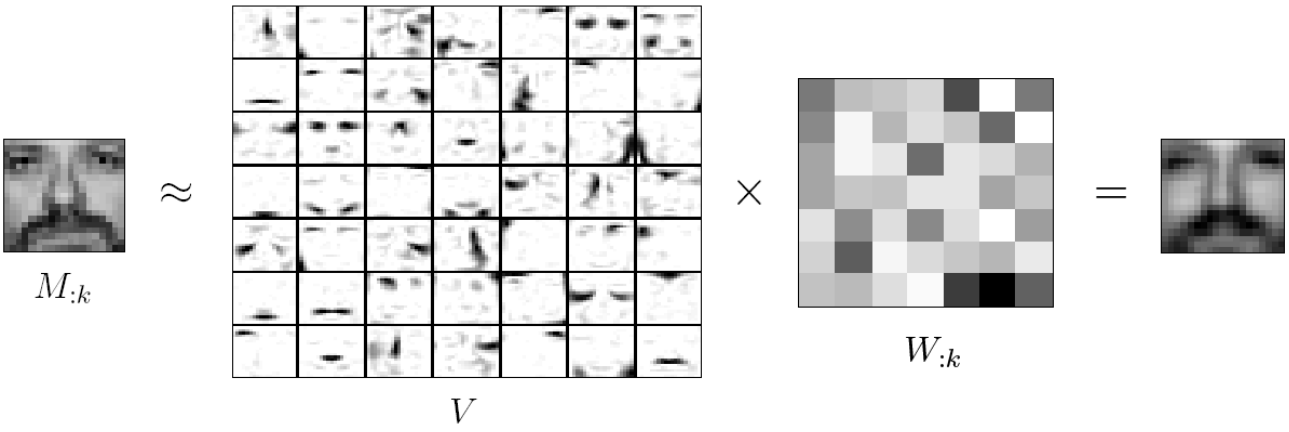

Figure 1: Approximation of the CBCL Face Database #1, MIT Center For Biological and Computation Learning. Available at *http://cbcl.mit.edu/cbcl/software-datasets/FaceData2.html*. It consists of 2429 gray-level images of faces (columns of M) with 19×19 pixels (rows of M) for which we set $r = 49$.

1. Project the (potentially dense) solution to the nearest point lying in a space of specified sparsity [29];
2. Add penalty terms in V and W in the cost function. For example, it is well known that adding l_1-norm penalty terms induces sparser factors (see e.g. [30]).

The first possibility is more difficult to implement and the sparsity of the factors has to be chosen a priori. The second is simpler but an appropriate penalizing term has to be chosen and its parameters have to be tuned. In this paper, we will introduce a new way to enforce sparsity in such a way that it is naturally achieved.

2 Nonnegative Matrix Factorization (NMF)

Let formulate our problem more formally: given $M \in \mathbb{R}_+^{m \times n}$ and $1 \leq r < \min(m, n)$, the *NMF optimization problem* is defined as

$$\min_{V \in \mathbb{R}^{m \times r}, W \in \mathbb{R}^{r \times n}} ||M - VW||_F^2 \quad V \geq \mathbf{0}, W \geq \mathbf{0}. \tag{NMF}$$

$\mathbb{R}^{m \times n}$ denotes the set of real matrices of dimension $m \times n$; $\mathbb{R}_+^{m \times n}$ the set $\mathbb{R}^{m \times n}$ of matrices with every entry nonnegative, $\mathbb{R}_{++}^{m \times n}$ the set $\mathbb{R}^{m \times n}$ of matrices with every entry positive and $\mathbf{0}$ the zero matrix of appropriate dimensions. Although (NMF) has been shown to be NP-hard [39], there exists a wide range of algorithms to find approximate solutions to this problem. In general, some initial guess matrices (V, W) are improved iteratively using nonlinear optimization schemes, e.g. projected gradient methods [35], Newton-like methods [16, 10], (block) coordinate descent (which corresponds to alternating nonnegative least squares - NNLS) [9, 11, 30], multiplicative updates [32], etc. (see also [12, 27] and references therein).
Some effort has recently been made to develop greedy techniques based on rank-one updating (see e.g. [6, 21]). This paper will focus on this second type of methods. The technique we propose is based on an underapproximation idea and will be introduced and analyzed in Section 3. But first we describe briefly two standard algorithms for NMF which will be useful throughout the paper.

2.1 Multiplicative Updates (MU)

In [32], Lee and Seung propose multiplicative update rules to minimize the Frobenius norm between M and VW whose underlying idea is a variable metric gradient descent method, for which they are able to guarantee the monotonicity of the objective function without having to tune the step size. This method is described by Algorithm 1.

Algorithm 1 Multiplicative Updates

Require: $M \in \mathbb{R}_+^{m \times n}$, $V \in \mathbb{R}_+^{m \times r}$ and $W \in \mathbb{R}_+^{r \times n}$.
Ensure: $(V, W) \geq \mathbf{0}$ s.t. $VW \approx M$.

1: **for** $i = 1, 2, \ldots$ **do**
2: $\quad V \leftarrow V \circ \frac{[MW^T]}{[VWW^T]}$;
3: $\quad W \leftarrow W \circ \frac{[V^T M]}{[V^T VW]}$.
4: **end for**

$\circ$ and $\frac{[.]}{[.]}$ are the Hadamard (component-wise) multiplication and division.

Theorem 1 ([32])**.** *The Frobenius norm $||M - VW||_F$ is nonincreasing under the updates of Algorithm 1.*

Although this algorithm does not guarantee convergence to a stationary point, it is possible to modify it slightly in order to get this property [34, 22]. Moreover, if one of the two matrices V or W is fixed, the problem becomes convex and the multiplicative updates have then been proved to converge to a global optimum [38]. Finally, for $r = 1$, those updates are equivalent to the standard power method [21].

2.2 Hierarchical Alternating Least Squares (HALS)

Another possibility is to minimize alternatively the cost function *separately for each entry* of V and of W while all the others are fixed. This is the simple method of *coordinate descent* (also called alternating variables). Specifically, for the Frobenius norm, if all the variables are fixed except for V_{ik}, the problem becomes

$$V_{ik}^* = \text{argmin}_{V_{ik} \geq 0} \ ||M - VW||_F^2 \tag{2.1}$$

which requires minimization of a quadratic function over the nonnegative real line. The optimal solution of (2.1) is

$$V_{ik}^* = \max\left(0, \frac{(MW^T)_{ik} - \sum_{l=1, l \neq k}^{r} V_{il}(WW^T)_{lk}}{(WW^T)_{kk}}\right).$$

We observe that the optimal value of each entry of V does not depend on the other entries of the same column. Therefore, the optimal value for each column of V can be computed using an optimal closed-form solution (cf. Algorithm 2). By symmetry, the same property holds for each row of W. Algorithm 2 alternatively updates the columns of V and the rows of W. It is called *Hierarchical Alternating Least Squares* (HALS) [11][2] and has been shown to work remarkably well in practice: it outperforms, in most cases, the others algorithms for NMF [12, 27, 23]. In particular, it has the same computational complexity per iteration as the MU while it can be proved to be locally more efficient [22] (see Figure 2 for an example) and be shown to converge, under some mild assumptions, to a stationary point [26].

[2] In [27], it is called Rank-one Residue Iteration (RRI) method and in [23] Alternating NMF (ANMF).

Algorithm 2 Hierarchical Alternating Least Squares

Require: $M \in \mathbb{R}_+^{m\times n}$, $r > 0$, $V \in \mathbb{R}_+^{m\times r}$, $W \in \mathbb{R}_+^{r\times n}$.
Ensure: (V, W) s.t. $VW \approx M$.

for $i = 1, 2, \ldots$ **do**
  Compute $A = MW^T$ and $B = WW^T$.
  for $k = 1 : r$ **do**
    $V_{:k} \leftarrow \max\left(\mathbf{0}, \frac{A_{:k} - \sum_{l=1, l\neq k}^{r} V_{:l} B_{lk}}{B_{kk}}\right)$
  end for
  Compute $C = V^T M$ and $D = V^T V$.
  for $k = 1 : r$ **do**
    $W_{k:} \leftarrow \max\left(\mathbf{0}, \frac{C_{k:} - \sum_{l=1, l\neq k}^{r} D_{kl} W_{l:}}{D_{kk}}\right)$
  end for
end for

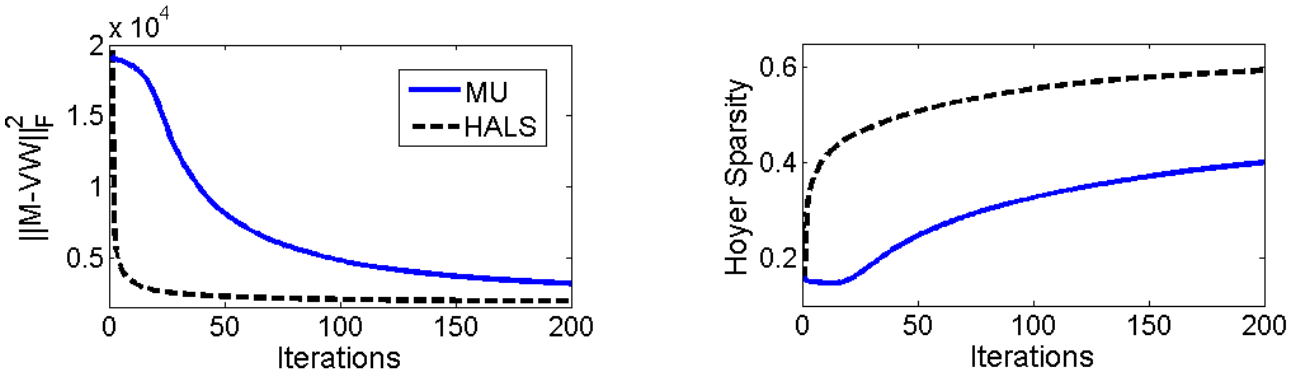

Figure 2: Comparison of the MU and the HALS applied to the CBCL database with $r = 49$ and the same scaled random initialization has been used for both algorithms.

Remark 1. *The updates of the columns of V and the rows of W can be performed in any cyclic order. Algorithm 2 first updates the columns of V from left to right and then the rows of W from top to bottom.*

Remark 2. *Algorithm 2 is sensitive to the scaling of the initial matrices. For example, if the initial matrices V and W are chosen such that $VW \ggg M$, optimal columns of V and optimal rows of W will most likely be equal to zero. This will lead to rank deficient approximations ($V_{:k}W_{k:} = 0$ for some k) and numerical problems (for $V_{:k} = 0$, update of $W_{k:}$ is not well defined and vice versa). If the initial matrices (V, W) are scaled [22] i.e.*

$$\text{argmin}_\alpha \, ||M - \alpha VW||_F = \frac{\langle M, VW\rangle}{\langle VW, VW\rangle} = 1. \tag{2.2}$$

where $\langle A, B\rangle = \sum_{i,j} A_{ij}B_{ij} = trace(AB^T)$, this behavior is in general avoided. Therefore, from now on, we will always make sure that initial matrices are chosen scaled.

Remark 3. *As explained in Section 1, a l_1 penalty term can be added to the cost function to get sparser solutions*

$$\min_{V\in\mathbb{R}_+^{m\times r}, W\in\mathbb{R}_+^{r\times n}} \; ||M - VW||_F^2 + \lambda||V||_1$$

where $||V||_1 = \sum_{ik} |V_{ik}|$*, and Algorithm 2 can easily be adapted accordingly [26, p.76] (of course, the same is valid for* W *by symmetry).*

3 Nonnegative Matrix Underapproximation

The rank-one NMF problem is not as hard as the general problem. When using the Frobenius norm as the cost function, the first rank-one factor of the singular value decomposition is an optimal solution: indeed, the Perron-Frobenius theorem implies that the dominant left and right singular vectors of a nonnegative matrix are nonnegative, while the Eckart-Young theorem states that the outer product of these dominant singular vectors is the best possible rank-one approximation. One can therefore think of applying this result recursively: after identification of an optimal rank-one NMF solution (v, w), one could subtract the vw factor from M and apply the same technique to $M - vw$. Unfortunately, this cannot work: the difference between M and its rank-one approximation may contain negative values so that the next SVD factor no longer gives a nonnegative solution. Moreover, finding the optimal nonnegative rank-one approximation of a matrix which is not nonnegative becomes NP-hard [22].

In order to build a recursive algorithm, we therefore add the *upper bound constraint* $VW \leq M$ to (NMF) and obtain a new problem we call *Nonnegative Matrix Underapproximation* (NMU): given $M \in \mathbb{R}^{m \times n}_+$and $1 \leq r < \min(m, n)$, the NMU optimization problem is defined as

$$\begin{aligned} \min_{V \in \mathbb{R}^{m \times r}, W \in \mathbb{R}^{r \times n}} & ||M - VW||_F^2 \\ & VW \leq M \\ & V \geq \mathbf{0}, W \geq \mathbf{0}. \end{aligned} \tag{NMU}$$

Assuming we are able to solve it for $r = 1$, an underapproximation of any rank can then be built by following the recursive procedure outlined above. More precisely, if $(V_{:1}, W_{1:})$ is a rank-one underapproximation for M, i.e. $V_{:1}W_{1:} \approx M$ and $V_{:1}W_{1:} \leq M$, we have that $R_1 = M - V_{:1}W_{1:}$ is nonnegative. R_1 can then be underapproximated $V_{:2}W_{2:} \leq R_1$, leading to $R_2 = R_1 - V_{:2}W_{2:}$, and so on. After r steps, we get an underapproximation of rank r

$$\begin{aligned} M &\geq V_{:1}W_{1:} + V_{:2}W_{2:} + \cdots + V_{:r}W_{r:} \\ &= [V_{:1}\ V_{:2}\ \ldots\ V_{:r}][W_{1:};\ W_{2:}; \ldots; W_{r:}] \\ &= VW. \end{aligned}$$

3.1 Complexity

In this section, we prove that (NMU) is NP-hard, even in the rank-one case (unlike (NMF)). In order to do this, we first prove that the rank-one version of the problem is equivalent to the biclique problem, which is NP-hard. We then prove that (NMU) with arbitrary r is NP-hard as well using a simple construction.

A *bipartite graph* G_b is a graph whose vertices can be divided into two disjoint sets such that there is no edge between two vertices in the same set. A *biclique* K_b is a complete bipartite graph i.e. a bipartite graph where all the vertices from different sets are connected by an edge. Finally, the so-called maximum edge biclique problem (the biclique problem for short) in a bipartite graph

$$G_b = \Big(V = V_1 \cup V_2, E \subset (V_1 \times V_2)\Big)$$

is the problem of finding a biclique $K_b = (V', E')$ in G_b (i.e. $V' \subseteq V$ and $E' \subseteq E$) with a maximum number of edges $|E'|$.

Letting $M \in \{0,1\}^{m\times n}$ be the adjacency matrix of G_b with $V_1 = \{s_1, \dots s_m\}$ and $V_2 = \{t_1, \dots t_n\}$:

$$M_{ij} = 1 \Leftrightarrow (s_i, t_j) \in E,$$

and introducing indicator binary variables v_i (resp. w_j) to denote whether s_i (resp. t_j) belongs to the biclique K_b, the Maximum edge Biclique Problem (MBP) in a bipartite graph can be formulated as follows

$$\begin{aligned} \min_{v,w} \quad & \sum_{i,j} (M_{ij} - v_i w_j)^2 \\ & v_i w_j \leq M_{ij},\ \forall i, j \\ & v \in \{0,1\}^m, w \in \{0,1\}^n. \end{aligned} \tag{MBP}$$

One can check that this objective is equivalent to $\max_{v,w} \sum_{i,j} v_i w_j$. In fact, $M_{ij} - v_i w_j = (M_{ij} - v_i w_j)^2$ since M, v and w are binary and $M_{ij} \geq v_i w_j$.
The decision problem: *Given K, does G_b contain a biclique with at least K edges?* has been shown to be NP-complete [37]. Therefore, the corresponding optimization problem (MBP) is at least NP-hard.

For $r = 1$, (NMU) can be written as

$$\begin{aligned} \min_{v\in\mathbb{R}^m, w\in\mathbb{R}^n} \quad & \sum_{i,j} (M_{ij} - v_i w_j)^2 \\ & v_i w_j \leq M_{ij},\ \forall i, j \\ & v \geq \mathbf{0}, w \geq \mathbf{0}, \end{aligned} \tag{NMU1}$$

which is very close to (MBP): the difference is that vectors v and w are required to be *binary* for (MBP) and *nonnegative* for (NMU1). The next lemma proves that the two problems are actually equivalent.

Lemma 1. *For $M \in \{0,1\}^{m\times n}$, every optimal solution (v, w) of* (NMU1) *is binary i.e. $vw \in \{0,1\}^{m\times n}$. It can then be trivially transformed into a binary optimal solution $(v', w') \in \{0,1\}^m \times \{0,1\}^n$ of* (MBP).

Proof. For $M = 0$, this is trivial. Otherwise, suppose (v, w) is an optimal solution of (NMU1). Let define (v', w') as

$$v'_i = \begin{cases} 1 & \text{if } v_i \neq 0 \\ 0 & \text{otherwise} \end{cases} \qquad w'_j = \begin{cases} 1 & \text{if } w_j \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

and analyze the different possibilities: as $v_i w_j \leq M_{ij}$, we have either

- $M_{ij} = 1$ and $0 < v_i w_j \leq 1 \Rightarrow v'_i w'_j = 1$;
- $M_{ij} = 1$ and $v_i w_j = 0 \Rightarrow v'_i w'_j = 0$;
- $M_{ij} = 0 \Rightarrow v_i w_j = 0 \Rightarrow v'_i w'_j = 0$.

Therefore, $v_i w_j \leq v'_i w'_j \leq M_{ij}$ which implies

$$||M - v'w'||_F \leq ||M - vw||_F.$$

By optimality of (v, w), we must have $vw = v'w' \in \{0,1\}^{m\times n}$. Therefore, $(v', w') = (v/\max(v), w/\max(w))$ is an optimal binary solution of (NMU1) which is then also an optimal solution of (MBP) (note that we must have $\max(v) = \max(w)^{-1}$). □

Corollary 1. (NMU1) *is NP-hard.*

We now generalize Corollary 1 to the more general case of (NMU) with $r > 1$.

Theorem 2. (NMU) *is NP-hard.*

Proof. Let $M \in \{0,1\}^{m\times n}$ be the adjacency matrix of a bipartite graph G_b. We define the matrix A as

$$A = \operatorname{diag}(M, r) = \begin{pmatrix} M & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & M & & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \dots & & M \end{pmatrix}$$

which is the adjacency matrix of another bipartite graph G_b^r which is nothing but the graph G_b repeated r times. Let (V, W) be an optimal solution of (NMU). Since $VW = \sum_{k=1}^{r} V_{:k}W_{k:}$, we have $VW \leq A \Rightarrow V_{:k}W_{k:} \leq A$. Therefore $(V_{:k}, W_{k:})$ is a feasible solution of (NMU1) for the matrix A i.e. for the graph G_b^r.

Hence, each $(V_{:k}, W_{k:})$ corresponds to a biclique $B_G^k = (V_1^k \cup V_2^k, E^k)$ of G_b^r with

$$V_{ik} \neq 0 \Leftrightarrow s_i \in V_1^k \quad \text{and} \quad W_{kj} \neq 0 \Leftrightarrow t_j \in V_2^k.$$

By optimality and since there are at least r independent maximum biclique in G_b^r, each $(V_{:k}, W_{k:})$ must coincide with a maximum biclique of G_b^r which corresponds to a maximum biclique of G_b. In fact, G_b^r is the graph G_b repeated r times and clearly a biclique cannot span two disjoint subgraphs of G_b^r. Therefore, (NMU) is NP-hard since any instance of (MBP) can be polynomially reduced to an instance of (NMU). □

In the next two sections, we present two different approaches to solve (NMU). The first one was originally introduced in the master's thesis of the first author [21].

3.2 Convex Formulations

In [21], the rank-one NMU problem is analyzed in order to build a recursive algorithm. In particular, the aim is to find alternative cost functions such that the corresponding problem is convex, hence efficiently solvable.

A first possibility is to use the following formulation

$$\min_{v,w\geq \mathbf{0}} \Big\{ \max_{i,j} |M_{ij} - v_i w_j| \Big\}, \quad vw \leq M \tag{3.1}$$

which can be solved with a sequence of linear systems of inequalities. This is done in two stages:

1. Reduce the problem to the case $M > \mathbf{0}$. Since

   $$M_{kl} = 0 \Rightarrow v_k = 0 \text{ or } w_l = 0,$$

   $M_{kl} = 0$ implies that either the k^{th} row or the l^{th} column of M is approximated by $\mathbf{0}$. Hence

   $$\min\Big(\max_j(M_{kj}), \max_i(M_{il}) \Big) \leq \max_{i,j}(M_{ij} - v_i w_j),$$

   for any feasible solution (v, w) of (3.1). Therefore, if the maximum entry of the k^{th} row (resp. l^{th} column) of M is greater that the maximum entry of the l^{th} column (resp. k^{th} row) of M, one can then set $w_l = 0$ (resp. $v_k = 0$). Indeed, if any optimal solution features one v_k set to zero (resp. w_l), w_l (resp. v_k) could also be set to zero without deteriorating the solution.
   We then successively remove columns or rows corresponding to the zero entries of M and we end up with a reduced problem with strictly positive matrix M.

2. For $M > \mathbf{0}$, one can check that there always exists a positive optimal solution $(v, w) > \mathbf{0}$ (if an entry of v or of w is equal to zero in an optimal solution, it can be replaced by a sufficiently small positive constant such that vw remains smaller than M). We then introduce the variables $w'_j = w_j^{-1}\ \forall j$ and w.l.o.g. we impose $v_i \geq 1\ \forall i$; if the following linear system of inequalities

$$\begin{aligned} v_i &\leq w'_j M_{ij}, \quad \forall i,j \\ w'_j M_{ij} - v_i &\leq B w'_j, \quad \forall i,j \\ v_i &\geq 1, \quad \forall i \end{aligned}$$

is solvable, it means that the optimal value of (3.1) is lower than B (and a solution of this system is a feasible solution of (3.1) with optimal value lower than B), greater than B otherwise. Hence (3.1) can be solved using a sequence of such linear systems of inequalities, e.g. with a bisection algorithm for values of B starting in $[0, \max_{ij}(M_{ij})]$.

Unfortunately, this formulation is not really interesting for practical applications since it focuses only on approximating the large entries of M.

For the specific case of *positive* matrices ($M > \mathbf{0}$), (NMU) has also been shown to be NP-hard for any factorization rank [21]. However, it is possible to design other cost functions for which the problem is not as hard. Consider for example a cost function based on the ratio between M and its approximation

$$\min_{v,w>0} \sum_{i,j} \left| \log\left(\frac{M_{ij}}{v_i w_j}\right) \right|.$$

By the logarithmic change of variables: $\forall i,j$

$$c_{ij} = \log(M_{ij}),\ x_i = \log(v_i) \ \text{ and } \ y_j = \log(w_j)$$

one can compute the optimal rank-one *positive* matrix factorization with the above cost function with the following linear program :

$$\min_{x,y} \sum_{i,j} |c_{ij} - x_i - y_j|.$$

Observing that the additional underapproximation constraints can be written $x_i + y_j \leq c_{ij}\ \forall i,j$, we get a linear program

$$\begin{aligned} \max_{x,y}\ & n\sum_i x_i + m\sum_j y_j \\ & x_i + y_j \leq c_{ij}, \quad \forall i,j. \end{aligned} \tag{LP}$$

Note that this is the dual of a flow problem, namely, the Hitchcock problem [25] so that it can be solved efficiently with dedicated algorithms [20].
Another possible formulation for the rank-one positive matrix factorization is

$$\min_{v,w>0} \sum_{i,j} \max\left(\frac{M_{ij}}{v_i w_j}, \frac{v_i w_j}{M_{ij}}\right)$$

which can be cast as a Geometric Program [19]. Recall that a geometric program has the following form

$$\begin{aligned} \min_x\ & f_0(x) \\ & f_i(x) \leq 1 \\ & x > \mathbf{0} \end{aligned}$$

where f_i are *posynomials* i.e.

$$f_i(x) = \sum_k c_{ik} x_1^{a_{ik1}} x_2^{a_{ik2}} ... x_n^{a_{ikn}} \quad \text{with} \quad c_{ik} \geq 0, \, \forall i, k.$$

and that it can be *convexified* through a logarithmic change of variables (see e.g. [8]).
Hence, the rank-one underapproximation problem can be formulated as the following instance

$$\begin{aligned} \min_{v,w>\mathbf{0}} \quad & \sum_{i,j} \frac{M_{ij}}{v_i w_j} \\ & \frac{v_i w_j}{M_{ij}} \leq 1. \end{aligned} \tag{GP}$$

Unfortunately, (LP) and (GP) can only be applied to positive matrices. Some additional work has to be done if one wants to deal with nonnegative matrices. For example, one could think of extracting, from the nonnegative matrix, a rectangular positive submatrix (cf. Example 1) on which those methods can be applied. Of course, we would like to extract the larger submatrix possible: this amounts exactly to finding the maximum weighted biclique problem, which is NP-hard as well [14]. However, heuristics algorithms exists for this problem and good approximate solutions can often be computed efficiently (see e.g. [28, 21, 22, 2]). Combining these ideas, the following recursive NMU algorithm can be designed (Algorithm 3).

Algorithm 3 Recursive NMU [21]

Require: $R_1 = M$, $r > 0$.
Ensure: (V, W) s.t. $VW \leq M$ with $VW \approx M$.

1: **for** $k = 1$: r **do**
2: Extract a positive submatrix $\tilde{R}_k$ from R_k;
3: Perform a rank-one underapproximation $(\tilde{V}_{:k}, \tilde{W}_{k:})$ of $\tilde{R}_k$ using (LP) or (GP);
4: Extend $(\tilde{V}_{:k}, \tilde{W}_{k:})$ to $(V_{:k}, W_{k:})$ by adding zeros at the entries corresponding to the ones of R_k not in $\tilde{R}_k$;
5: Compute $R_{k+1} = R_k - V_{:k} W_{k:} \geq \mathbf{0}$.
6: **end for**

Example 1. *One iteration of Algorithm 3 with* (LP) *:*

$$\begin{aligned} M &= \begin{pmatrix} 1 & 3 & 0 & 7 & 0 \\ 0 & 8 & 2 & 0 & 0 \\ \boxed{9} & 6 & \boxed{4} & 0 & \boxed{1} \\ \boxed{8} & 0 & \boxed{5} & 0 & \boxed{3} \end{pmatrix} \\ &\leq \begin{pmatrix} 0 \\ 0 \\ \boxed{0.8} \\ \boxed{1} \end{pmatrix} \begin{pmatrix} \boxed{8} & 0 & \boxed{5} & 0 & \boxed{1.25} \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \boxed{6.4} & 0 & \boxed{4} & 0 & \boxed{1} \\ \boxed{8} & 0 & \boxed{5} & 0 & \boxed{1.25} \end{pmatrix}. \end{aligned}$$

In [21], Algorithm 3 is used as an initialization technique for standard NMF algorithms to accelerate convergence and, in general, is observed to improve the final solution compared to random initializations. The same kind of behavior was already observed for other judicious initializations [1, 7, 13].

Remark 4. *The problem of rank-one underapproximation has been first introduced by Levin in [33] in the case of positive stochastic matrices, where he defines a specific cost function and uses a logarithm change of variables to analyze an iterative method based on the KKT conditions.*

An attempt is also made in [21] to compute higher rank factorizations using convex formulations. For example, using geometric programming, one could try to maximize (V, W) while imposing the underapproximation constraint :

$$\begin{aligned} \min_{V>\mathbf{0}, W>\mathbf{0}} \ & f(V,W) \\ & (VW)_{ij} \leq M_{ij}, \, \forall i,j \end{aligned} \tag{3.2}$$

where $f(V, W)$ is any posynomial nonincreasing in V and W e.g. $\sum_{ik} V_{ik}^{-1} + \sum_{kj} W_{kj}^{-1}$.
A first disadvantage of this formulation is that it can only deal with positive matrices and is not able to set variables to zero. Moreover, we observed that if $f(V, W)$ is symmetric with respect to each rank-one factor of VW i.e. that for any permutation σ of $[1, 2, \dots r]$

$$\tilde{V}_{:k}\tilde{W}_{k:} = V_{:\sigma(k)}W_{\sigma(k):} \, \forall k \Rightarrow f(\tilde{V}, \tilde{W}) = f(V, W), \tag{3.3}$$

then the optimal solution obtained (V^*, W^*) had the columns of V^* and the rows of W^* equal to each other.

Theorem 3. *For any formulation* (3.2) *with a symmetric* $f(V, W)$ *(cf. Equation* (3.3)*), there exists an optimal solution* (V^*, W^*) *with the columns of* V^* *and the rows of* W^* *equal to each other.*

Proof. Consider the case $r = 2$, the proof for $r > 2$ is similar. Let

$$(V, W) = ([V_{:1}\, V_{:2}], [W_{1:}; W_{2:}])$$

be an optimal solution of (3.2) with $V_{:1} \neq V_{:2}$ or $W_{1:} \neq W_{2:}$. In the convex reformulation of the Geometric Program (3.2), the change of variables is given by

$$X_{:1} = \log(V_{:1}),\ X_{:2} = \log(V_{:2}),$$

$$Y_{1:} = \log(W_{1:}) \ \text{ and } \ Y_{2:} = \log(W_{2:})$$

where the logarithm is taken componentwise. The symmetry assumptions implies that the permutation of the columns of V and the corresponding rows of W generates another optimal solution. Therefore, in the convex reformulation, the mean (a convex combination) of these permuted solutions

$$\left(\left[\frac{X_{:1} + X_{:2}}{2}, \frac{X_{:1} + X_{:2}}{2} \right], \left[\frac{Y_{1:} + Y_{2:}}{2}, \frac{Y_{1:} + Y_{2:}}{2} \right] \right)$$

is also an optimal solution. Hence

$$V_{:1}^* = V_{:2}^* = \exp\left(\frac{\log V_{:1} + \log V_{:2}}{2} \right) = \sqrt{V_{:1} \circ V_{:2}},$$

$$W_{1:}^* = W_{2:}^* = \exp\left(\frac{\log W_{1:} + \log W_{2:}}{2} \right) = \sqrt{W_{1:} \circ W_{2:}}$$

with component-wise square root, must be an optimal solution of (3.2). □

Moreover, a similar reasoning shows that any other reasonable convex reformulation with a symmetric objective function, approximate or exact, will suffer from the same drawback and lead to rank-one solutions.

Nevertheless, it is possible to design asymmetric cost functions. For example, we tried to give different weights to each rank-one factor in the cost function (e.g. $f(V,W) = \sum_{ik} V_{ik}^{-k} + \sum_{kj} W_{kj}^{-k}$). Unfortunately, we observe experimentally that the optimal solutions are such that the columns of V and the rows of W are multiple of each other although we have no theoretical proof for this fact. We also tried to introduce random weights but it does not give satisfactory results in practice.

To conclude, it seems hopeless to extend those convex formulations for higher ranks. We suspect that it would be quite difficult, if not impossible, to find an appropriate convex formulation for NMU (and a fortiori for NMF) for $r > 1$ mostly because of the symmetry of the problem.

3.3 Lagrangian Duality

We present here a different approach to solve NMU using Lagrangian relaxation and show some results for image processing applications.

Drop the $m \times n$ underapproximation constraints $(VW)_{ij} \leq M_{ij}$ of (NMU) and add them into the objective function with the corresponding Lagrange multipliers (dual variables) $\Lambda_{ij} \geq 0$, to obtain the Lagrangian relaxation of (NMU)

$$\min_{V,W \geq \mathbf{0}} L(V,W,\Lambda) \tag{3.4}$$

with

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

w.l.o.g. a factor of $\frac{1}{2}$ was introduced to make presentation nicer. We would like to find $\Lambda_{ij} \geq 0$ s.t. $(VW)_{ij} \leq M_{ij}$ is verified for optimal solutions of (3.4) i.e. to solve the standard Lagrangian dual problem of (NMU) expressed as

$$\max_{\Lambda \geq \mathbf{0}} \quad \min_{V,W \geq \mathbf{0}} L(V,W,\Lambda). \tag{3.5}$$

This is a nondifferentiable optimization problem with the nice property that $\min_{V,W \geq \mathbf{0}} L(V,W,\Lambda)$ is a concave function and its maximization (over a convex set) is then a convex problem (see [3] and references therein). We then propose to use the following two stages to find approximate solution of (3.5):

1. Given Λ, choose (V,W) to minimize $L(V,W,\Lambda)$ i.e. solve (3.4); this is discussed in Section 3.3.1;

2. Given (V,W), update the Lagrangian variables Λ; this is described in Section 3.3.2.

3.3.1 Lagrangian Relaxation

We would like to solve (3.4) for fixed value of Λ. First observe that

$$\begin{aligned}
L(V,W,\Lambda) &= \sum_{i,j} \frac{1}{2}(M - VW)_{ij}^2 + \sum_{i,j} \Lambda_{ij}(VW - M)_{ij} \\
&= \frac{1}{2}\sum_{i,j} M_{ij}^2 - \sum_{i,j} M_{ij}(VW)_{ij} + \frac{1}{2}\sum_{i,j}(VW)_{ij}^2 \\
&\quad + \sum_{i,j} \Lambda_{ij}(VW)_{ij} - \sum_{i,j} \Lambda_{ij} M_{ij} \\
&= \frac{1}{2}||(M - \Lambda) - VW||_F^2 - \frac{1}{2}||\Lambda||_F^2.
\end{aligned}$$

Hence, minimizing $L(V, W, \Lambda)$ is the same as minimizing $||(M - \Lambda) - VW||_F^2$. Since $R = (M - \Lambda)$ is not necessarily nonnegative, this is a more general problem than NMF. It is actually studied in detail[3] in [22] where it is called Nonnegative Factorization (NF) and is formulated as:

$$\min_{V \in \mathbb{R}^{m \times r}, W \in \mathbb{R}^{r \times n}} ||R - VW||_F^2$$
$$V, W \geq \mathbf{0} \tag{NF}$$

with $R \in \mathbb{R}^{m \times n}$ and $1 \leq r < \min(m, n)$. (NF) is NP-hard for any factorization rank (including $r = 1$) [22]. However, algorithms for NMF can be extended to NF: for the multiplicative updates of Lee and Seung, we have the following theorem

Theorem 4 ([22]). *For any $M, \Lambda, V, W \geq \mathbf{0}$, the cost function $||M - \Lambda - VW||_F$ is nonincreasing under :*

$$V \leftarrow V \circ \frac{[MW^T]}{[VWW^T + \Lambda W^T]}, W \leftarrow W \circ \frac{[V^T M]}{[V^T VW + V^T \Lambda]}. \tag{3.6}$$

We also note that Algorithm 2 (HALS) is already applicable to any real matrix i.e. not necessarily nonnegative.

3.3.2 Update of Λ

The second decision is how to update Λ in order to find an appropriate solution. Any optimal solution (Λ^*, V^*, W^*) of (3.5) must satisfy the complementarity slackness conditions

$$\Lambda_{ij}^*(M - V^*W^*)_{ij} = 0, \forall i, j.$$

Then, during the optimization process, there are two main possibilities

- if $(M - VW)_{ij} > 0$, Λ_{ij} should be decreased and eventually reach zero if $(M - V^*W^*)_{ij} > 0$;
- if $(M - VW)_{ij} < 0$, Λ_{ij} should be increased to give more importance to $(M - VW)_{ij}$ in the cost function to hopefully get a feasible solution $(M - V^*W^*)_{ij} \geq 0$.

We use the following rule to update Λ

$$\Lambda \leftarrow \max(\mathbf{0}, \Lambda - \mu_k(M - VW)), \quad \mu_k \to 0$$

where μ_k is a sequence of step lengths; Λ can for example be initialized to zero. This update is based on the idea of subgradient methods; in fact, one can easily check that $(VW - M)$ is a subgradient of

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

with respect to Λ i.e. if $(\bar{V}, \bar{W}) = \text{argmin}_{V,W \geq \mathbf{0}} L(V, W, \bar{\Lambda})$, we have

$$f(\Lambda) \leq f(\bar{\Lambda}) + \left\langle \bar{V}\bar{W} - M, \Lambda - \bar{\Lambda} \right\rangle, \quad \forall \Lambda.$$

Two questions arise

- When do we update Λ? since we are using iterative algorithms to approximate the Lagrangian relaxation.
- How do we choose the sequence of step length μ_k?

[3] The same kind of generalization is also addressed in [18].

In general, in subgradient methods, the Lagrangian relaxation (cf. Equation (3.4)) can be evaluated exactly and, choosing an appropriate diminishing step size μ_k, e.g.

$$\mu_k \to 0 \quad \text{such that} \quad \sum_{k=0}^{+\infty} \mu_k = +\infty \ \text{ and } \ \sum_{k=0}^{+\infty} \mu_k^2 = 0,$$

guarantees the convergence to an optimal solution, cf. [3] and references therein. For example, one can choose

$$\mu_k = \frac{\mu_0}{k}, \quad \mu_0 > 0.$$

However, in our case, it is not possible to solve (3.4) in a reasonable computational time since the problem is NP-hard. It would even be too expensive to wait for the stabilization of (V, W) (e.g. near a stationary point). We then suggest to update (V, W) only a few times between each update of Λ. Because of our approximation procedure of (3.4), Algorithm 4 is not guaranteed of any convergence

Algorithm 4 Lagrangian NMU (L-NMU)

Require: $M \in \mathbb{R}_+^{m \times n}$, $r > 0$, $V \in \mathbb{R}_+^{m \times r}$, $W \in \mathbb{R}_+^{r \times n}$, $\mu_0 > 0$, maxiter.
Ensure: (V, W) s.t. $VW \lesssim M$.

1: $\Lambda = \mathbf{0}$; $\mu = \mu_0$.
2: **for** $k = 1 :$ maxiter **do**
3: Update (V, W) using a few iterations of MU (3.6) or HALS (Algo. 2);
4: $\Lambda \leftarrow \max(\mathbf{0}, \Lambda - \mu(M - VW))$;
5: $\mu \leftarrow \frac{\mu_0}{k}$.
6: **end for**

but, as we will see, it produces satisfactory solutions in practice.

Remark 5. *In general, solutions obtained by Algorithm 4 are not feasible. However, it is not (too) difficult to extend a solution (V, W) generated by Algorithm 4 to a feasible solution of* (NMU)*. In fact,* (NMU) *is a convex problem in V when W is fixed (and vice versa):*

$$V^* = \text{argmin}_{V \geq \mathbf{0}, VW \leq M} ||M - VW||_F^2,$$

the objective function is a quadratic convex function and the constraints are linear. Therefore, one can compute a solution of this convex program for V (or W) to get a feasible solution of (NMU)*.*

On could think of applying this recursively to find a solution of (NMU)*. Unfortunately, this is quite inefficient in practice. In fact,*

1. *This is relatively computationally expensive to solve these linearly constrained quadratic programs (with $mn + mr + nr$ inequalities);*

2. *Since feasibility is enforced at each step, the algorithm will not be able to generate good solutions. For example, suppose M has one zero in each column; then for any positive matrix V, the optimal W is equal to 0 :*

$$\forall j, \exists i \ s.t. \ M_{ij} = 0 \Rightarrow \forall j, \exists i \ s.t. \ \sum_k V_{ik} W_{kj} = 0 \Rightarrow W_{kj} = 0, \forall k, j.$$

Therefore, to make this algorithm work, one would have to decide a priori which values in V and W should be equal to zero (which is the difficult part of the problem) otherwise the algorithm will most probably generate a trivial solution.

Remark 6. *The L-NMU algorithm is not particularly well suited for sparse matrices. In fact, one has to store a dense $m \times n$ matrix with the Lagrangian variables Λ. Nevertheless, Berry et al. [5] have obtained encouraging result in applying NMU to sparse problems (anomaly detection problems in text mining). Moreover, it is possible to take advantage of the sparsity and design a much cheaper method [22]. In fact, one can use the fact that the Lagrangian variables associated with a zero of M will be nondecreasing in the course of the algorithm [22]*

$$0 \leq (V^{(k)}W^{(k)})_{ij} \text{ and } (M)_{ij} = 0 \quad \Rightarrow \quad \Lambda_{ij}^{(k)} \leq \Lambda_{ij}^{(k+1)},$$

where index k denotes the solution at step k.

4 Application : Image Processing

A crucial aspect of NMF is to achieve sparse solutions [29]: it improves the compression and the part-based representation. We claimed in the introduction that our method was able to enforce sparsity naturally. In fact, the additional constraints of *NMU lead to sparser factors* (at the detriment of the cost function). In particular,

$$M_{ij} = 0 \ \Rightarrow \ (VW)_{ij} = 0 \ \Rightarrow \ V_{ik} = 0 \text{ or } W_{kj} = 0, \ \forall k.$$

Of course, Algorithm 4 does not produce feasible solutions. However, they are in general observed to be close to feasibility and the numerical results of this section will confirm this nice sparsity property of NMU.

Algorithm 4 proposed in the previous section can be used for any rank underapproximation. Therefore, there are many different possibilities to build a rank r underapproximation. In fact, the recursive approach based on rank-one NMU proposed in the introduction can be extended to higher-rank recursion (meaning compute r' factors at a time with $r' \leq r$). However, we only present here the results for two extreme possibilities: *recursive L-NMU* (rank-one recursion, $r' = 1$ and r successive Lagrangian relaxation subproblems) and *global L-NMU* (using directly the higher rank possible $r' = r$ with a single Lagrangian relaxation).

Remark 7. *Both recursive and global L-NMU have the same computational complexity per iteration as the NMF algorithms we have presented (Multiplicative Updates and HALS) i.e. $O(mnr)$ operations per iteration (for dense matrices) even though it is slightly more expensive in practice since Λ has to be stored and updated.*

4.1 Settings

For all the tests of this section, we use Algorithm 4 with the following parameters: maxiter = 250, $\mu_0 = 1$ and two iterations of MU or HALS to update (V, W) between each update of Λ. For the NMF solutions, we used 500 iterations with either the MU or HALS so that NMU and NMF have approximatively the same computational cost (cf. Remark 7). We run the different algorithms with the same 10 scaled randoms initializations and keep the best solution (with respect to the Frobenius norm of the approximation).

Of course, those parameters could be tuned to improve both efficiency and quality of solutions of Algorithm 4. However, in all the cases we have tried so far, it seems that these parameters already give nice results.

For the recursive L-NMU, we will compute the residual matrix as follows

$$R_1 = M, \ R_{k+1} = \max(\mathbf{0}, R_k - V_k W_k) \tag{4.1}$$

with

$$V_k W_k \lesssim R_k,$$

since Algorithm 4 does not necessarily generate feasible solutions (but hopefully not far away from feasibility).

We do our tests on gray-level images. We normalize the databases such that the factorized matrices M have values between 0 and 1; 0 represents white and 1 black (this make more sense when trying to decompose M as a sum of parts since the dark regions are the constitutive parts of the objects in the databases we analyze).

4.2 Swimmer Database

We first present a simple example on the Swimmer database: it consists of 256 binary images of a body with 4 limbs which can be each in 4 different positions. We performed a rank-16 factorization which allows the approximation to be exact. Therefore, the NMU constraints are not a restriction.

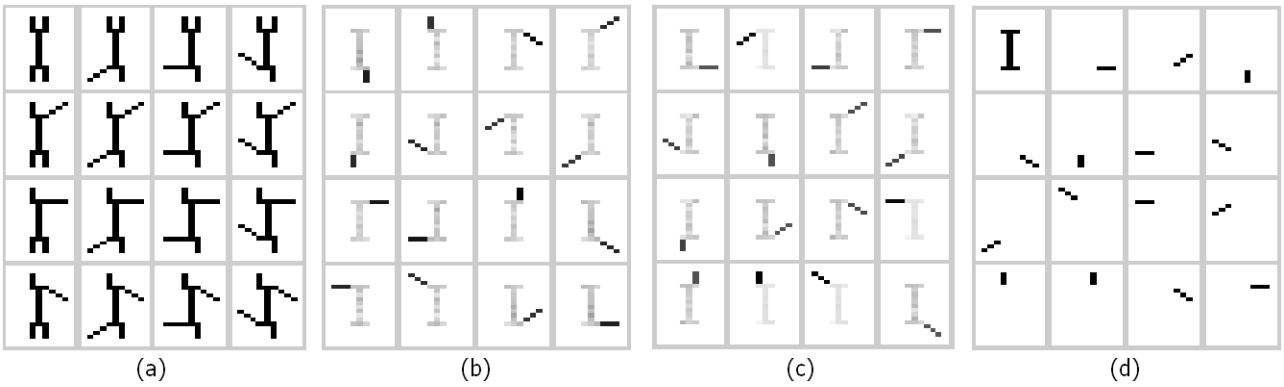

Figure 3: (a) Sample of images in the swimmer database, (b) NMF, (c) global L-NMU and (d) recursive L-NMU.

For this reason, NMF and global L-NMU generate essentially the same (optimal) solution (Figure 3). The recursive L-NMU is the only one able to extract perfectly the different parts (it needs one more rank-one factor to be exact though) in order of importance: first, the body and then the limbs.

In this case, because of the simplicity of the database, using either updates of MU or HALS for subproblem (3.4) generates basically the same optimal solutions (for most initializations).

4.3 CBCL Database

A more sophisticated test set is the CBCL faces database (cf. Figure 1). Tables 1 and 2 report the numerical results. Tables 1 gives the relative error in percent of the different approximate solutions:

$$\text{relative error } (\%) \ = 100\,\frac{||M - VW||_F}{||M||_F}.$$

Actually, it is not really fair to compare directly the error of NMF vs. NMU. In fact, the underapproximation error can be decreased by a simple rescaling i.e. multiplying VW by a scalar since $VW \lesssim M$. The optimal rescaling is given by (2.2); this is the second column of Table 1. The third column of Table 1 gives the error using the optimal weights W when V is fixed so sparsity of V is preserved. This can be computed with a Nonnegative Least Squares (NNLS) algorithm[4] [9, 30]. This confirms that, even if the Frobenius error of NMU is higher, solutions of NMU are competitive with solutions of NMF.

As could be expected, Global L-NMU achieves a smaller error than recursive L-NMU since it it not based on a greedy approach.

Clearly Algorithm 4 with HALS is much more efficient. Figure 4 shows the basis elements of L-NMU obtained with HALS. NMU achieves a better part-based representation, both the recursive and the global approaches. This is confirmed by Table 2 which gives the sparsity of the basis elements: NMU generates much sparser solutions. Note that the recursive L-NMU has the same kind of behavior as for the Swimmer database. It extracts successively parts in order of importance: the first basis

[4] An efficient implementation of NNLS [30] is available at *http://compbio.med.harvard.edu/hkim/nmf/index.html*.

Relative error (%)	Initial	Rescaled	NNLS
NMF	8.13	8.13	8.13
global L-NMU	12.49	11.23	8.71
recursive L-NMU	16.12	14.80	11.42

HALS

Relative error (%)	Initial	Rescaled	NNLS
NMF	9.37	9.37	9.28
global L-NMU	13.42	11.60	8.95
recursive L-NMU	22.15	19.17	13.24

MU

Table 1: Comparison of the error for the CBCL database.

Sparsity (V)	Hoyer	%0
NMF	0.66	54
global L-NMU	0.73	73
recursive L-NMU	0.64	51

Sparsity (V)	Hoyer	$\%10^{-3}$
NMF	0.48	25
global L-NMU	0.58	49
recursive L-NMU	0.74	80

Table 2: Comparison of the sparsity for the CBCL database (left: HALS, right: MU). Hoyer sparsity is the one from Equation (1.3), %0 means the percentage of entries equal to zero and $\%10^{-3}$ means the percentage of entries lower than 10^{-3} which is a more appropriate measure for MU since it is not able so set values to zero (only asymptotically). The columns of V were scaled ($||V_{:k}||_2 = 1, \forall k$) to make this measure well-defined.

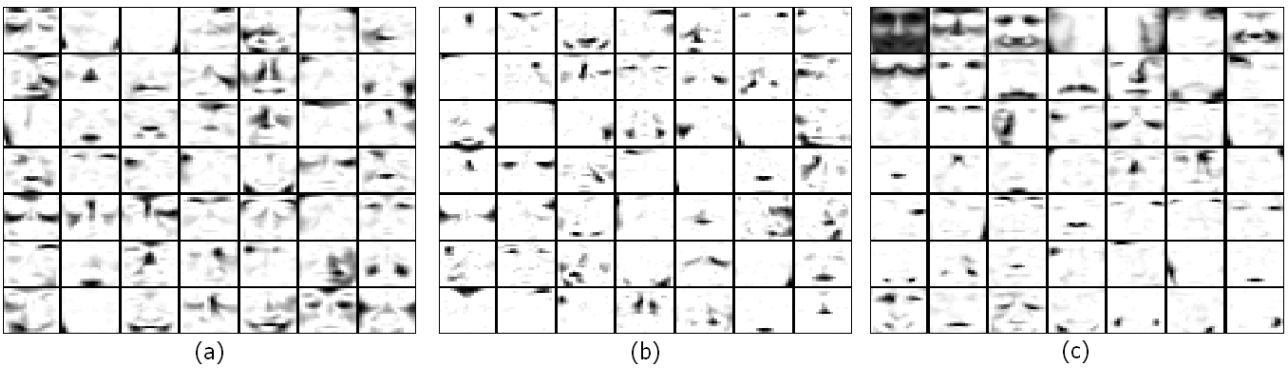

Figure 4: Basis for the CBCL database with HALS: (a) NMF, (b) global L-NMU and (c) recursive L-NMU.

element extracts what we could call a general face while the next ones extract different complementary parts.
It is worth mentioning that when using the MU, the solution obtained with the global L-NMU algorithm after a NNLS step has a smaller error than with the original MU for NMF. The reason is because the MU for NMF converge relatively slowly and cannot set value to zero while the global L-NMU algorithm is conduced faster to sparse solutions because of the underapproximation constraints.

It is not too surprising that NMU generates sparser solution than NMF because of the underapproximation constraints. The question is: *how does NMU compare with sparse NMF algorithms?* In

order to answer that, we consider the following three algorithms

1. Global L-NMU with HALS;
2. NMF with sparseness constraints[5] provided by Hoyer [29] for which we performed 1000 iterations for different sparsity levels;
3. HALS algorithm with l_1 penalty term on V (cf. [26, p.76] and Remark 3, we used the best solution obtained previously and increased $\lambda \in [0, 1.3]$ progressively).

To make a fair comparison with respect to the error, we perform one additional NNLS iteration on W for V fixed at the end of the three algorithms above. Finally, Figure 5 displays the evolution of the error versus the sparsity attained in the three cases above, and demonstrates that underlying NMU performs better than the above methods for sparse NMF on this problem.

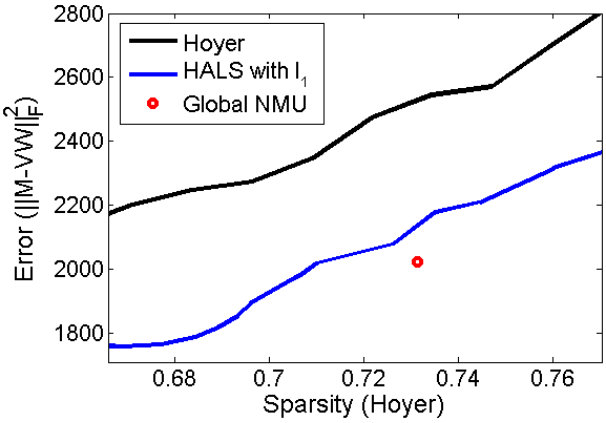

Figure 5: Sparsity versus Error for the CBCL database.

4.4 Hubble Space Telescope Spectral Images

Another database consists in 100 spectral images (128×128 pixels) of the Hubble telescope at different frequencies [40]. Figure 6 displays a sample of those images. Using $r = 8$ allows the NMF approximation to be nearly perfect. Figure 7 and Table 3 and 4 gives the visual and computational results as we did for the CBCL database.

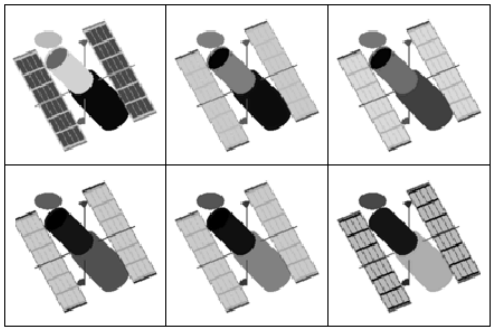

Figure 6: Sample of Hubble Space Telescope Spectral Images.

[5]Available at *http://www.cs.helsinki.fi/u/phoyer/software.html*.

Relative error (%)	Initial	Rescaled	NNLS
NMF	0.13	0.13	0.13
global L-NMU	0.46	0.39	0.21
recursive L-NMU	3.67	3.67	2.29

HALS

Relative error (%)	Initial	Rescaled	NNLS
NMF	0.79	0.79	0.62
global L-NMU	0.87	0.81	0.60
recursive L-NMU	3.80	3.22	2.06

MU

Table 3: Comparison of the error for the Hubble Telescope.

Sparsity (V)	Hoyer	%0
NMF	0.58	71
global L-NMU	0.60	72
recursive L-NMU	0.70	86

Sparsity (V)	Hoyer	$\%10^{-3}$
NMF	0.57	68
global L-NMU	0.56	68
recursive L-NMU	0.70	86

Table 4: Comparison of sparsity for the Hubble Telescope (left: HALS, right: MU).

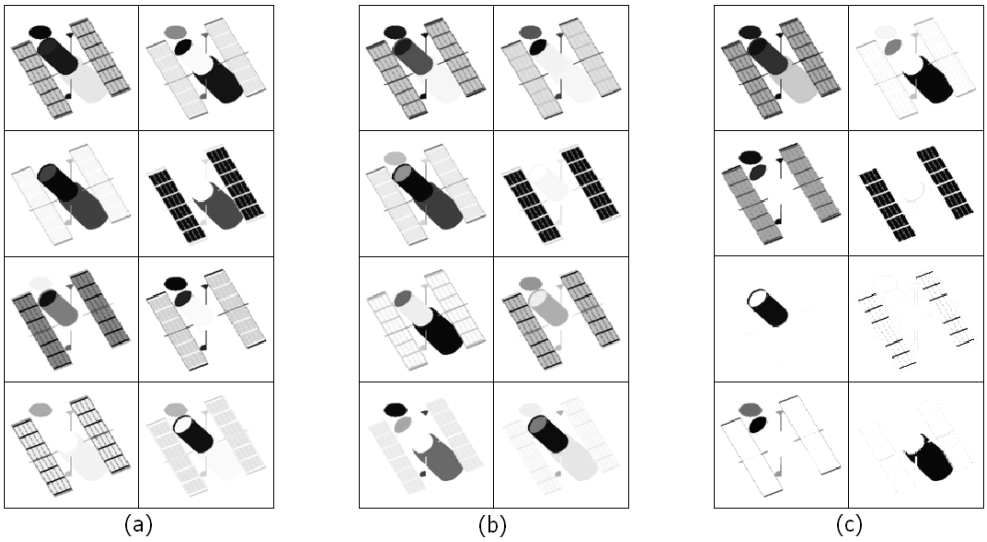

Figure 7: Basis for the Hubble telescope with HALS: (a) NMF, (b) global L-NMU and (c) recursive L-NMU.

We get essentially the same kind of behavior as before. Since NMF is nearly an exact reconstruction (Table 3), the NMU constraints are redundant. Again, recursive-NMU extracts similar parts in order of importance: first, a global vision of the telescope and then different constituent parts. It is also the sparsest solution (Table 4) achieving the best decomposition into parts (Figure 7).

Remark 8. *It does not really make sense to compare NMU with sparse NMF algorithms as we did for the CBL database (cf. Figure 5) since global L-NMU generates essentially the same solution as NMF and recursive L-NMU is a greedy technique whose error is much higher than NMF.*

4.5 Frey Database

The Frey Database is a data set of faces of Brendan Frey taken from sequential frames of a small video[6]. There are 1965 images (28×20 pixels). Table 5 and 6 and Figure 8 gives computational and visual results. We obtain similar results as before: global L-NMU manages to generate sparse

Relative error (%)	Initial	Rescaled	NNLS
NMF	9.65	9.65	9.65
global L-NMU	14.23	12.89	10.87
recursive L-NMU	18.01	16.51	15.12

HALS

Relative error (%)	Initial	Rescaled	NNLS
NMF	10.67	10.67	10.60
global L-NMU	14.95	13.01	11.17
recursive L-NMU	25.16	22.27	16.20

MU

Table 5: Comparison of the error for the Frey database.

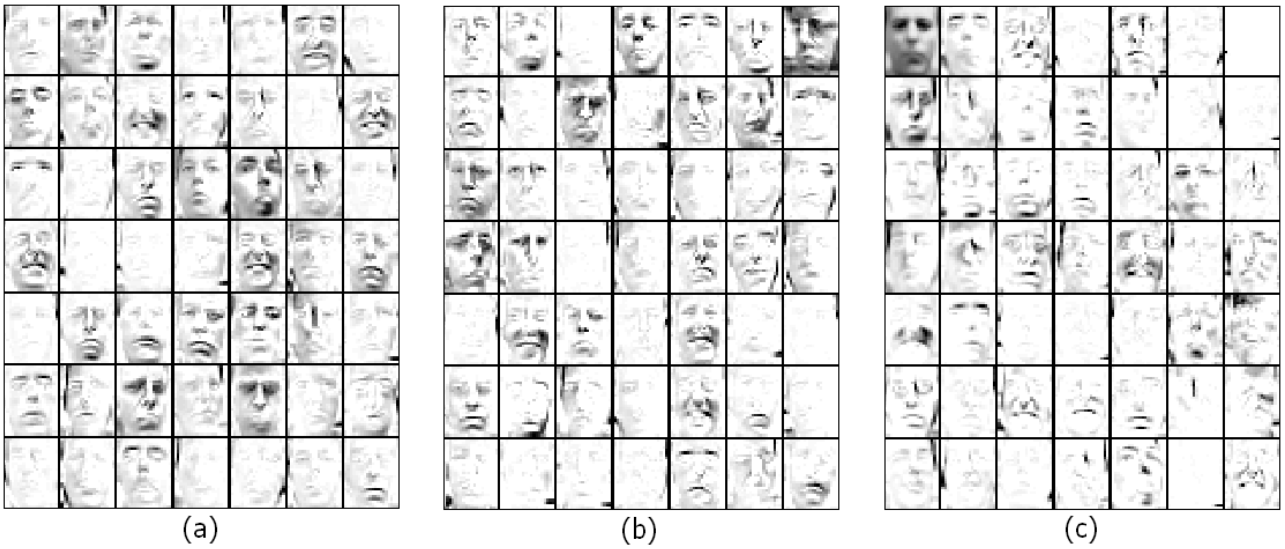

Figure 8: Basis for the Frey database with HALS: (a) NMF, (b) global L-NMU and (c) recursive L-NMU.

Sparsity (V)	Hoyer	%0
NMF	0.49	34
global L-NMU	0.56	47
recursive L-NMU	0.62	49

Sparsity (V)	Hoyer	$\%10^{-3}$
NMF	0.43	27
global L-NMU	0.53	48
recursive L-NMU	0.73	77

Table 6: Comparison of sparsity for the Frey database (left: HALS, right: MU).

[6]Available *http://www.cs.toronto.edu/ roweis/data.html*.

solution with a reconstruction error close to the one of NMF (after a NNLS step on W) while recursive L-NMU generates the sparsest solution and seems to generate a better decomposition into parts.

Figure 9 displays the error for global L-NMU compared with sparse NMF algorithms. In this case, global L-NMU is located above the two curves. However, with one additional NNLS step performed on V (which can be motivated by the fact that any NMU solution can be further improved with any NMF algorithm since it is in general far away from stationary for (NMF)), NMU performs as well as the two other algorithms.

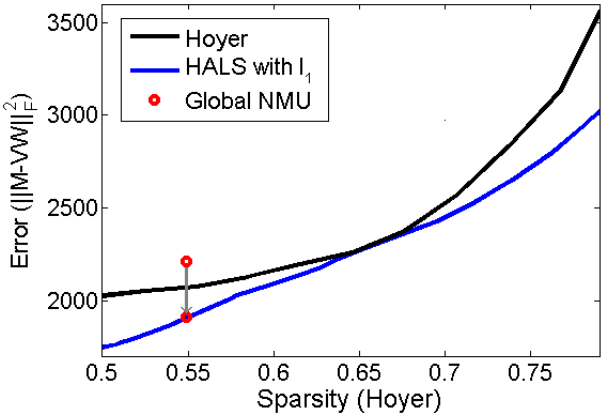

Figure 9: Sparsity versus Error for the Frey database.

5 Conclusion and Further Work

In order to solve the NMF problem in a recursive way, we introduced a new problem, namely Nonnegative Matrix Underapproximation (NMU), which was shown to be NP-hard using a reduction to the maximum-edge biclique problem. We first addressed the problem of solving NMU through a convex formulation/relaxation in the rank-one case and we explained why the inherent symmetry of the problem makes this approach nearly useless to compute factorizations of higher rank. We then proposed an algorithm based on Lagrangian relaxation to solve (NMU).

The additional constraints of NMU lead to sparser solutions and therefore an improved part-based representation and a higher compression of the columns of the input matrix. In fact, we showed on several standard databases that NMU provides an alternative way to achieve sparse factorizations while keeping a fairly good reconstruction. This is due to our underapproximation formulation, which presents the advantage of not requiring any parameter to be tuned in order to improve sparsity.

Each successive step in the recursive resolution of the underapproximation problem can involve a factorization of any given rank. On the one hand, when using directly the highest possible rank (global L-NMU), we find sparse factorizations with small reconstruction error. On the other hand, building the factorization recursively one rank-one factor at a time (recursive L-NMU) leads to even sparser factorizations, albeit with an increased reconstruction error (due to the greedy approach). This last approach has the advantage that it is able to extract parts in order of importance. Moreover, in a situation where the factorization rank is not fixed a priori, the fact that it is recursive allows the user to stop the procedure as soon as the reconstruction error becomes satisfactory, without having to recompute a completely different solution from scratch every time a higher rank factorization is considered.

References

[1] R. Albright, J. Cox, D. Duling, A. Langville, and C. Meyer, *Initializations and Convergence for the Nonnegative Matrix Factorization*, in 12th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2006.

[2] B. Ames and S. Vavasis, *Nuclear norm minimization for the planted clique and biclique problems.* arXiv:0901.3348v1, 2009.

[3] K. M. Anstreicher and L. A. Wolsey, *Two "well-known" properties of subgradient optimization*, Mathematical Programming, (2007). Forthcoming.

[4] M. Berry, M. Browne, A. Langville, P. Pauca, and R.J. Plemmons, *Algorithms and Applications for Approximate Nonnegative Matrix Factorization*, Computational Statistics and Data Analysis, 52 (2007), pp. 155–173.

[5] M. Berry, N. Gillis, and F. Glineur, *Document Classification Using Nonnegative Matrix Factorization and Underapproximation*, in Proc. IEEE International Symposium on Circuits and Systems (ISCAS), 2009.

[6] M. Biggs, A. Ghodsi, and S. Vavasis, *Nonnegative Matrix Factorization via Rank-One Downdate.* arXiv:0805.0120, 2008.

[7] C. Boutsidis and E. Gallopoulos, *SVD based initialization: A head start for nonnegative matrix factorization*, Journal of Pattern Recognition, 41 (2008), pp. 1350–1362.

[8] S. Boyd, A. Hassibix, S.J. Kimy, and L. Vandenberghe, *A Tutorial on Geometric Programming*, Optimization and Engineering, 8(1) (2007), pp. 67–127.

[9] D. Chen and R. Plemmons, *Nonnegativity Constraints in Numerical Analysis.* Paper presented at the Symposium on the Birth of Numerical Analysis, Leuven Belgium. To appear in the Conference Proceedings, to be published by World Scientific Press, A. Bultheel and R. Cools, Eds., 2007.

[10] C. Cichocki, R. Zdunek, and S. Amari, *Non-negative Matrix Factorization with Quasi-Newton Optimization*, in ICAISC 2006, Zakopane, Poland, June 25-29, 2006, Lecture Notes in Artificial Intelligence, Vol. 4029, Springer, pp. 870-879, 2006.

[11] ——, *Hierarchical ALS Algorithms for Nonnegative Matrix and 3D Tensor Factorization*, in ICA07, London, UK, September 9-12, Lecture Notes in Computer Science, Vol. 4666, Springer, pp. 169-176, 2007.

[12] ——, *Nonnegative Matrix and Tensor Factorization*, IEEE Signal Processing Magazine, (2008), pp. 142–145.

[13] James Curry, Anne Dougherty, and Stefan Wild, *Improving non-negative matrix factorizations through structured initialization*, Journal of Pattern Recognition, 37(11) (2004), pp. 2217–2232.

[14] M. Dawande, P. Keskinocak, and S. Tayur, *On the biclique problem in bipartite graphs.* GSIA Working Paper 1996-04, Carnegie Mellon University, Pittsburgh, PA 15213, USA, 1996.

[15] K. Devarajan, *Nonnegative Matrix Factorization: An Analytical and Interpretive Tool in Computational Biology*, PLoS Computational Biology, 4(7), e1000029 (2008).

[16] I.S. Dhillon, D. Kim, and S. Sra, *Fast Newton-type Methods for the Least Squares Nonnegative Matrix Approximation problem*, in Proceedings of SIAM Conference on Data Mining, 2007.

[17] I.S. Dhillon and S. Sra, *Nonnegative Matrix Approximations: Algorithms and Applications*, tech. report, University of Texas (Austin), 2006. Dept. of Computer Sciences.

[18] C. Ding, T. Li, and M.I. Jordan, *Convex and Semi-Nonnegative Matrix Factorizations*, tech. report, 2006. LBNL 60428.

[19] R.J. Duffin, E.L. Peterson, and C. Zener, *Geometric Programming: Theory and Applications*, Wiley, New York, 1967.

[20] L.R. Ford Jr. and D.R. Fulkerson, *Flows in Networks*, Princeton University press, Paris, 1962.

[21] N. Gillis, *Approximation et sous-approximation de matrices par factorisation positive: algorithmes, complexité et applications*, master's thesis, Université catholique de Louvain, 2007. In French.

[22] N. Gillis and F. Glineur, *Nonnegative Factorization and The Maximum Edge Biclique Problem*. CORE Discussion paper 2008/64, 2008.

[23] ——, *Nonnegative Matrix Factorization and Underapproximation*. Communication at 9th International Symposium on Iterative Methods in Scientific Computing, Lille, France, 2008.

[24] G.H. Golub and C.F. Van Loan, *Matrix Computation, 3rd Edition*, The Johns Hopkins University Press Baltimore, 1996.

[25] F.L. Hitchcock, *The Distribution of a Product from Several Sources to Numerous Localities*, J. Math. Phys., 10 (1941), pp. 224–230.

[26] N.-D. Ho, *Nonnegative Matrix Factorization - Algorithms and Applications*, PhD thesis, Université catholique de Louvain, 2008.

[27] N.-D. Ho, P. Van Dooren, and V. Blondel, *Descent methods for nonnegative matrix factorization*, In: Numerical Linear Algebra in Signals, Systems and Control, Springer Verlag, (2008).

[28] D.S. Hochbaum, *Approximating clique and biclique problems*, J. Algorithms, 29(1) (1998), pp. 174–200.

[29] P.O. Hoyer, *Nonnegative Matrix Factorization with Sparseness Constraints*, J. Machine Learning Research, 5 (2004), pp. 1457–1469.

[30] H. Kim and H. Park, *Non-negative Matrix Factorization Based on Alternating Non-negativity Constrained Least Squares and Active Set Method*, SIAM J. Matrix Anal. Appl.,, to appear (2008).

[31] D.D. Lee and H.S. Seung, *Learning the Parts of Objects by Nonnegative Matrix Factorization*, Nature, 401 (1999), pp. 788–791.

[32] ——, *Algorithms for Non-negative Matrix Factorization*, In Advances in Neural Information Processing, 13 (2001).

[33] B. Levin, *On Calculating Maximum Rank One Underapproximations for Positive Arrays*, tech. report, Columbia University, 1985. Div. of Biostatistics.

[34] Chih-Jen Lin, *On the Convergence of Multiplicative Update Algorithms for Nonnegative Matrix Factorization*, in IEEE Transactions on Neural Networks, 2007.

[35] ——, *Projected Gradient Methods for Nonnegative Matrix Factorization*, Neural Computation, 19 (2007), pp. 2756–2779. MIT press.

[36] P. Paatero and U. Tapper, *Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values*, Environmetrics, 5 (1994), pp. 111–126.

[37] R. Peeters, *The maximum edge biclique problem is NP-complete*, Discrete Applied Mathematics, 131(3) (2003), pp. 651–654.

[38] F. Sha, Y. Lin, L.K. Saul, and D.D. Lee, *Multiplicative Updates for Nonnegative Quadratic Programming*, Neural Computation, 19 (2007), pp. 2004–2031.

[39] Stephen Vavasis, *On the Complexity of Nonnegative Matrix Factorization*. arXiv:0708.4149v2, 2007.

[40] Q. Zhang, H. Wang, R. Plemmons, and P. Pauca, *Tensor Methods for Hyperspectral Data Analysis of Space Objects*. To appear in J. Optical Soc. Amer. A, 2008.

Recent titles

CORE Discussion Papers

2008/52. Yuri YATSENKO, Raouf BOUCEKKINE and Natali HRITONENKO. Estimating the dynamics of R&D-based growth models.

2008/53. Roland Iwan LUTTENS and Marie-Anne VALFORT. Voting for redistribution under desert-sensitive altruism.

2008/54. Sergei PEKARSKI. Budget deficits and inflation feedback. 2008/55. Raouf BOUCEKKINE, Jacek B. KRAWCZYK and Thomas VALLEE. Towards an understanding of tradeoffs between regional wealth, tightness of a common environmental constraint and the sharing rules.

2008/56. Santanu S. DEY. A note on the split rank of intersection cuts.

2008/57. Yu. NESTEROV. Primal-dual interior-point methods with asymmetric barriers.

2008/58. Marie-Louise LEROUX, Pierre PESTIEAU and Gregory PONTHIERE. Should we subsidize longevity?

2008/59. J. Roderick McCRORIE. The role of Skorokhod space in the development of the econometric analysis of time series.

2008/60. Yu. NESTEROV. Barrier subgradient method.

2008/61. Thierry BRECHET, Johan EYCKMANS, François GERARD, Philippe MARBAIX, Henry TULKENS and Jean-Pascal VAN YPERSELE. The impact of the unilateral EU commitment on the stability of international climate agreements.

2008/62. Giorgia OGGIONI and Yves SMEERS. Average power contracts can mitigate carbon leakage.

2008/63. Jean-Sébastien TANCREZ, Philippe CHEVALIER and Pierre SEMAL. A tight bound on the throughput of queueing networks with blocking.

2008/64. Nicolas GILLIS and François GLINEUR. Nonnegative factorization and the maximum edge biclique problem.

2008/65. Geir B. ASHEIM, Claude D'ASPREMONT and Kuntal BANERJEE. Generalized time-invariant overtaking.

2008/66. Jean-François CAULIER, Ana MAULEON and Vincent VANNETELBOSCH. Contractually stable networks.

2008/67. Jean J. GABSZEWICZ, Filomena GARCIA, Joana PAIS and Joana RESENDE. On Gale and Shapley '*College admissions and stability of marriage*'.

2008/68. Axel GAUTIER and Anne YVRANDE-BILLON. Contract renewal as an incentive device. An application to the French urban public transport sector.

2008/69. Yuri YATSENKO and Natali HRITONENKO. Discrete-continuous analysis of optimal equipment replacement.

2008/70. Michel JOURNÉE, Yurii NESTEROV, Peter RICHTÁRIK and Rodolphe SEPULCHRE. Generalized power method for sparse principal component analysis.

2008/71. Toshihiro OKUBO and Pierre M. PICARD. Firms' location under taste and demand heterogeneity.

2008/72. Iwan MEIER and Jeroen V.K. ROMBOUTS. Style rotation and performance persistence of mutual funds.

2008/73. Shin-Huei WANG and Christian M. HAFNER. Estimating autocorrelations in the presence of deterministic trends.

2008/74. Yuri YATSENKO and Natali HRITONENKO. Technological breakthroughs and asset replacement.

2008/75. Julio DÁVILA. The taxation of capital returns in overlapping generations economies without financial assets.

2008/76. Giorgia OGGIONI and Yves SMEERS. Equilibrium models for the carbon leakage problem.

2008/77. Jean-François MERTENS and Anna RUBINCHIK. Intergenerational equity and the discount rate for cost-benefit analysis.

2008/78. Claire DUJARDIN and Florence GOFFETTE-NAGOT. Does public housing occupancy increase unemployment?

Recent titles

CORE Discussion Papers - continued

2008/79. Sandra PONCET, Walter STEINGRESS and Hylke VANDENBUSSCHE. Financial constraints in China: firm-level evidence.
2008/80. Jean GABSZEWICZ, Salome GVETADZE, Didier LAUSSEL and Patrice PIERETTI. Pubic goods' attractiveness and migrations.
2008/81. Karen CRABBE and Hylke VANDENBUSSCHE. Are your firm's taxes set in Warsaw? Spatial tax competition in Europe.
2008/82. Jean-Sébastien TANCREZ, Benoît ROLAND, Jean-Philippe CORDIER and Fouad RIANE. How stochasticity and emergencies disrupt the surgical schedule.
2008/83. Peter RICHTÁRIK. Approximate level method.
2008/84. Çağatay KAYI and Eve RAMAEKERS. Characterizations of Pareto-efficient, fair, and strategy-proof allocation rules in queueing problems.
2009/1. Carlo ROSA. Forecasting the direction of policy rate changes: The importance of ECB words.
2009/2. Sébastien LAURENT, Jeroen V.K. ROMBOUTS and Francesco VIOLANTE. Consistent ranking of multivariate volatility models.
2009/3. Dunia LÓPEZ-PINTADO and Juan D. MORENO-TERNERO. The principal's dilemma.
2009/4. Jacek B. KRAWCZYK and Oana-Silvia SEREA. A viability theory approach to a two-stage optimal control problem of technology adoption.
2009/5. Jean-François MERTENS and Anna RUBINCHIK. Regularity and stability of equilibria in an overlapping generations model with exogenous growth.
2009/6. Nicolas GILLIS and François GLINEUR. Using underapproximations for sparse nonnegative matrix factorization.

Books

H. TULKENS (ed.) (2006), *Public goods, environmental externalities and fiscal competition.* New York, Springer-Verlag.
V. GINSBURGH and D. THROSBY (eds.) (2006), *Handbook of the economics of art and culture.* Amsterdam, Elsevier.
J. GABSZEWICZ (ed.) (2006), La différenciation des produits. Paris, La découverte.
L. BAUWENS, W. POHLMEIER and D. VEREDAS (eds.) (2008), *High frequency financial econometrics: recent developments*. Heidelberg, Physica-Verlag.
P. VAN HENTENRYCKE and L. WOLSEY (eds.) (2007), *Integration of AI and OR techniques in constraint programming for combinatorial optimization problems*. Berlin, Springer.
P-P. COMBES, Th. MAYER and J-F. THISSE (eds.) (2008), *Economic geography: the integration of regions and nations*. Princeton, Princeton University Press.
J. HINDRIKS (ed.) (2008), *Au-delà de Copernic: de la confusion au consensus* ? Brussels, Academic and Scientific Publishers.

CORE Lecture Series

C. GOURIÉROUX and A. MONFORT (1995), Simulation Based Econometric Methods.
A. RUBINSTEIN (1996), Lectures on Modeling Bounded Rationality.
J. RENEGAR (1999), A Mathematical View of Interior-Point Methods in Convex Optimization.
B.D. BERNHEIM and M.D. WHINSTON (1999), Anticompetitive Exclusion and Foreclosure Through Vertical Agreements.
D. BIENSTOCK (2001), Potential function methods for approximately solving linear programming problems: theory and practice.
R. AMIR (2002), Supermodularity and complementarity in economics.
R. WEISMANTEL (2006), Lectures on mixed nonlinear programming.